POWERING POTENTIAL

Welcome to:
COURSE TITLE

x Day Course

09.30 – 11.00
11.15 – 12.30
13.30 – 15.00
15.15 – 16.30

Trainer:

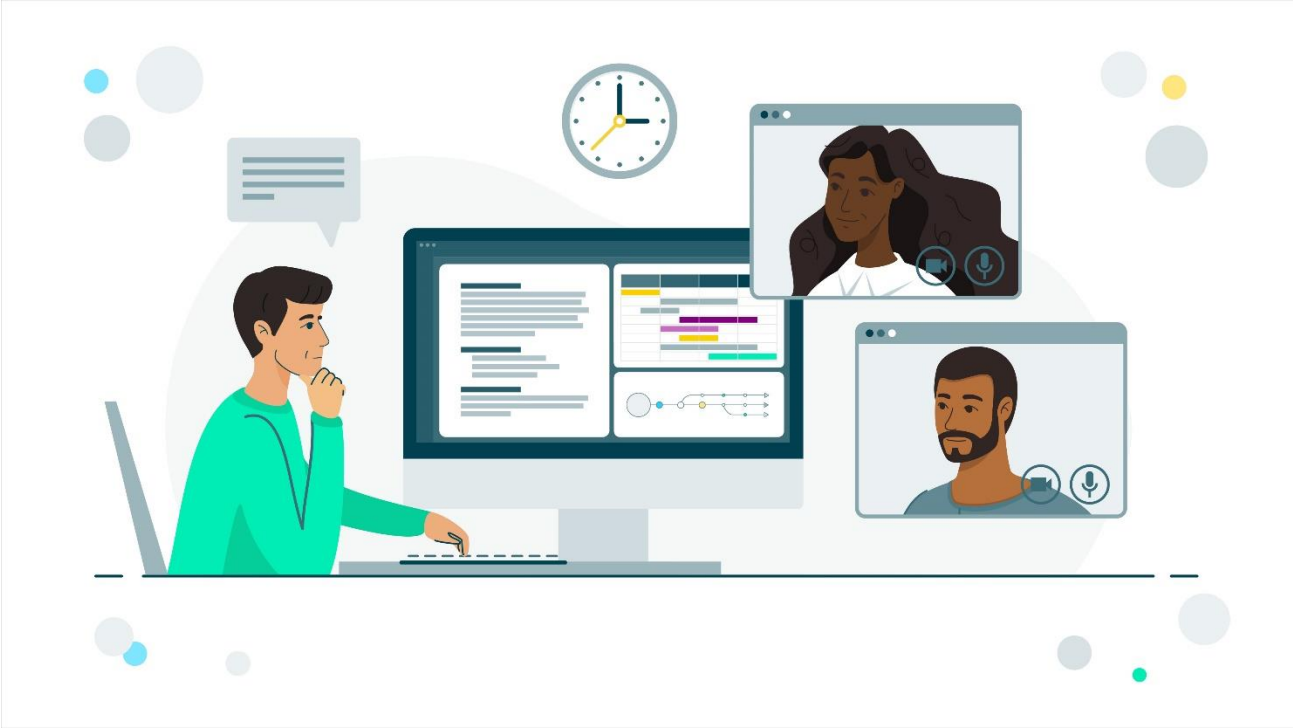
Housekeeping

AM breaks
Lunch
PM breaks

WE WOULD LOVE TO SEE YOU...

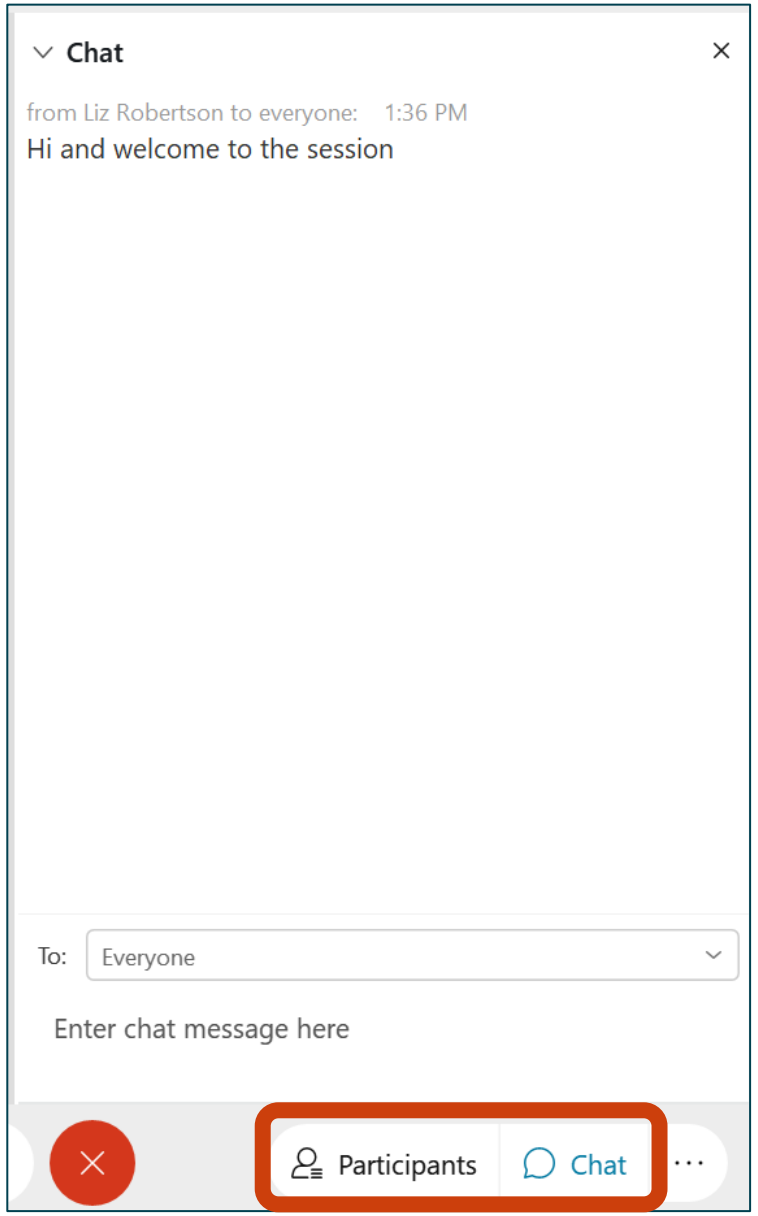
PLEASE START YOUR VIDEO!

Click on Start / Stop video at the bottom of the meeting window.



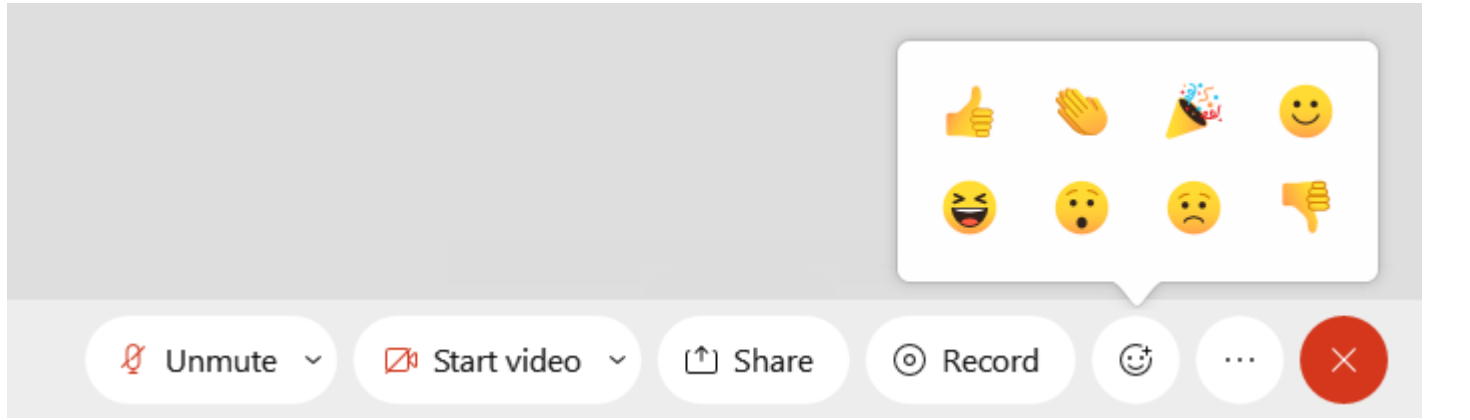
CHAT WINDOW

- Ask and answer questions
- Add comments



REACTIONS AND RAISED HANDS

- Finding a topic interesting?
- Got a question?



Troubleshooting

If **I** disconnect:	• One of you will automatically become the host and when I return, I'll resume the course. • Please don't leave the session.
If **you** disconnect:	• Don't panic! • Re-join using your link.
Contact:	• **Chat link from your joining email** • virtual.learningteam@qa.com • 0203 908 2376

Any questions?

Just ask!

INTRODUCTIONS

Introductions

- Name
- Where do you work?
- Knowledge and experience
- Your aims for the course?

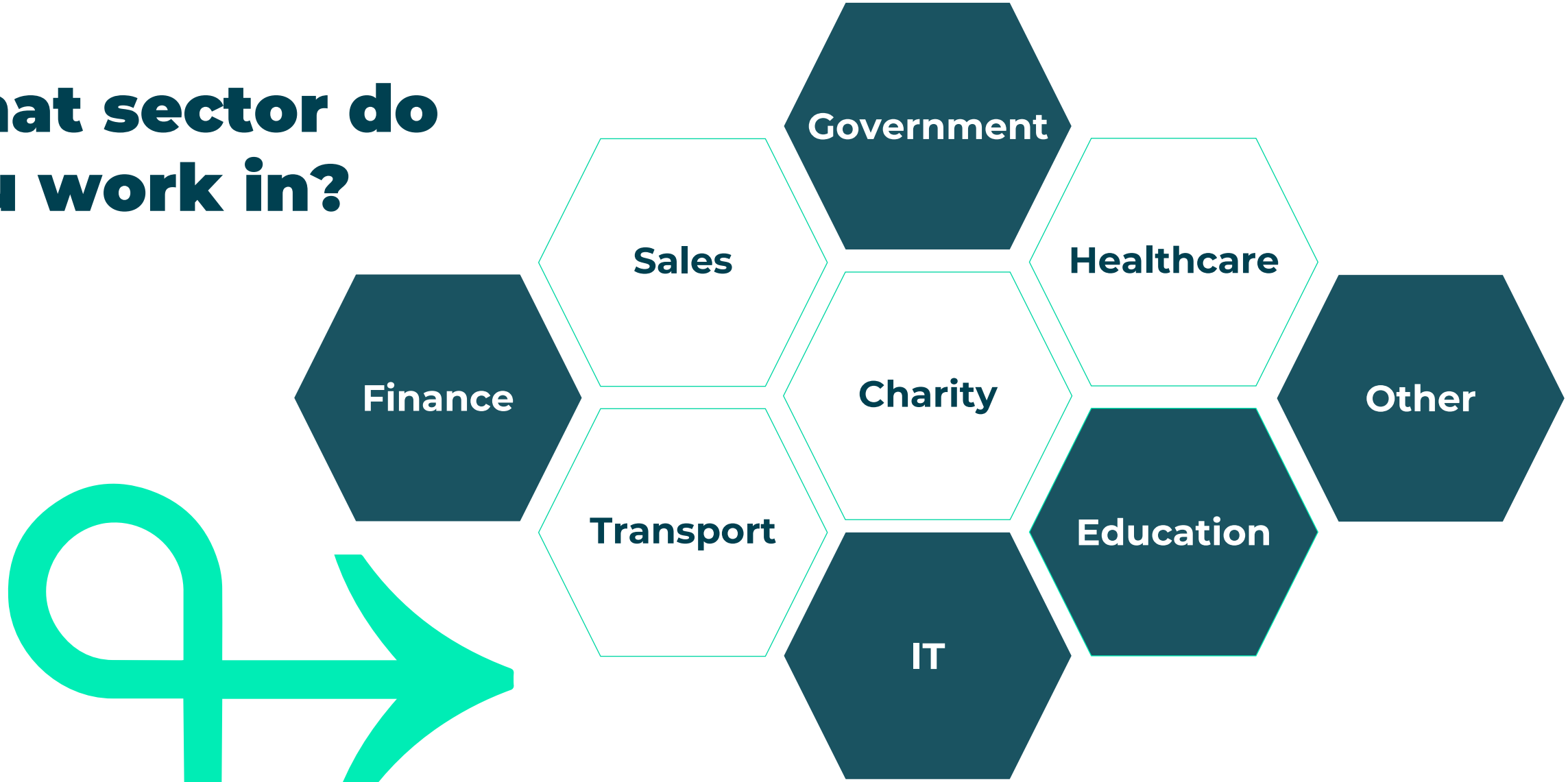
YOUR EXPERIENCE

Basic

Intermediate

Expert

What sector do you work in?



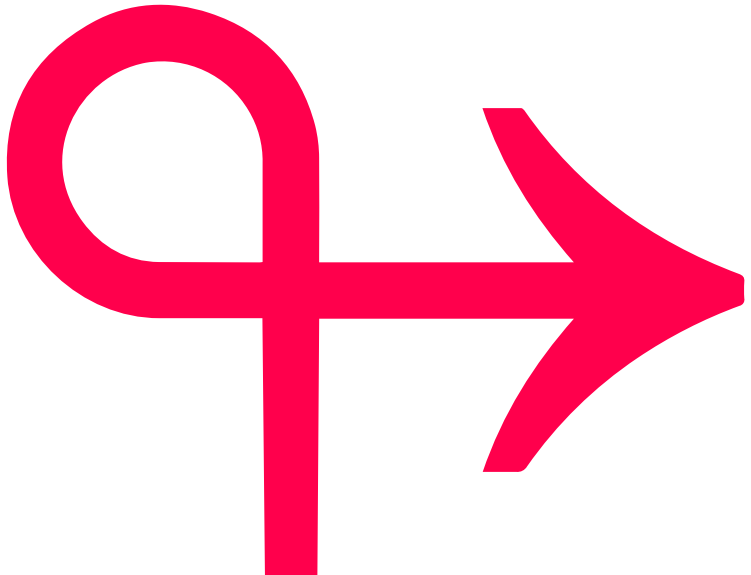
POWERING POTENTIAL

TITLE

			Data Scientists	Data Engineers	Data Analysts
Introduction to Data Science for Data Professionals					
R for Data Handling					
Statistics for Data Analysis					
R Programming					
R for Data Science and Machine Learning					
Practical Big Data Analysis					
Fundamentals of AI and Deep Learning					
Maths and Statistics for Data Science, Big Data, and Operational Analysis					
Machine Learning, AI, and Deep Learning with R					
Microsoft Azure	**Amazon AWS**	**Google GCP**			
MDP900	AMWPDSAS	GCPFBDML			
MDP100	AMWSMLP	GCPDEGP		DRAFT	

Course overview

- Summary of the course aims
- Should match the marketing materials for the course
- Pre-requisites

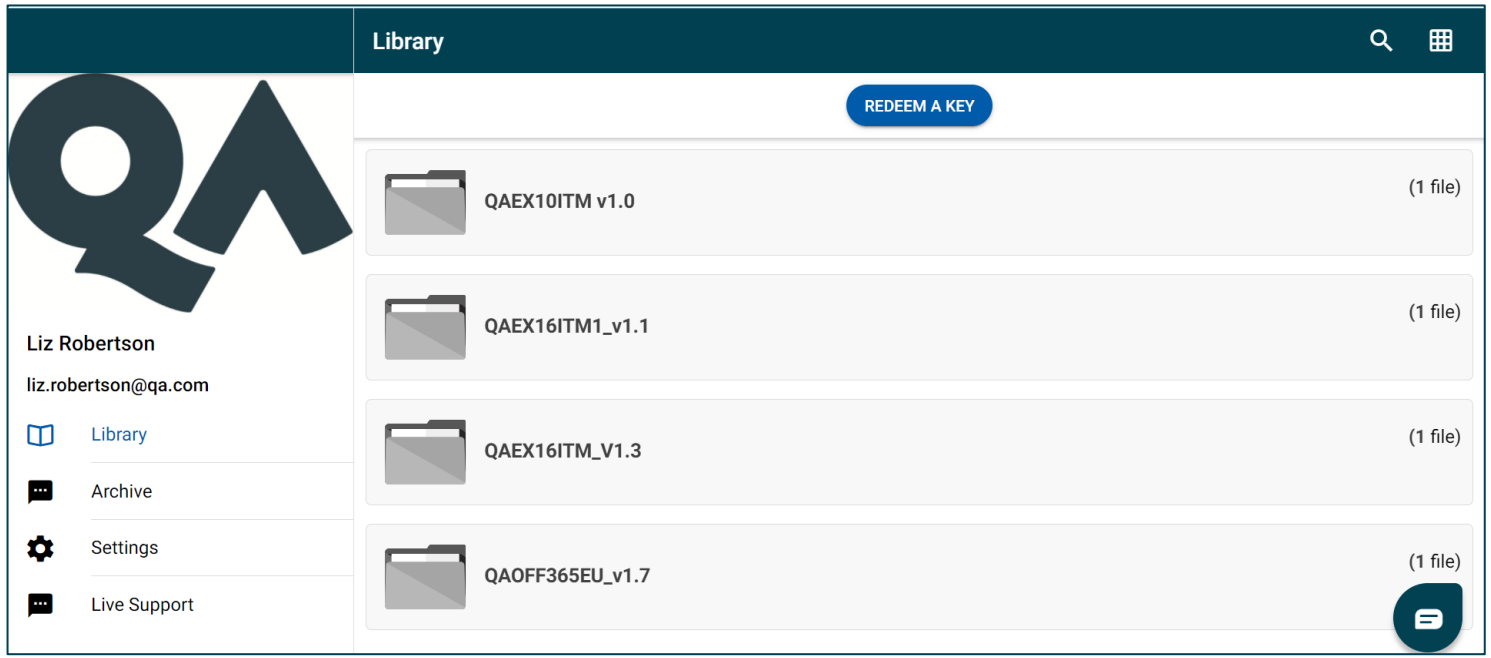


DIGITAL COURSEWARE

Visit www.mimeo.digital/#/

Create an account

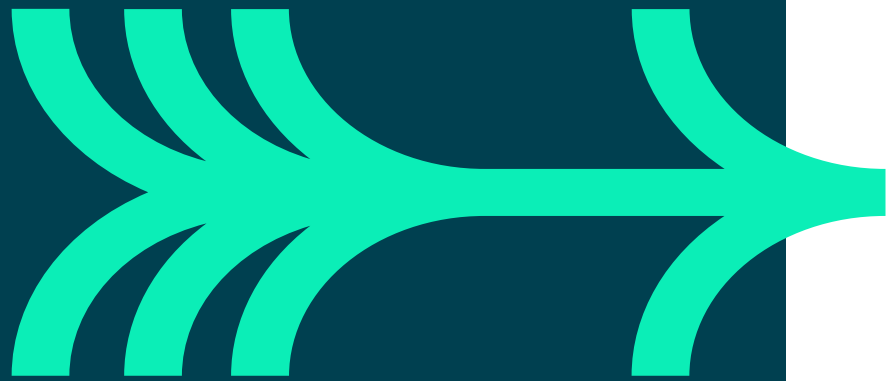
REDEEM YOUR KEY: XXXX



VIRTUAL MACHINES

LET'S GET STARTED...

COURSE OUTLINE



DAY 1 : SUMMARY THEME

- Module 1:
- Module 2:
- Module 3:

DAY 2: SUMMARY THEME

- Module 4:
- Module 5:
- Module 6:

DAY 3: SUMMARY THEME

- Module 7:
- Module 8:
- Module 9:

DAY 4: SUMMARY THEME

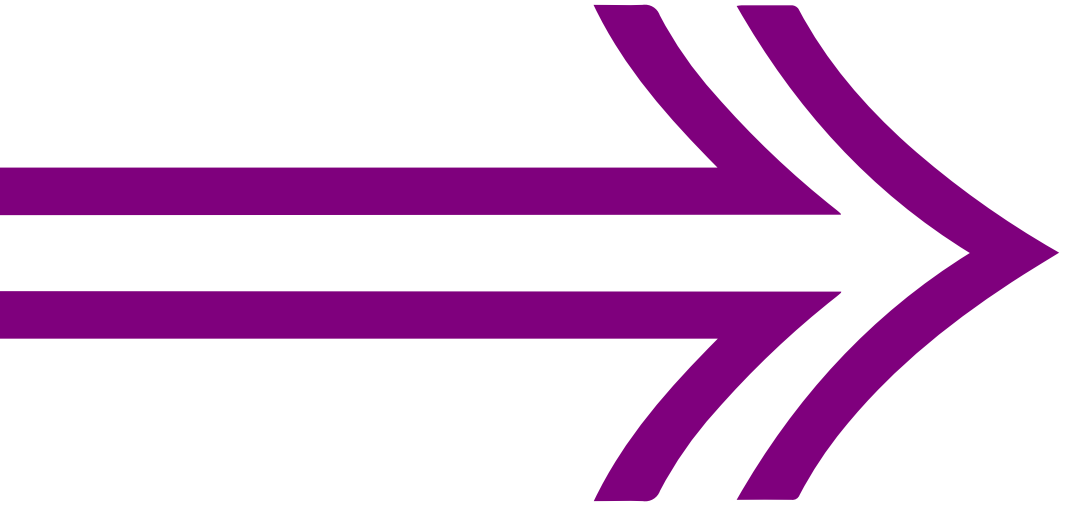
- Module 10:
- Module 11:
- Module 12:

DAY 5: SUMMARY THEME

- Module 13:
- Module 14:
- Module 15:

Day 2 starter

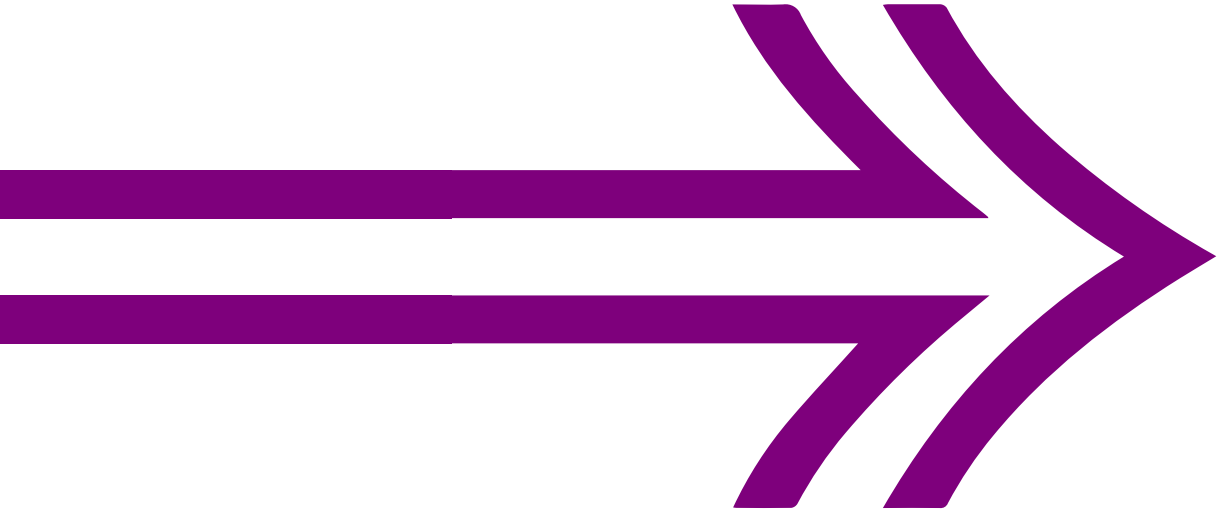
- Module 4:
- Module 5:
- Module 6:



REFLECTION

What have you learnt since the start of the course?

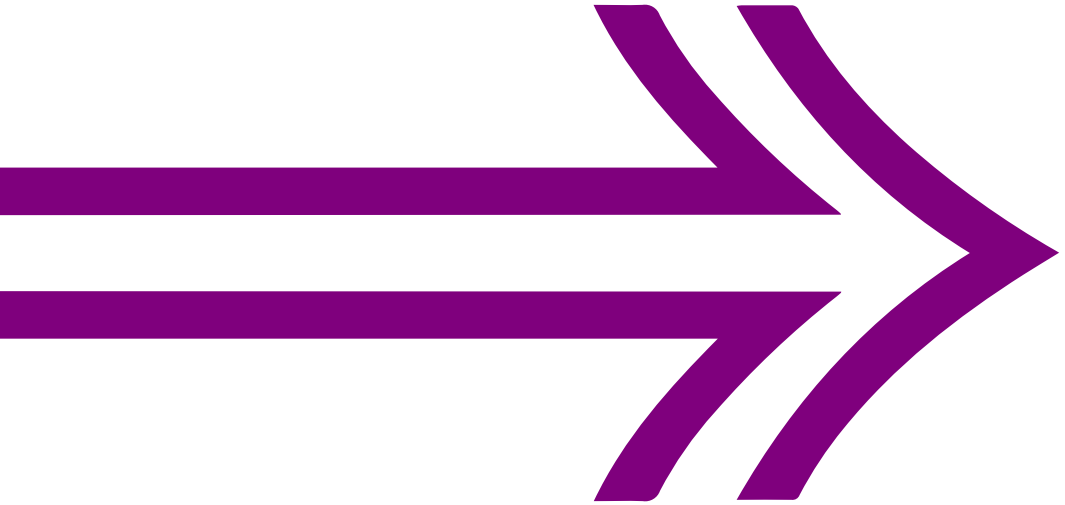
What are you looking forward to learning next?



SEE YOU TOMORROW!

Day 3 starter

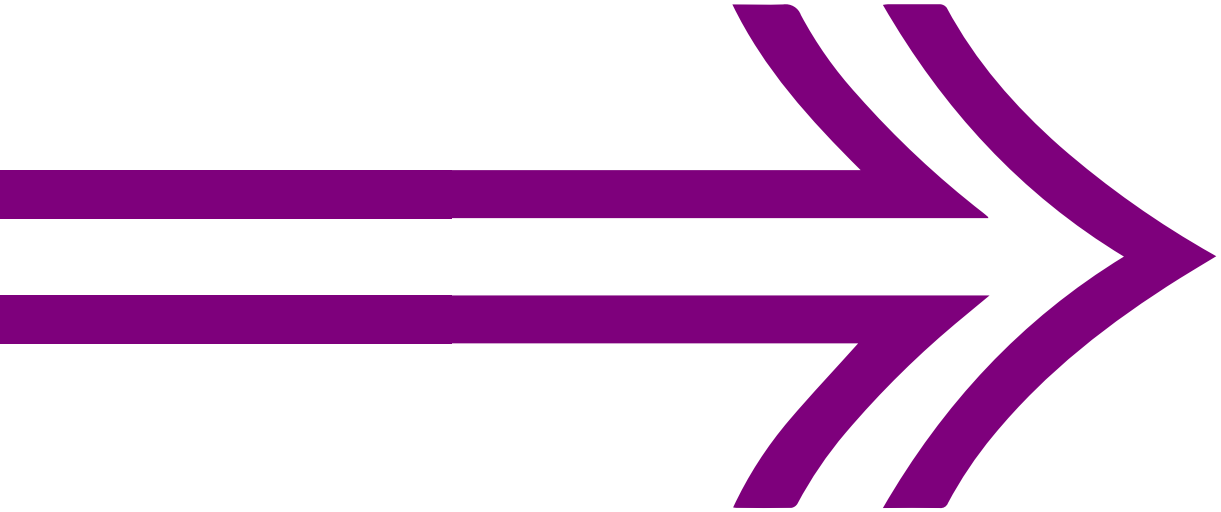
- Module 7:
- Module 8:
- Module 9



REFLECTION

What have you learnt since the start of the course?

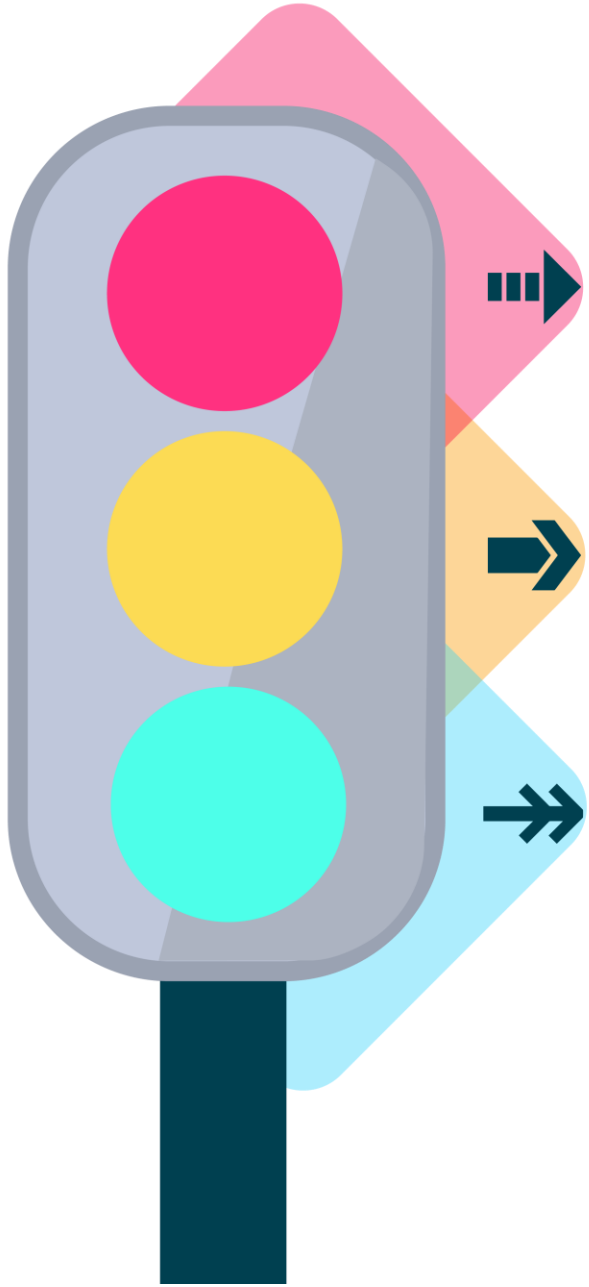
What are you looking forward to learning next?



SEE YOU TOMORROW!

LET'S REVIEW

WHAT ARE YOU GOING TO...



...Stop?

...Continue?

...Start?

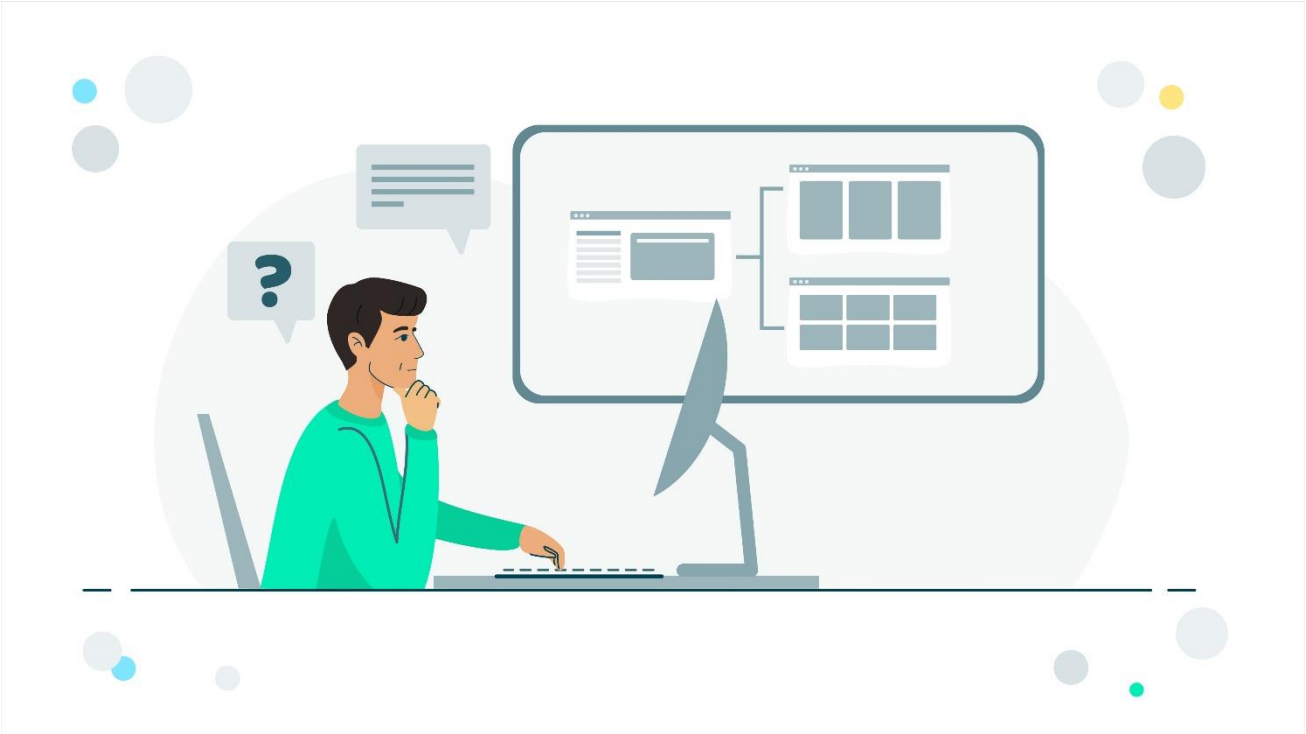
Future course options and support

			Data Scientists	Data Engineers	Data Analysts
Introduction to Data Science for Data Professionals					
R for Data Handling					
Statistics for Data Analysis					
R Programming					
R for Data Science and Machine Learning					
Practical Big Data Analysis					
Fundamentals of AI and Deep Learning					
Maths and Statistics for Data Science, Big Data, and Operational Analysis					
Machine Learning, AI, and Deep Learning with R					
Microsoft Azure	**Amazon AWS**	**Google GCP**			
MDP900	AMWPDSAS	GCPFBDML			
MDP100	AMWSMLP	GCPDEGP		DRAFT	

Congratulations and thank you!

You will receive your course attendance certificate via email.

WHAT DID YOU THINK?



https://evaluation.qa.com

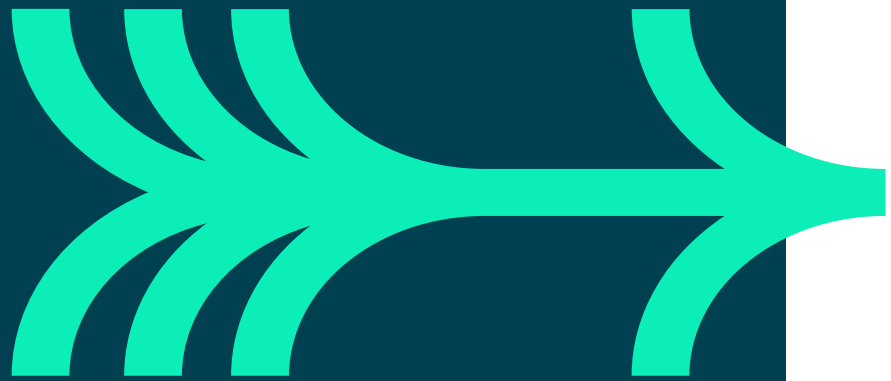
Course code:
XXXXXXX

PIN:
XXXX

Leaving the session?

Click FILE >
Leave Training Session

INTRODUCTION TO DATA SCIENCE AND MACHINE LEARNING



Learning objectives

- Explain the role of the Data Scientist and the skillset it requires.
- Describe common application areas of Data Science, and examples of its usage in industry.
- Outline the Data Science process detailed in the CRISP-DM methodology.
- Detail the characteristics of problems which Data Science can be used to solve.
- Define how to evaluate the success of a Data Science project.

Expected prior knowledge

- Nothing is assumed about your background.

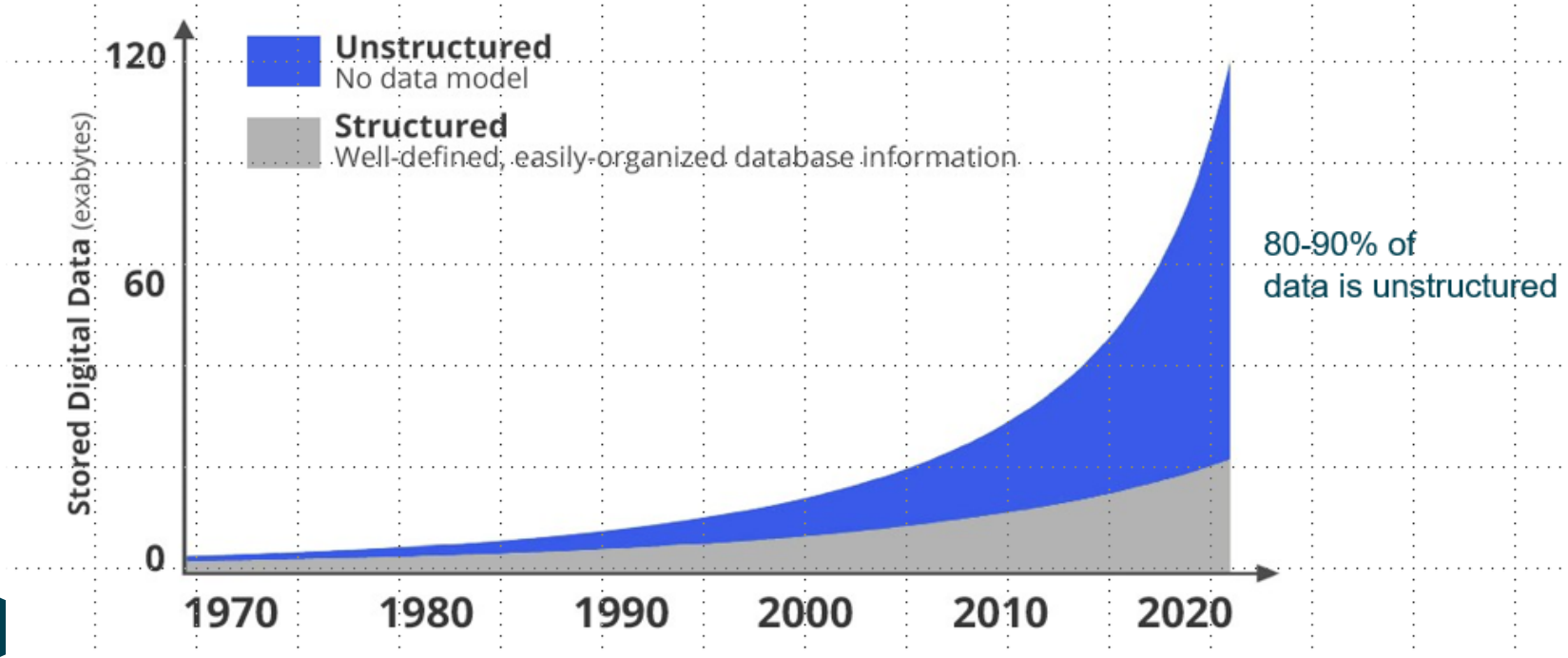
What is Data Science?

WHAT IS DATA SCIENCE

Data science is exploring, simplifying, visualising, teasing out information, and fitting models to data in order to provide analytical insight into big data.

WHY DATA SCIENCE?

- **Volume of unstructured data is growing, and that growth is accelerating**



DATA SCIENTISTS

Data scientists:

- can work with both structured and unstructured data.
- collect, prepare, and process raw data.
- develop predictive models.
- test and implement machine learning algorithms.
- identify organisational information requirements.
- communicate recommendations to senior stakeholders.
- look to unlock insights for the future.

Analysis of text

Social media accounts, see what the general feel is about a certain topic

Medical analysis

Whether you're likely to get a certain illness based on your lifestyle or other factors

Financial

Is the market going to go up or down?

Environment

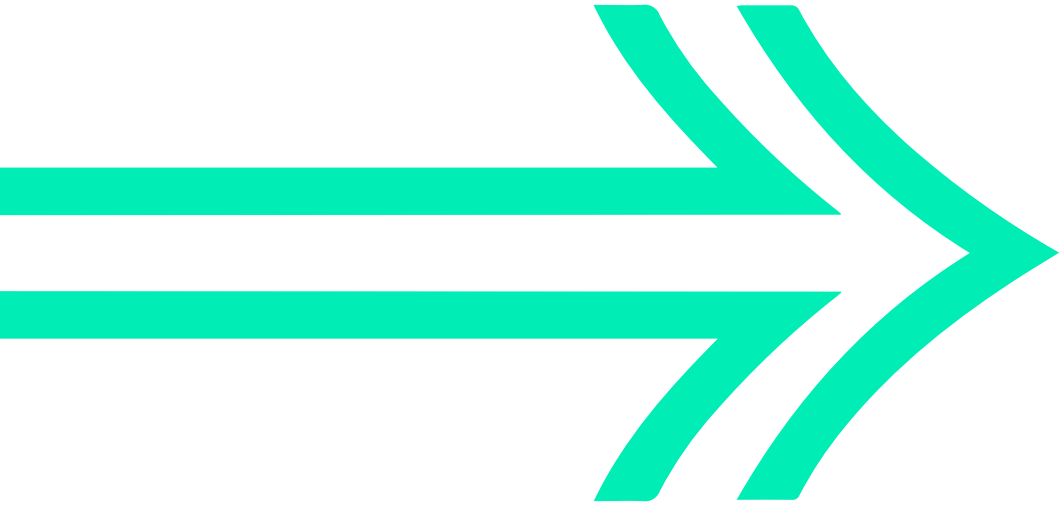
What's going on with the weather? How can we work out if it's going to rain?

Retail

Who's most likely to buy this thing?

Personal

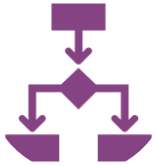
Why can't I hit 10,000 steps a day?



What questions can we ask?

DATA SCIENCE SKILLS

Mathematics and data analytics



Algorithms and implementation

Domain application

Community engagement

Policy, security, and ethics

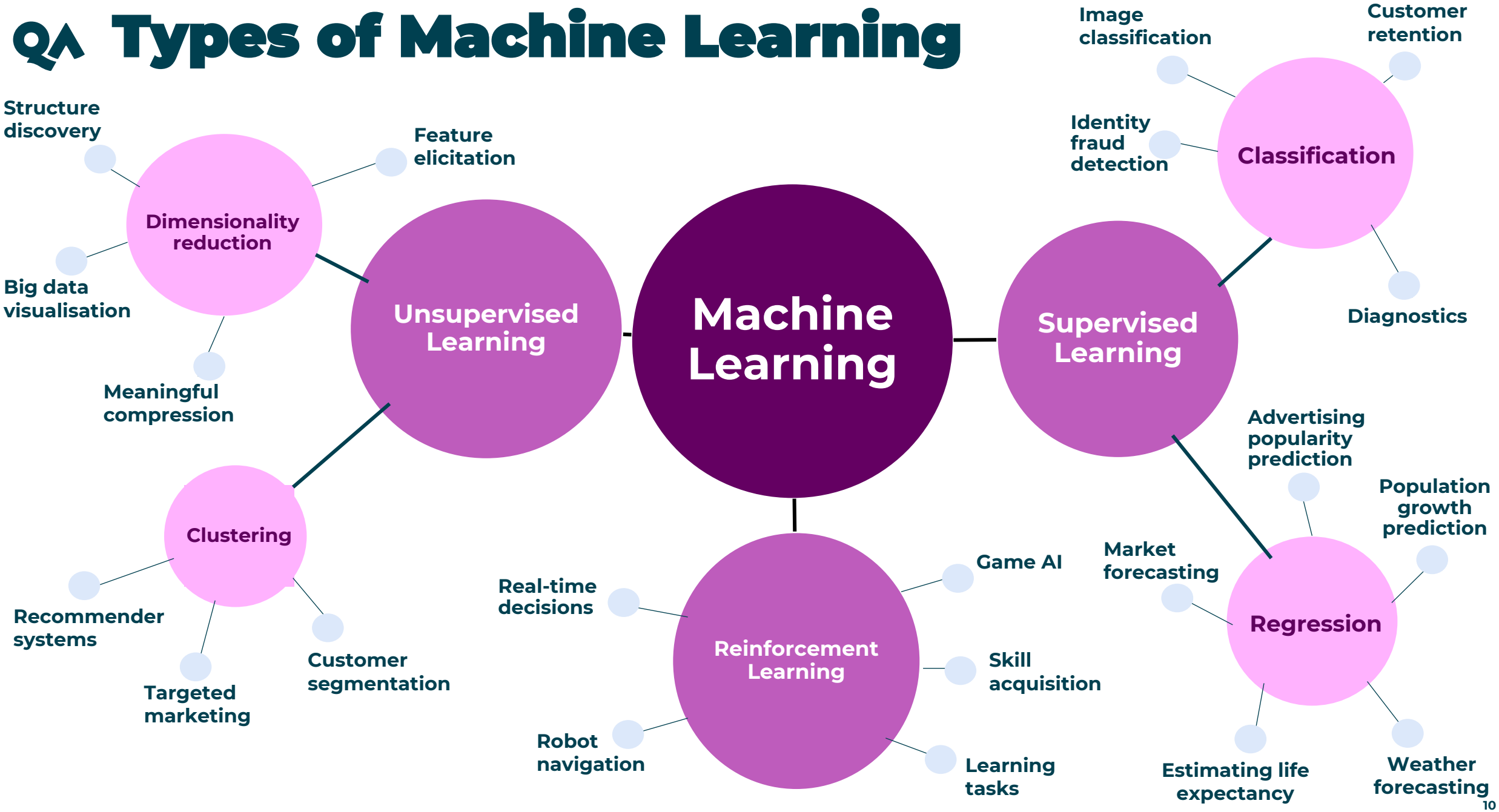
What is Machine Learning?

WHAT IS MACHINE LEARNING

Machine learning uses sophisticated algorithms to ‘learn’ from data.

It is a sub-field of Artificial Intelligence.

Types of Machine Learning



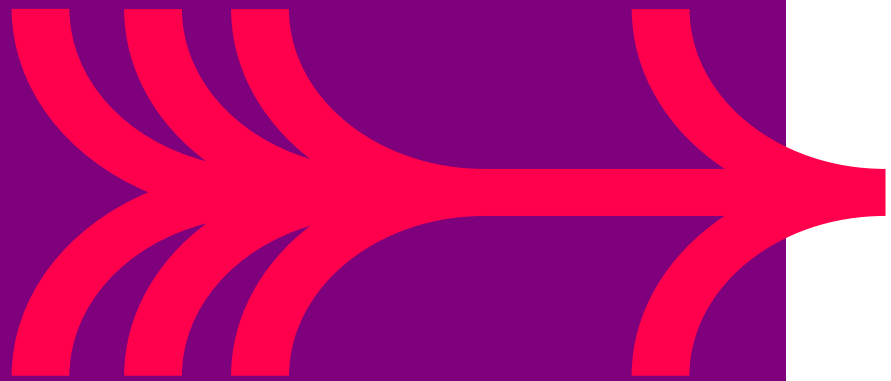
EXERCISE

Identify a domain of interest.

Within that domain, define:

- a regression problem.
- classification.
- clustering.

Consider the data you would need to solve the problem.



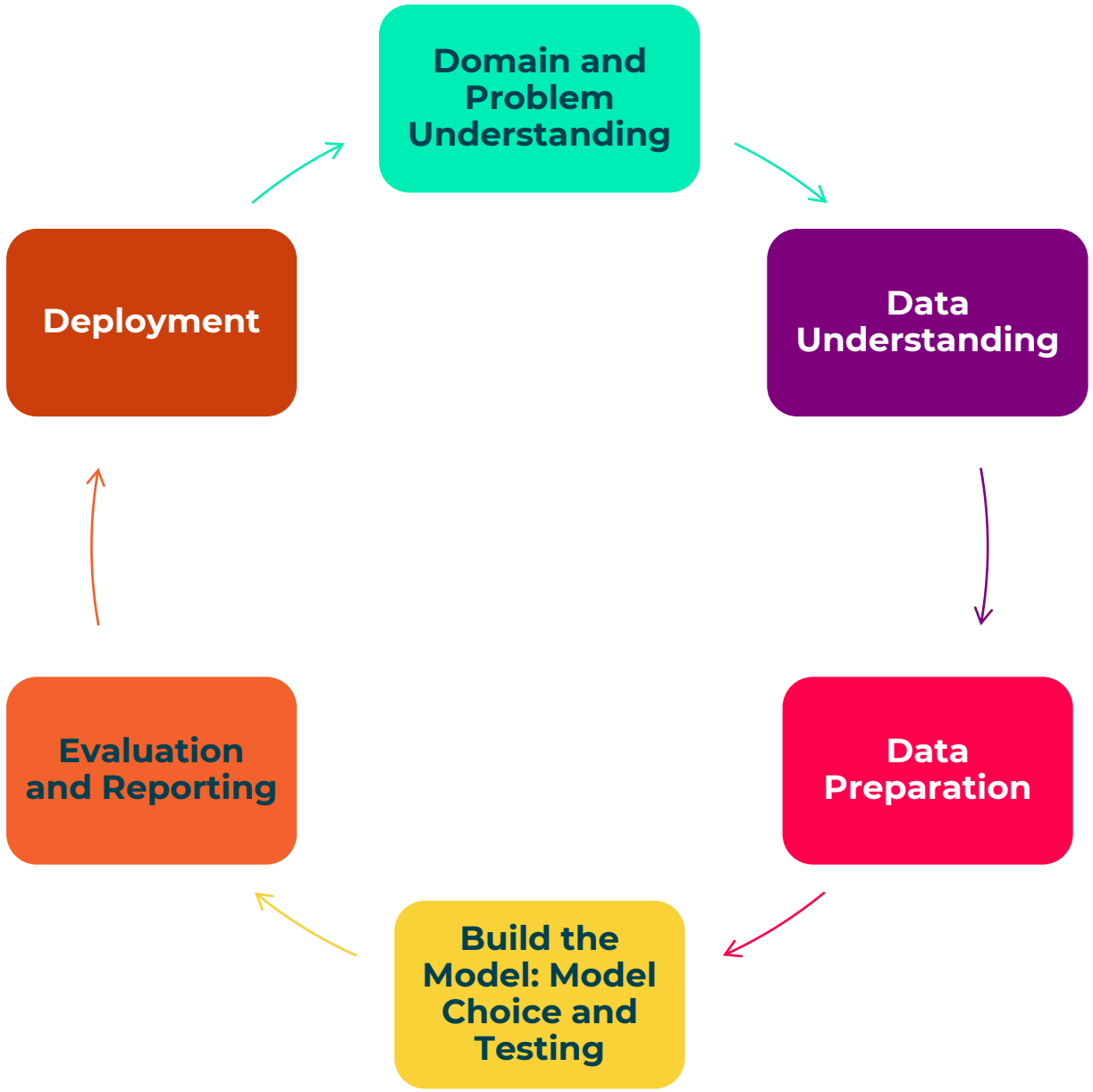
Data science projects

WHAT MAKES A GOOD PROJECT?

Good candidate projects are typically:

- rich in data.
- human-time consuming.
- well-defined.
- important enough to warrant a project.

DATA SCIENCE PROJECT LIFECYCLE

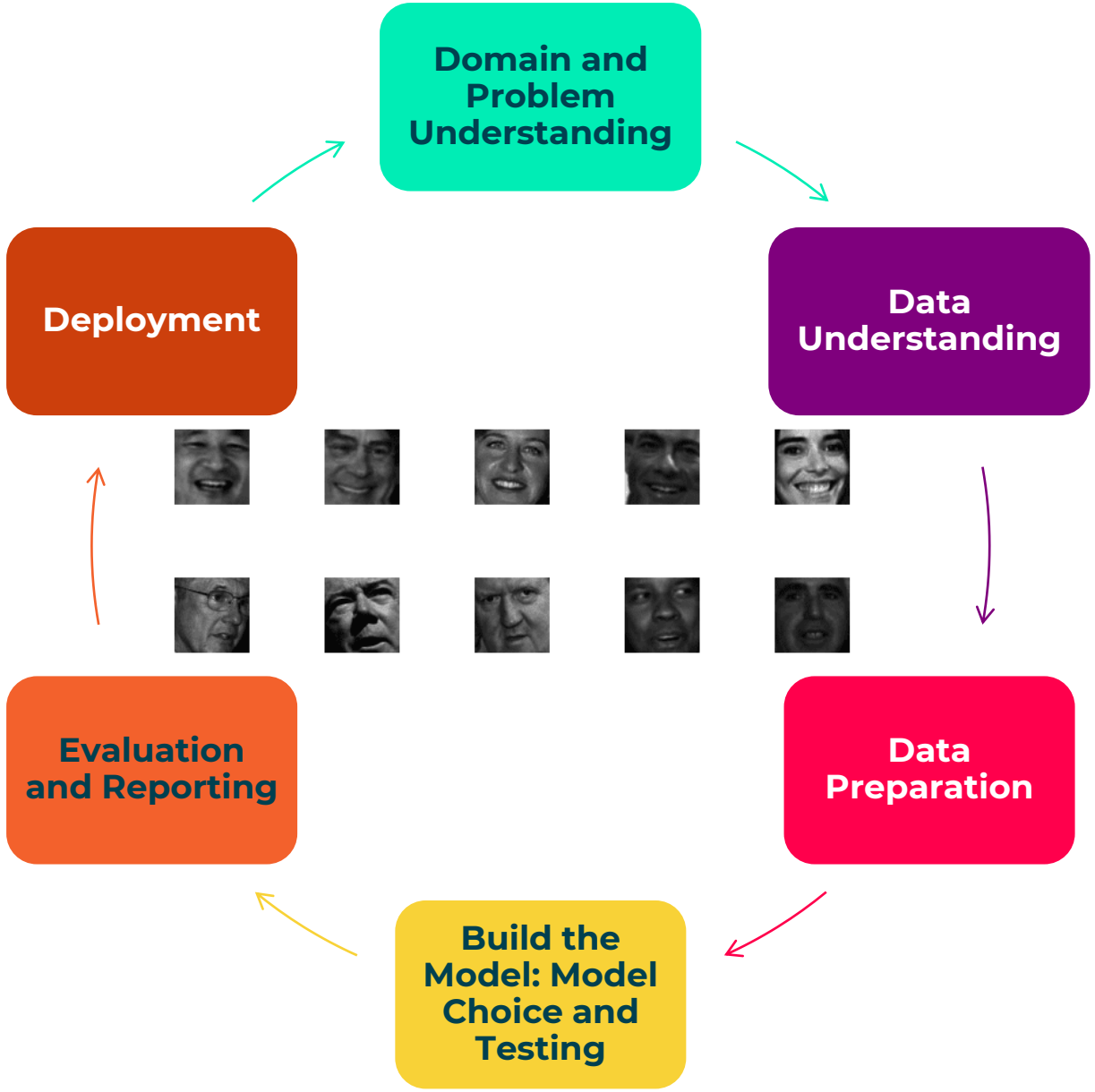


EXAMPLE: SMILING DETECTION

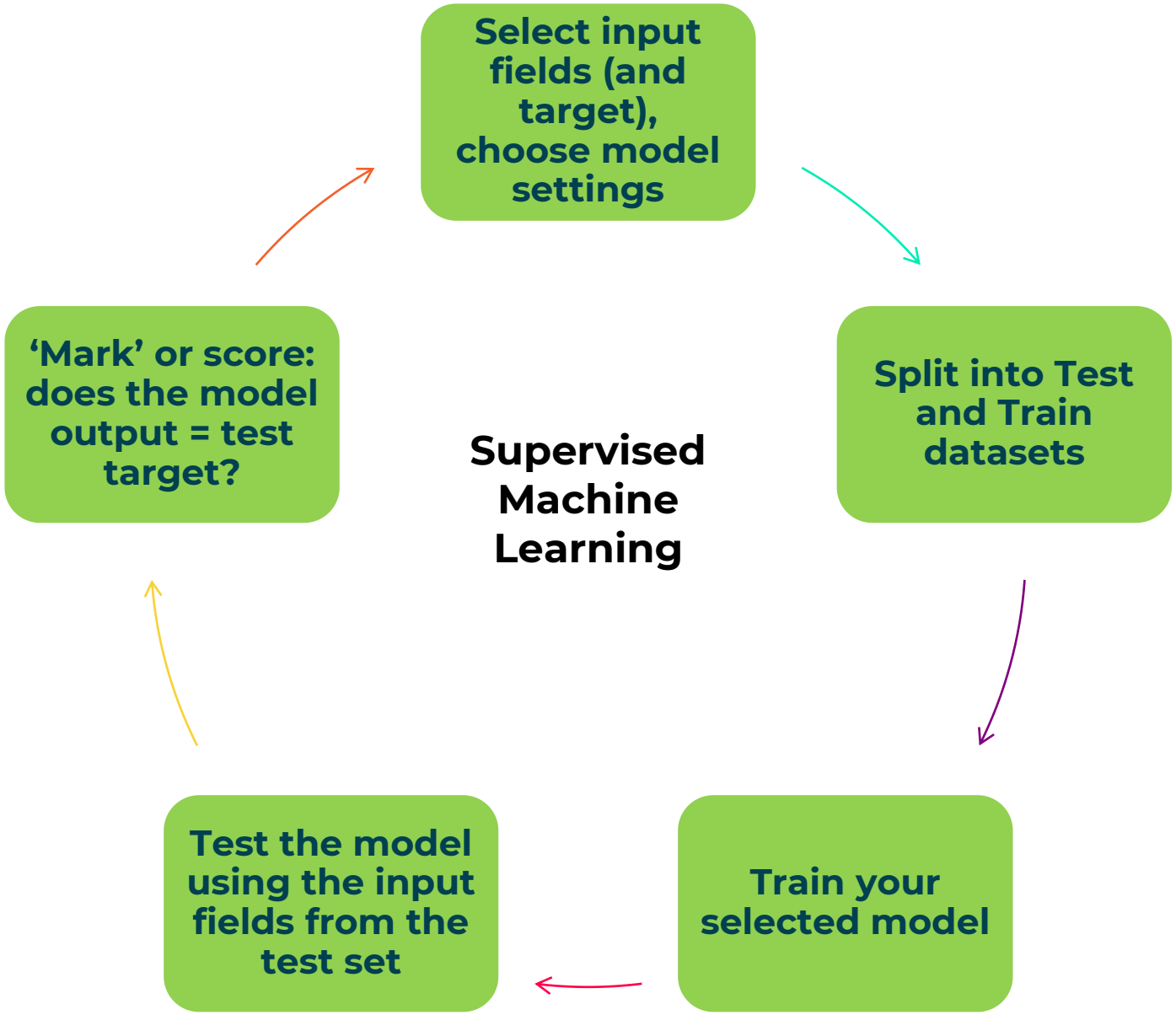
In which of these photos do you think someone is smiling?

There are strict rules for passport photos – many countries now have automatic photo checking tools. Smile detection is just one of the things these systems could check for.

DISCUSSION: WHAT THINGS DO YOU THINK WOULD BE INVOLVED AT EACH STAGE OF THE PROJECT?



BUILD THE MODEL



DATA SCIENCE PROJECT PITFALLS

Which stages would you expect to take the longest? What could go wrong and what could you do to avoid / mitigate?

Hints:

- Sources of data being used.
- Robust exploration right from the initial stage.
- Appropriate skills and / or training for team members.
- Stick to the agreed objectives.
- Consider how to analyse and evidence the business value of the results.
- Ensure there is sufficient governance oversight prior to deployment.

EVALUATING DATA SCIENCE PROJECT SUCCESS

Projects success can be measured using traditional metrics, such as:

- ROI.
- improvement over existing process (A/B test).
- customer satisfaction.

However, traditional ROI may not capture:

- improvement in organisational capabilities.
- time saved on future development.

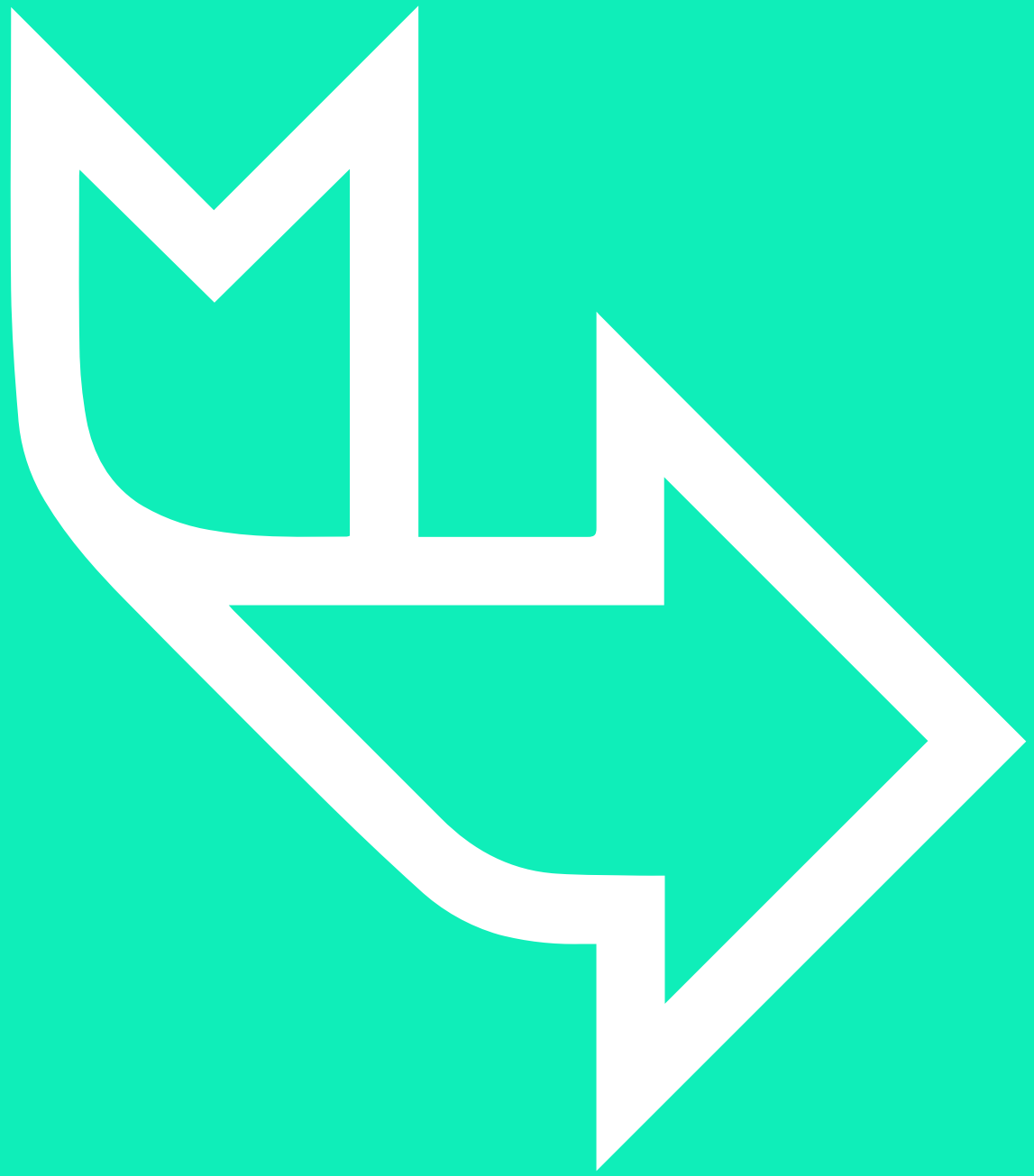
QA Roles for operationalising ML / AI models

Data Scientist	Analysis techniques expertise, data modelling, Machine Learning / AI, Model selection, meets project objectives.
Data Architect and / or Governance Lead	Database configuration requirements for analysis, access to key data sources, ensure there is appropriate human oversight.
Data Engineer	Data extraction and manipulation, data modelling (may not be a separate role from the data scientist depending on the organisation).
Data Analyst	Project analysis, use of reporting tools for the project or for creation of automated monitoring dashboards.
AIOps	Support with automating processes and monitoring ongoing model performance.
Project Manager (meeting chair)	Setting and monitoring of objectives and project time management.
End user (consult)	Strong domain knowledge, knowledge of data sources, will usually benefit from the results. May be a business analyst or manager.
Project sponsor (consult)	Project driver, clarifies desired outputs, usually with some seniority in the organisation – framing the core business problem, provides funding / resourcing.

LEARNING CHECK

Think about your answers to these questions:

- What are the three over-arching types of Machine Learning?
- Can you give an example technique for each one?
- What is the typical flow of a data science project?
- How can we evaluate data science project success?



HOW DID YOU GET ON?

Learning objectives

- Explain the role of the Data Scientist and the skillset it requires.
- Describe common application areas of Data Science, and examples of its usage in industry.
- Outline the Data Science process detailed in the CRISP-DM methodology.
- Detail the characteristics of problems which Data Science can be used to solve.
- Define how to evaluate the success of a Data Science project.

Appendix: Quotes

DATA SCIENTIST

‘A data scientist is someone who is better at statistics than any software engineer and better at software engineering than any statistician.’ – Wills (2012, Director of Data Science at Cloudera)

‘...the sexiest job... will be statisticians... data... understand it... extract value from it... visualise it...’ – Varian (2008, Chief Economist at Google)

‘A data scientist is somebody who is inquisitive, who can stare at data and spot trends... like a Renaissance individual who really wants to learn and bring change to an organisation.’ – Bhambhi (2012, VP of Big Data products at IBM)

DATA SCIENTIST

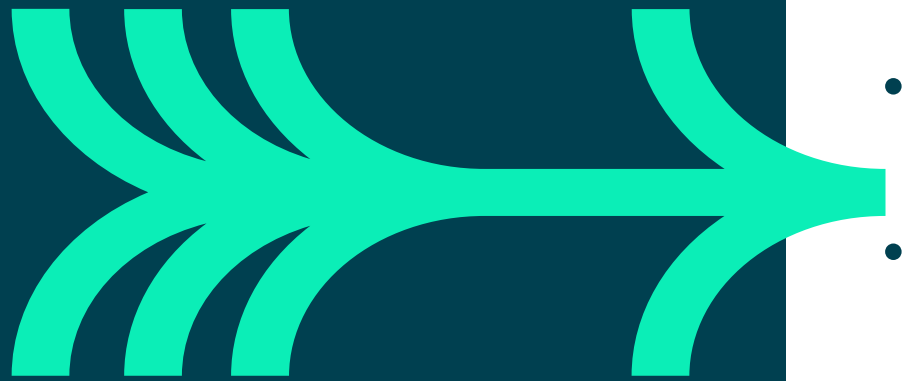
‘Effective data science should not be treated like just another business process and cannot be operationalised assembly-line style.

Data science – as the name suggests – is a mode of inquiry and exploration similar to “real” science.

Just as a physicist uses math to reason about the natural world, data scientists harness mathematical and computational tools to reason about the business world.’

– Peter Wang (2019, CEO Anaconda)

ARTIFICIAL INTELLIGENCE



Neural Networks
A particular pattern-finding algorithm.

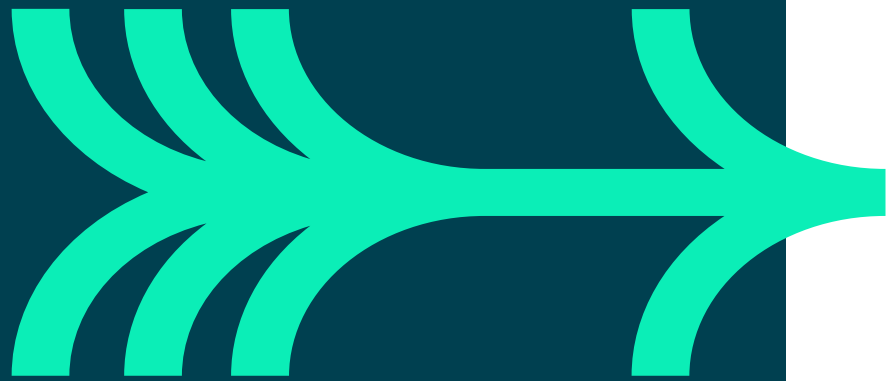
Deep Neural Networks
A particular generalisation of the neural network algorithm.

Artificial Intelligence
(boad def.) the project of simulating animal intelligence.

Artificial Intelligence

- whatever artificial system is the best at decision making.
- **50s:** Computers running simple programs.
- **80s:** Computers running expert programs.
- **00s:** Computers running machine learning programs.
- **10s:** Computers running machine learning with neural networks.
- **20s:** Computers running machine learning with huge neural networks.

INTRODUCTION TO PYTHON FOR DATA SCIENCE



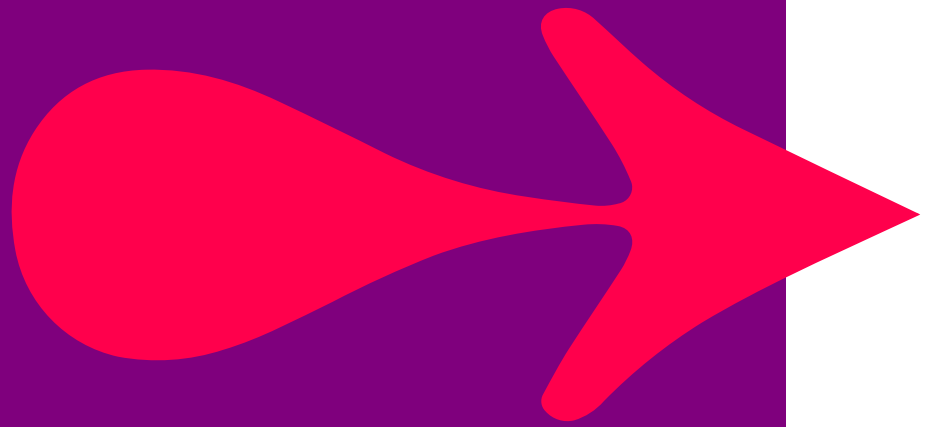
Learning objectives

- Understand why notebooks are often used in Data Science projects.
- Use Python and associated libraries to manipulate datasets.
- Describe why virtual environments are used.
- Visualise data using Python.

Expected prior knowledge

- Nothing is assumed about your background.

ACTIVITY: DISCUSSION



What do you know about the Python programming language?

For example:

- What type of language is it?
- Is it high or low level?
- How is it organised?
- Is it compiled or interpreted?
- How does it enforce type?

WHAT IS PYTHON?

Python is an object-oriented scripting language

- First published in 1991 by Guido van Rossum.
- Designed as an OOP language from day one.
- Does not need knowledge of OO to use.

It is powerful

- General purpose, fully functional, rich.
- Many extension modules.

It is free

- Open source: Python licence is less restrictive than GPL.

It is portable

- UNIX, Linux, Windows, OS X, Android, etc...
- Ported to the Java and .NET virtual machines.

The most common version is written in C

- No platform specific functions in the base language.

Virtual environments

VIRTUAL ENVIRONMENTS

- Best practice is one per project
- A folder containing everything your project needs
  - Including Python!
  - Modules and Packages
- Managed using package manager
  - PIP
  - Conda

Anaconda and Jupyter

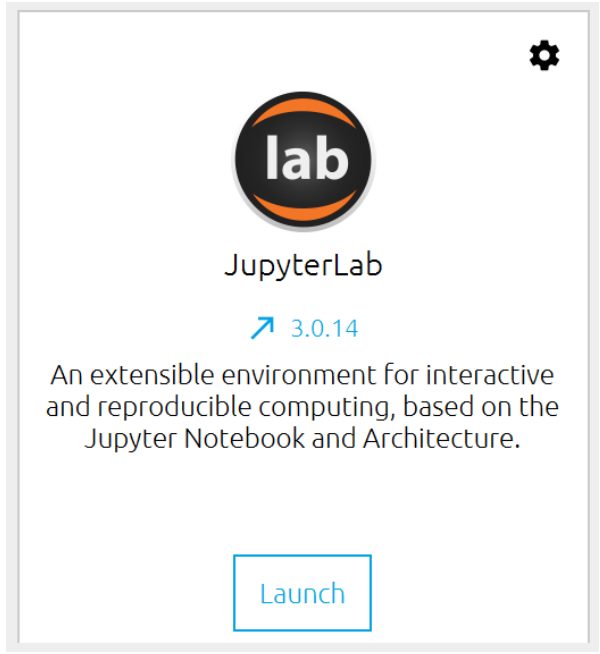
Anaconda and Jupyter

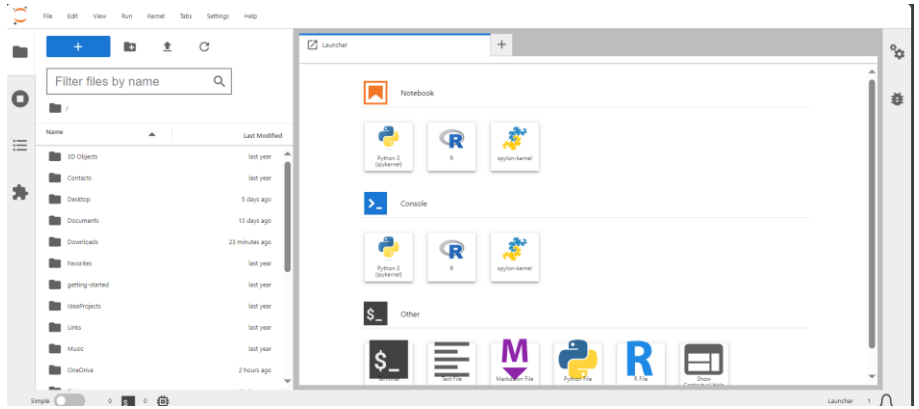
Anaconda is a collection of Python packages, programs, and other software for data analysis and data science.

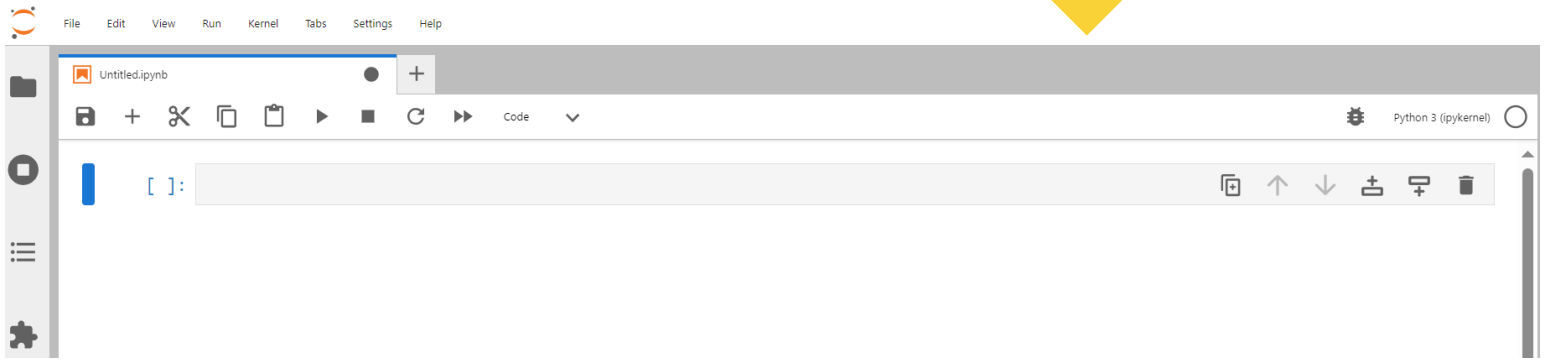
Jupyter notebooks are a documentation or commentary-first style of programming, where the key elements are explanations and comments. The code is less important, and often only small amounts.

WHY JUPYTERLAB?

- JupyterLab is a superb web-based interactive application for working with Jupyter Notebooks
  - User friendly
  - Easy to debug at each line of code







ANACONDA NAVIGATOR

How do I start Anaconda?

Start Menu > Anaconda Navigator

How do I start JupyterLab?

In Anaconda Navigator, press LAUNCH underneath the JupyterLab icon.

JupyterLab

WORKING WITH CELLS

How do I add and modify cells?

Command mode:	ESC (blue bar)
Arrows:	move around notebook
m:	markdown / text cell
y:	code cell / Python cell
a:	insert cell above
b:	insert cell below
dd:	delete cell
z:	undo delete

CTRL + ENTER to run

SHIFT + ENTER to run **and** move one cell below

WORKING WITH CELLS

How do I edit the contents of cells?

- Edit mode: ENTER (green bar)
- Type
- Use arrows to move around text
- CTRL+ENTER: run cell
- SHIFT+ENTER: run cell and move one cell below

WORKING WITH CELLS

How do I add Python code to a notebook?

- Add a cell (e.g., press ‘b’) and press ENTER to edit.

How do I run a Python code cell?

- Press CTRL + ENTER or the RUN button.

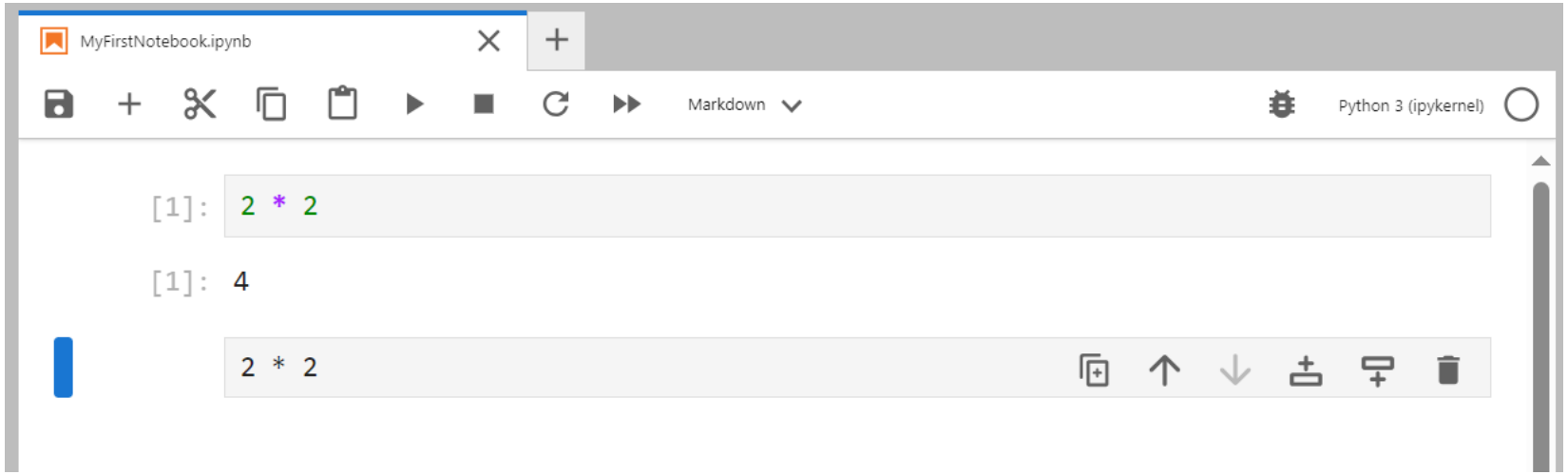
What happens when I run a code cell in Jupyter?

- Hopefully you don’t get any errors!
- Jupyter always ‘print()’s the last line.
- In a usual Python IDE, you would ‘print()’ everything you wish to show on the screen. (What does IDE stand for?)

WORKING WITH CELLS

How do I add a text cell?

- Add a cell, change type to markdown, **or**
- Press ‘m’ in **command mode**.



How do I get help?

- ‘command?’
- ‘command??’
- ‘help(command)’

WORKING WITH CELLS

How do I find the arguments of a function?

- Import statistics
- statistics.mean?

How do I stop Jupyter from printing the last line of a cell?

- Use ‘;’

How do I run command line programs within Jupyter?

- Exclamation mark
- !dir

How do I install Python packages with Jupyter?

- !conda install plotly pyspark

Python fundamentals

VARIABLES AND OBJECTS

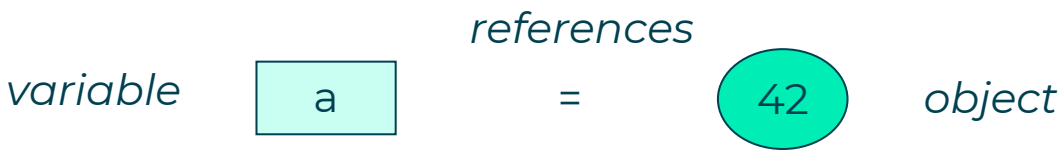
If that is an object, what is a variable?

What is an object?

- To a programmer the term describes a specific area of memory.
- Objects have type, state, and identity.

An object's type is called its class

- Describes the size and format of the area of memory.
- Describes the actions which may be carried out on the object.



Python variables are references to objects

- Variables can be deleted with **`del`**
- An object's memory can be reused when it is no longer referenced.

CAREFUL!

PYTHON RESERVED WORDS

False	None	True	and	as*	assert
async^	await^	break	class	continue	def
del	elif	else	except	exec~	finally
for	from	global	if	import	in
is	lambda	nonlocal+	not	or	pass
raise	return	try	while	with*	yield

* version 2.6 and later
\+ version 3.0
~ not in version 3.0
^ version 3.7

exec and **print** were keywords prior to 3.0, now they are built-in functions.

Python 3 types

- Numbers

```
3.142, 42, 0x3f, 0o664
```

Sequences

Immutable

- Bytes

```
b'Norwegian Blue', b"Mr. Khan's bike"
```

- Strings

```
'Norwegian Blue', "Mr. Khan's bike", r'C:\Numbers'
```

- Tuples

```
(47, 'Spam', 'Major', 683, 'Ovine Aviation')
```

Mutable

- Lists

```
['Cheddar', ['Camembert', 'Brie'], 'Stilton']
```

- Bytearrays

```
bytearray(b'abc')
```

- Dictionaries

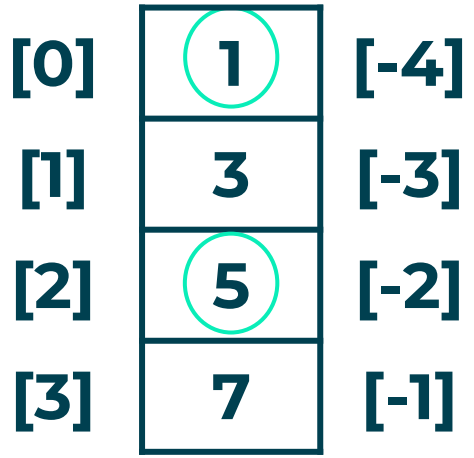
```
{'Sword':'Excalibur', 'Bird':'Unladen Swallow'}
```

- Sets

```
{'Chapman', 'Cleese', 'Idle', 'Jones', 'Palin'}
```

ADDRESSING AN ELEMENT USING ITS INDEX

```
numbers = [1,3,5,7]

print(numbers[0])
print(numbers[2])
print(numbers[-2])
```

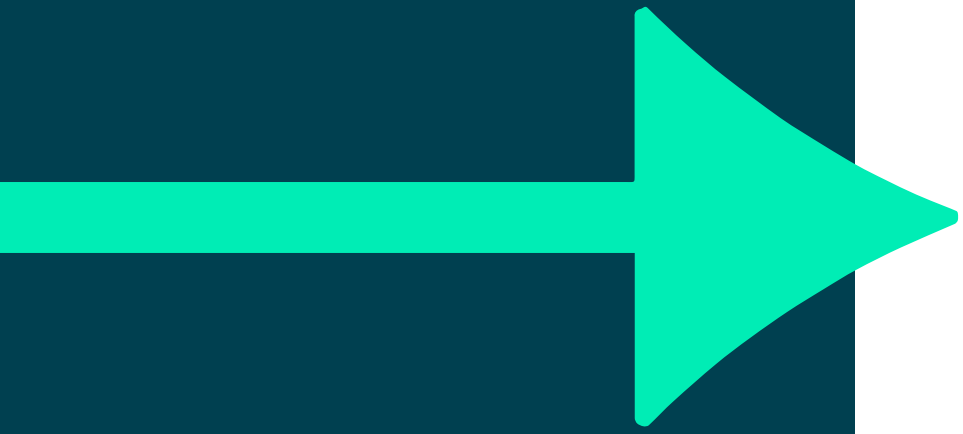



```
names = ["Bob", "Steve", "Helen"]

print(names[1])
Print(names[-1])
```

Applies to strings(lists of characters)too

```
s = 'Hello world!'
print(s[1])
print(s[-5])
```

ADDRESSING MULTIPLE ELEMENTS – SLICING



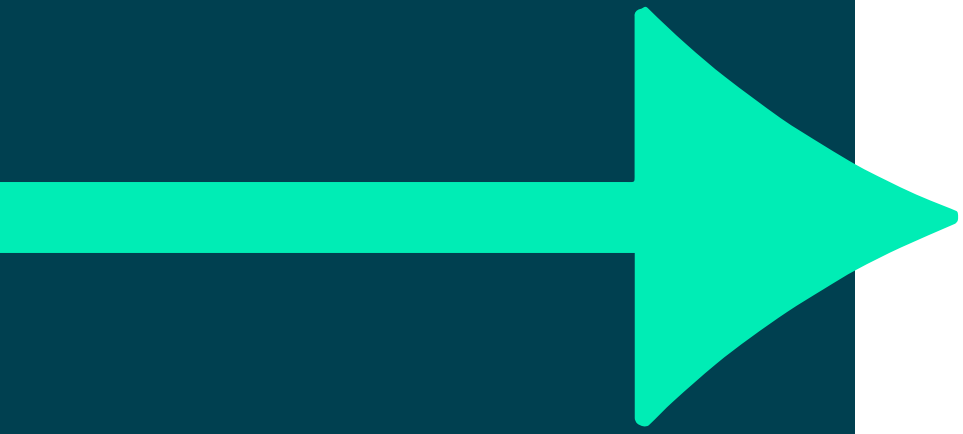
identifier[start:stop:step]

- ***stop*** is the position after the last index
- Default ***start*** is 0
- Default ***stop*** is the last index
- Default ***step*** is 1

```
numbers = [1,3,5,7,9,11,13,15,17,19,21]

print(numbers[3:10])
print(numbers[2:])
print(numbers[:6])
Print(numbers[::2])
```

```
Applies to strings (lists of characters) too
s = 'Hello world!'
print(s[3:10])
print(s[:6])
Print(s[::-1])
```

USER DEFINED FUNCTIONS



No parameters, no return value

```
def hello():
    print("Hello world")
```

Calling the function and output:

```
hello()
```

```
Hello world
```

Function with parameters, no return value

```
def hello(name):
    print("Hello", name)
```

Calling the function and output:

```
hello("everybody")
```

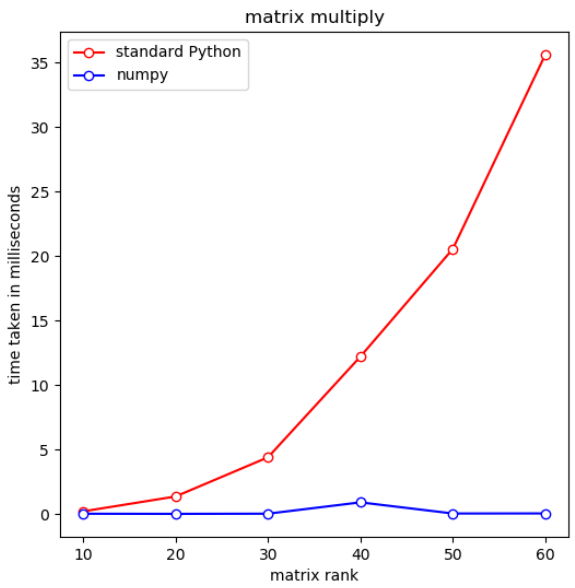
```
Hello everybody
```

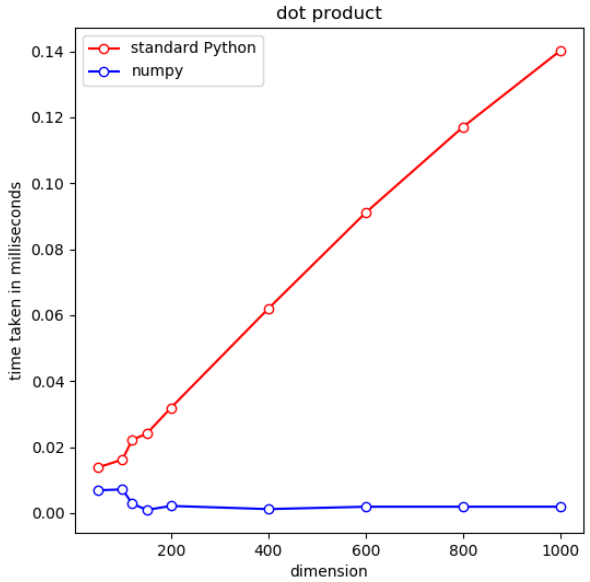
Why NumPy?

PYTHON IS SLOW

Python collections are not designed for computational efficiency.

C and FORTRAN arrays are much more efficient computationally than Python lists for large datasets.





WHAT IS NUMPY?

NumPy was introduced in 2006 to address the inefficiencies of Python in dealing with large amounts of data.

- Written in C and FORTRAN
- Internal data structure uses C arrays
- Python API for seamless integration with Python
- **Provides its own array types (ND-arrays)**
- **Arrays retain most Python collection behaviours, so that it looks and feels 'native' to Python language**
- Incorporates fast maths libraries, such as OpenBLAS (default, open source), for efficient linear algebraic operations (dot products, matrix multiply, etc.)

Note: To use NumPy, it is necessary to import it.

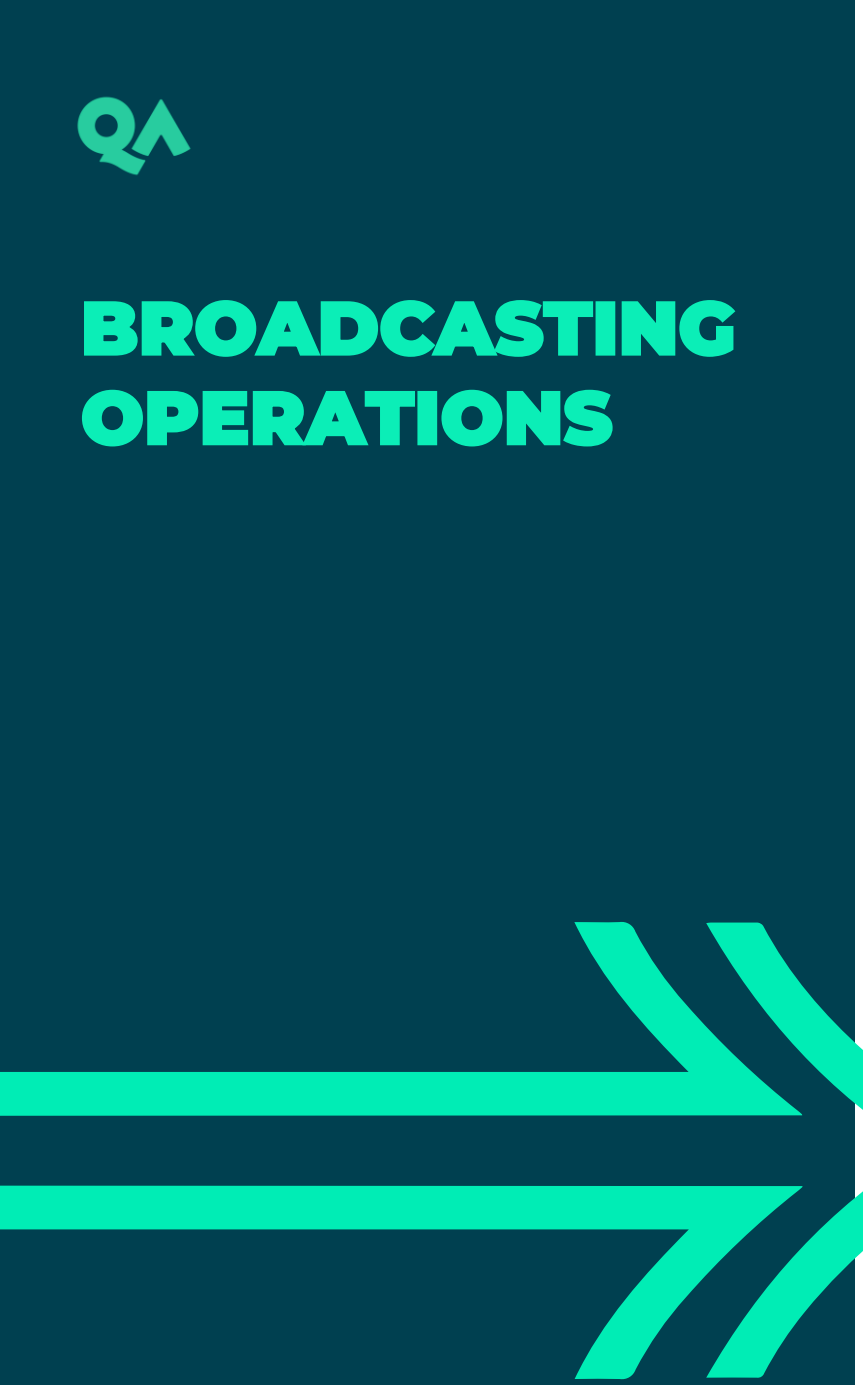
```
import numpy as np
```

ND-ARRAYS

```
payments = np.array([6.99, 12.40, 75.00, 1.55])
```

ND-arrays stands for **N-dimensional arrays**

- The basic data type in NumPy, intended to replace Python's list.
- Can be created from Python's list using **numpy.array()**.
- nd-arrays are **mutable**.
- **numpy.arange()** produces a sequence of numbers contained in an array.



BROADCASTING OPERATIONS

```
payments = np.array([6.99, 12.40, 75.00, 1.55])
transaction_fee = 1.00

payments - transaction_fee
```

```
array([ 5.99, 11.4 , 74.  ,  0.55])
```

```
payments = np.array([6.99, 12.40, 75.00, 1.55])
vat = 1.20

payments * vat
```

```
array([ 8.388, 14.88 , 90.  ,  1.86 ])
```

BROADCASTING OPERATIONS

Elementwise operators

- ~ NOT
- & AND
- | OR
- ^ XOR

```
payments = np.array([6.99, 12.40, 75.00, 1.55])

payments > 5.00
```

```
array([ True,  True,  True, False])
```

```
payments = np.array([6.99, 12.40, 75.00, 1.55])

(payments > 5.00) & (payments < 50.00)
```

```
array([ True,  True, False, False])
```

UNIVERSAL FUNCTIONS

A universal function, or ufunc, is a function that performs element-wise operations on data in ndarrays.

They are fast!

numpy.sqrt()
numpy.square()
numpy.exp()
numpy.log()
numpy.sign()
numpy.isnan()
numpy.sin()
numpy.add()

```
np.sign(payments)
```

```
array([1., 1., 1., 1.])
```

Why Pandas?

WHAT IS PANDAS

- Pandas is Python's ETL package for structured data.
- Built on top of NumPy, designed to mimic the functionality of R data frames.
- Provides a convenient way to handle tabular data.
- Can perform all SQL functionalities, including group-by and join.
- Compatible with many other Data Science packages, including visualisation packages such as Matplotlib and Seaborn.
- Defines two main data types:
    - pandas.Series
    - pandas.DataFrame

CREATING SERIES

A **Pandas series** can be created from a Python list or a NumPy array:

```
import pandas as pd

X = [1, 3, 5, 7]
mySeries = pd.Series(X)
print(mySeries)
```

```
0    1
1    3
2    5
3    7
dtype: int64
```

Index array

Values array

The index starts from 0 and increments by 1 for each subsequent element in the series.

The index is used to access the corresponding value.

```
print(mySeries[1])
```

```
3
```

CREATING DATA FRAMES

- Creating from Python lists, or NumPy arrays:

```
data = {
    "age": [34, 42, 27],
    "height": [1.78, 1.82, 1.75],
    "weight": [75, 80, 70]
}
df = pd.DataFrame(data)
print(df)
```

```
   age  height  weight
0   34    1.78      75
1   42    1.82      80
2   27    1.75      70
```

- Use a dictionary with column names as keys and a list of the row values.
- Creating from CSV files:

  pandas.read_csv(csv_file_name)
- The first row is used for column names.

COLUMN RETRIEVAL

Getting entire columns:

`my_dataframe[column_name]`

```
df['temp']
```

```
0    15.68
1    25.16
2    13.26
3    24.63
4    12.78
5    23.52
6    17.80
7    24.98
8    23.48
9    23.30
Name: temp, dtype: float64
```

```
df[['temp', 'humidity']]
```

	temp	humidity
0	15.68	73.18
1	25.16	83.88
2	13.26	80.05
3	24.63	82.37
4	12.78	83.10
5	23.52	85.35
6	17.80	85.64
7	24.98	76.81
8	23.48	80.86
9	23.30	79.96

SLICING DATAFRAMES

Getting individual elements from row and column IDs:

```
my_dataframe.loc[row_id, col_name]
my_dataframe.iloc[i, j]
```

```
df_1.loc["ind1","height"]
```

```
1.78
```

Row index “ind1”
Column “height”

```
df_1.iloc[0,1]
```

```
1.78
```

Row 0 Column 1

FILTERING

- Dataframes can be filtered row-wise using a sequence of Trues and Falses.
- These can be generated by queries.

```
df[(df["Income"] > 20000) & (df["Debt"] == 0)]
```

	ID	Income	Term	Balance	Debt	Score	Default
3	370	21600	Short Term	920	0	NaN	False
4	756	24300	Short Term	1260	0	495.0	False
6	373	20400	Short Term	1200	0	556.0	False
7	818	24600	Short Term	1470	0	301.0	False
9	621	25400	Short Term	1130	0	729.0	True
...	...	...	...	...	...	...	...
847	96	26300	Long Term	1760	0	489.0	False
848	762	29200	Long Term	1500	0	755.0	False
849	516	36200	Short Term	1510	0	812.0	False
850	627	27000	Short Term	1510	0	436.0	False
852	932	42500	Long Term	1550	0	779.0	False

299 rows × 7 columns

GROUP BY

GROUP BY and:

- Counting the number of rows in each group:

```
my_dataframe.groupby(criteria).size()
```

- Sum of every numerical column in each group:

```
my_dataframe.groupby(criteria).sum()
```

- Mean of every numerical column in each group:

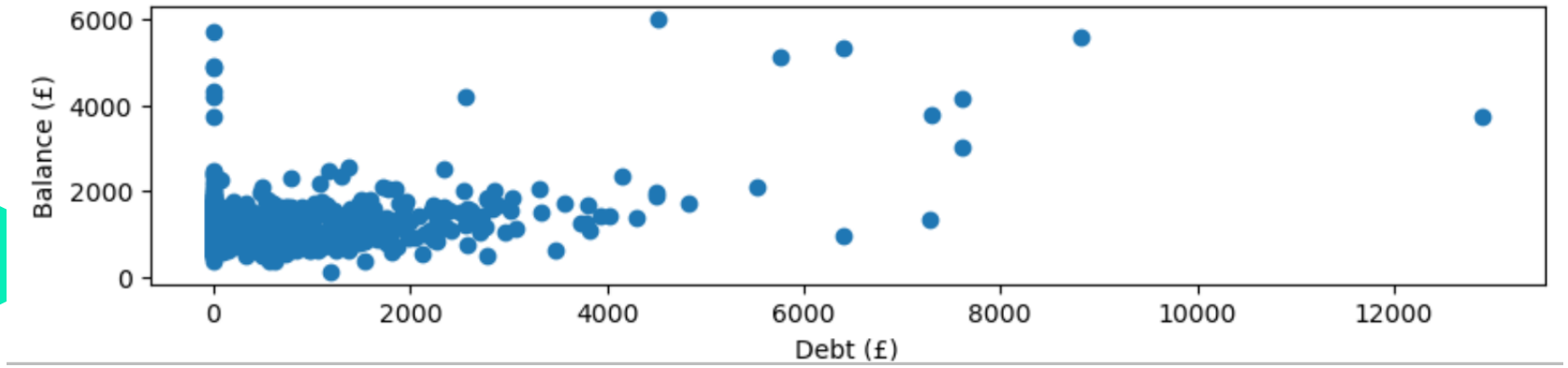
```
my_dataframe.groupby(criteria).mean()
```

Plotting

MATPLOTLIB

- Editing here is done using the MATLAB style

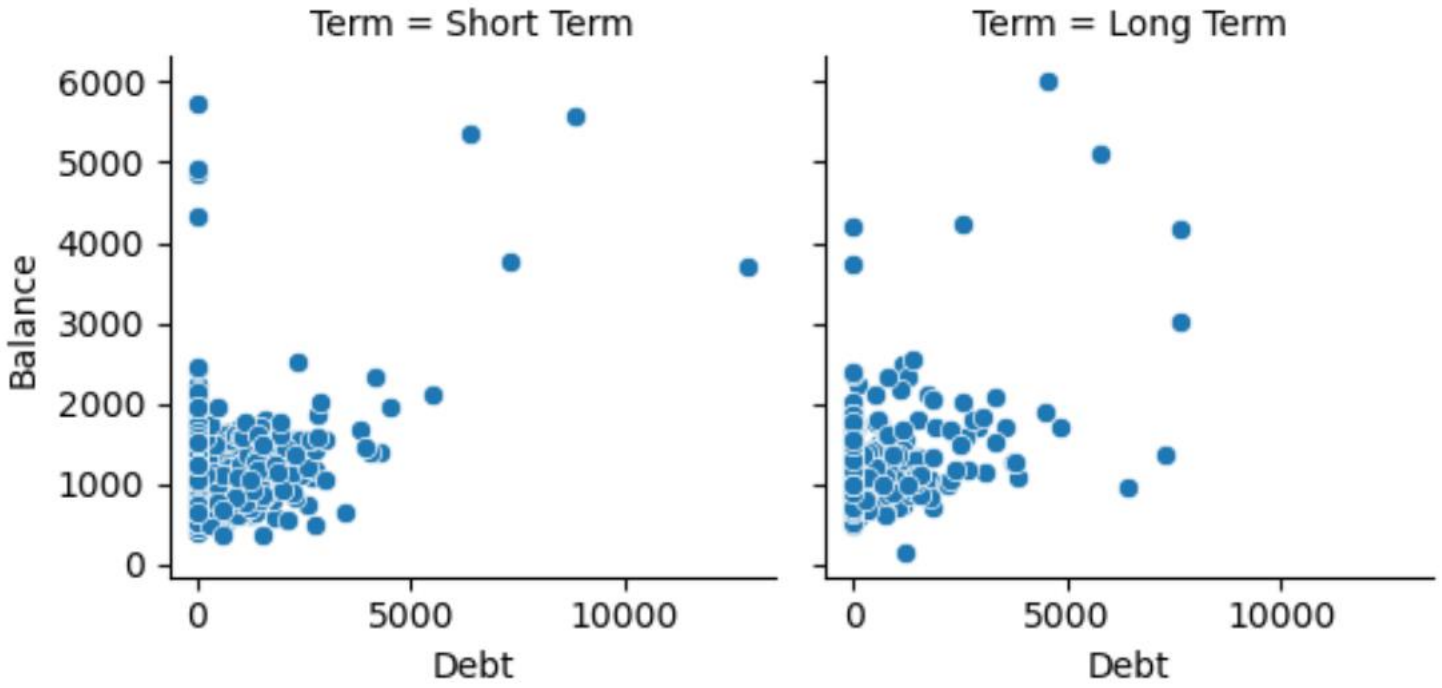
```
fig, ax = plt.subplots(figsize=(10, 2))

plt.scatter(df['Debt'],
            df['Balance'])

plt.xlabel('Debt (£)')
plt.ylabel('Balance (£)');
```

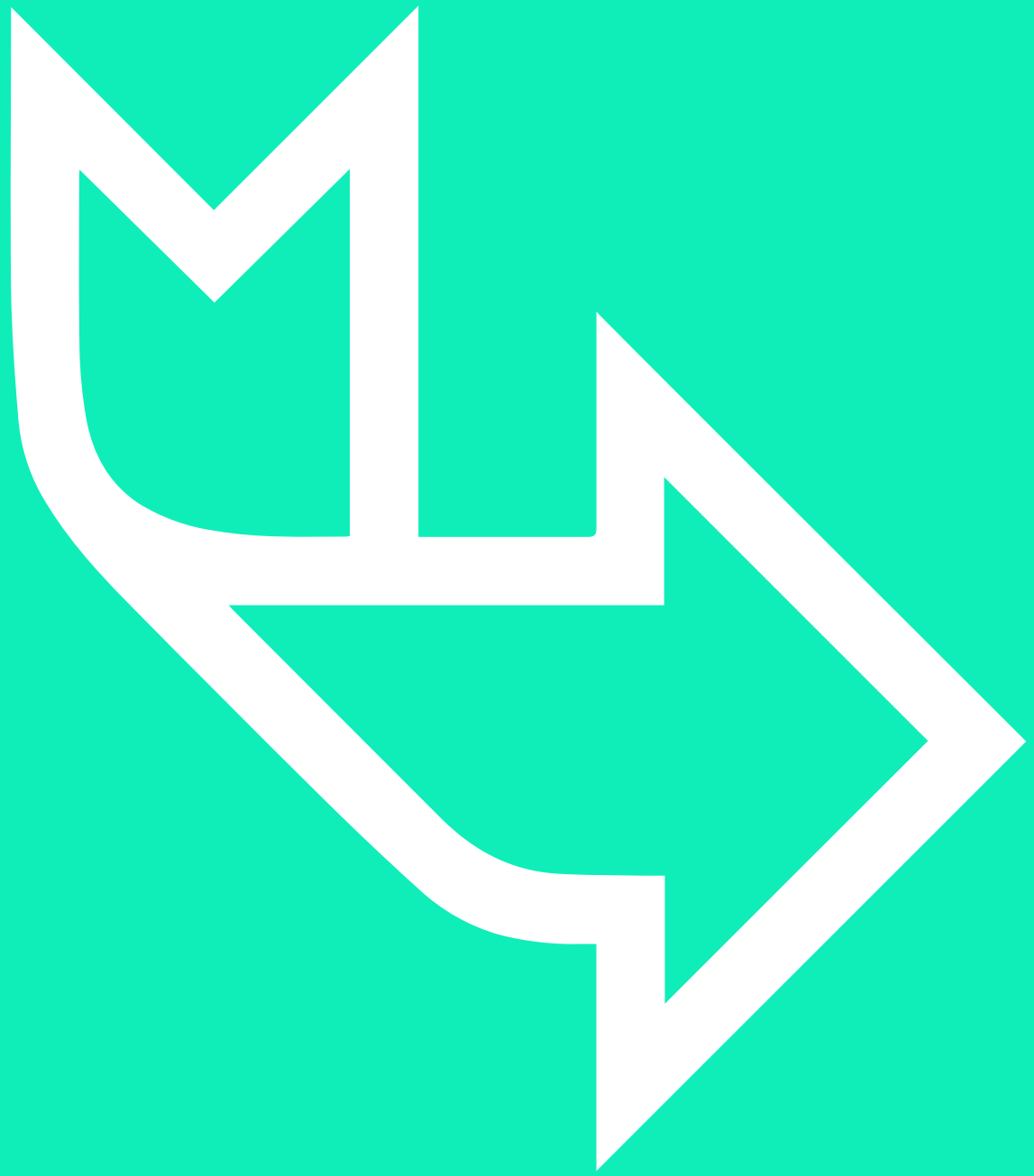


SEABORN

- Seaborn makes plotting easier.
  - It is still matplotlib under the hood.

```
sns.FacetGrid(data=df,
              col='Term').map(sns.scatterplot, 'Debt', 'Balance');
```





EXERCISE

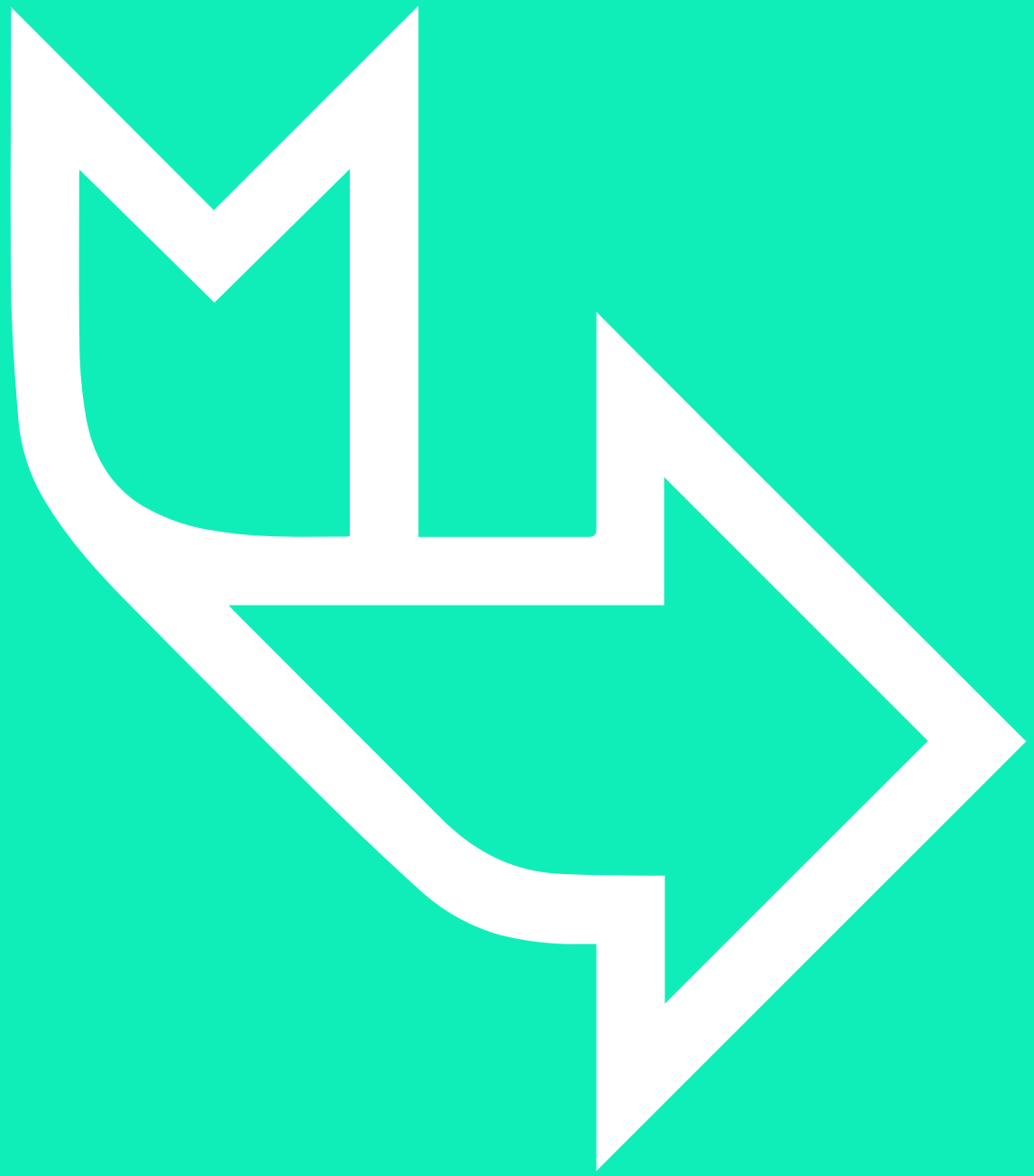
Work though Module 2 exercises:

Introduction to Python for Data Science

LEARNING CHECK

Think about your answers to these questions:

- Why are notebooks are often used in Data Science projects?
- Which structures do we use to hold data in Python?
- Which libraries are commonly used in Python? What do they do?
- What are virtual environments used for?

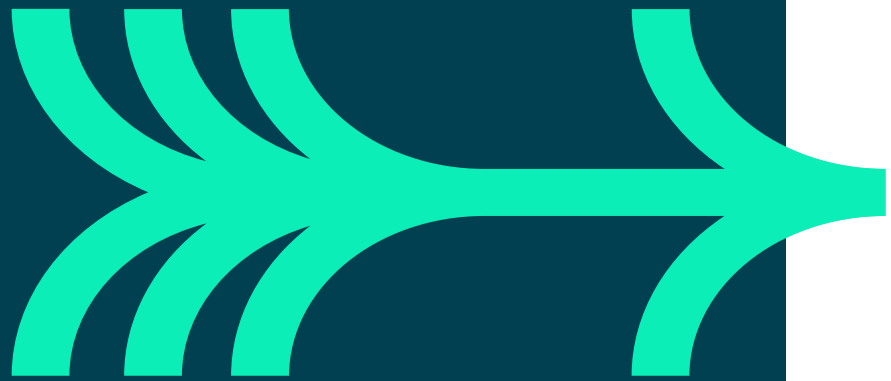


HOW DID YOU GET ON?

Learning objectives

- Understand why notebooks are often used in Data Science projects.
- Use Python and associated libraries to manipulate datasets.
- Describe why virtual environments are used.
- Visualise data using Python.

DESCRIPTIVE AND INFERENTIAL STATISTICS WITH PYTHON



Learning objectives

- Understand the role that descriptive and inferential statistics play in Data Science.
- Use measures of central tendency, variation, and correlation to understand data.
- Use hypothesis tests to establish the significance of effects.
- Use statistical visualisations to understand data distributions.
- Describe the role of Exploratory Data Analysis in a Data Science project.

Expected prior knowledge

- Nothing is assumed about your background.

Descriptive statistics

MEASURES OF CENTRAL TENDENCY (AVERAGES!)

e.g.:
2, 5, 4, 5, 3

Mean

$\bar{x}$ = (2+5+4+5+3)/5 = 3.8

Mode

Frequency	
2:	1
3:	1
4:	1
5:	2

Median

2, 3, 4, 5, 5

(2+5)/2 = 3.5

MEASURES OF CENTRAL TENDENCY: PYTHON

```python
import pandas as pd

data = pd.Series([2, 5, 4, 5, 3])

mean = data.mean()
mode = data.mode().iloc[0]
median = data.median()
mid_range = (data.min() + data.max()) / 2

print("Mean:", mean)
print("Mode:", mode)
print("Median:", median)
print("Mid-Range:", mid_range)
```

```
Mean: 3.8
Mode: 5
Median: 4.0
Mid-Range: 3.5
```

MEASURES OF VARIATION: RANGE AND IQR

e.g.:
2, 3, 4, 6, 7, 7, 8, 19

Range:

Width of spread for the whole data set (largest – smallest).

19 – 2 = 17

Inter-Quartile Range:

Width of spread for the middle half of the data set.

Similar to finding the median, we find the first (or lower) quartile and the third (or upper) quartile before finding the difference.

IQR = Q3 – Q1
= 7.5 – 3.5
= 4

MEASURES OF VARIATION: STANDARD DEVIATION AND VARIANCE (POPULATION)

e.g.:
2, 3, 4, 6, 7, 7, 8, 19

$x_i - \bar{x}$ is the distance from each piece of data to the mean.

Some distances are positive, and some are negative. Before calculating the average distance of the data from the mean we need to deal with the signs.

For standard deviation and variance, this is done by squaring each of the distances first.

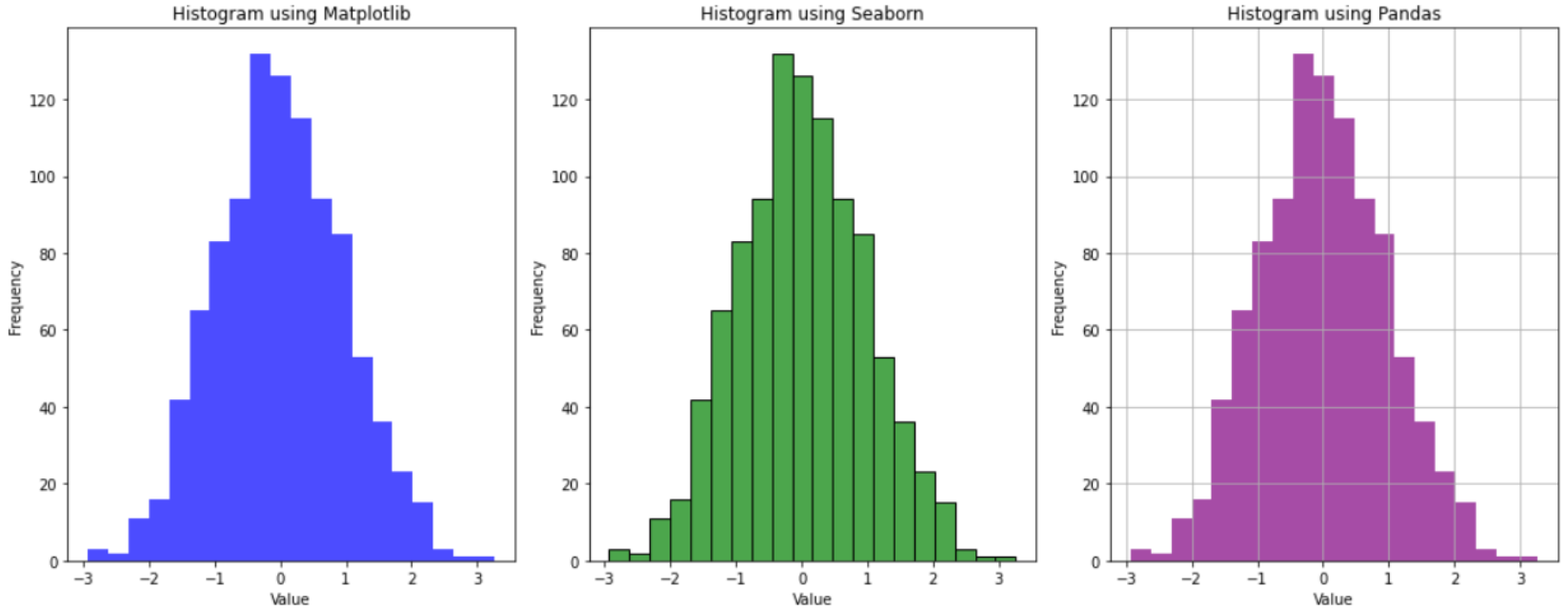
Variance, s^2, is the average of these squared distances from the mean. Standard Deviation, s, is the result of completing the calculation by square rooting.

$$s^2 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})^2}{n} = 24.5$$

$$s = 4.95 \text{ (3 s.f.)}$$

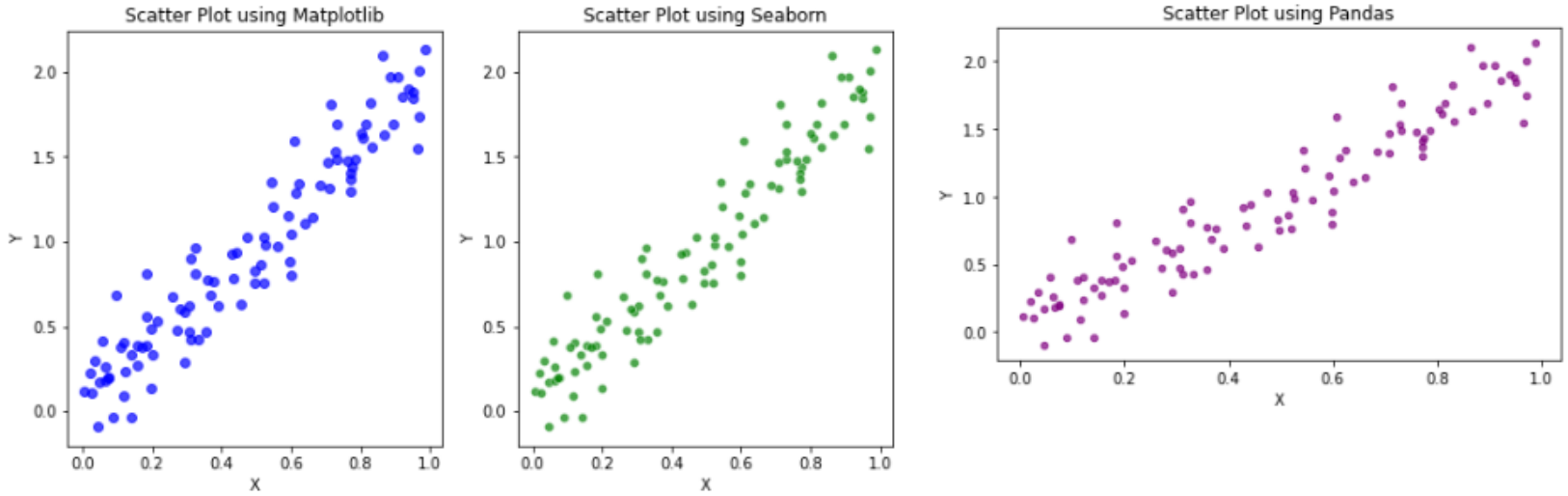
Descriptive visualisations

HISTOGRAMS: PYTHON




```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Create a sample dataset
data = np.random.randn(1000)  # Generating random data

# Creating a Pandas DataFrame
df = pd.DataFrame({'data': data})

plt.hist(df['data'], bins=20, color='blue', alpha=0.7)

sns.histplot(data=df, x='data', bins=20, color='green', alpha=0.7)

df['data'].hist(bins=20, color='purple', alpha=0.7)
```

SCATTER PLOTS: PYTHON




```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Create a sample dataset
np.random.seed(42)
n_points = 100
x = np.random.rand(n_points)
y = 2 * x + np.random.normal(0, 0.2, n_points)

# Creating a Pandas DataFrame
df = pd.DataFrame({'X': x, 'Y': y})
```

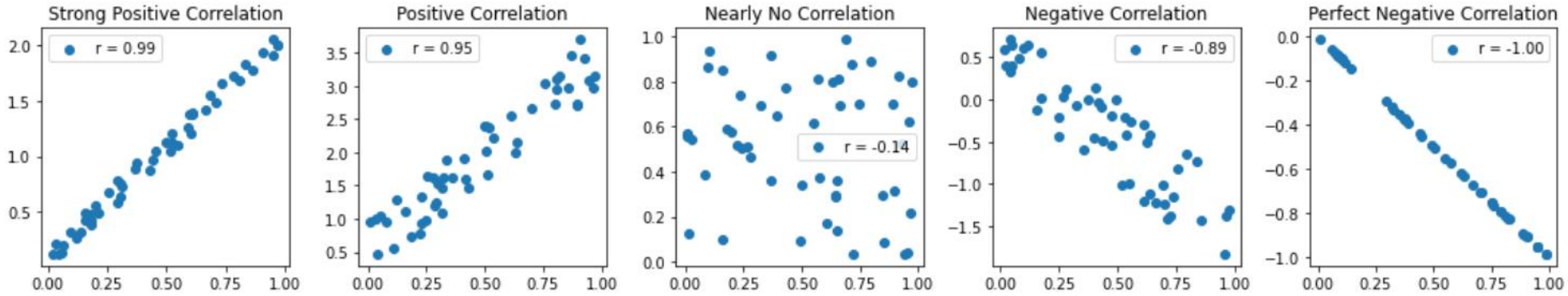
```python
plt.scatter(df['X'], df['Y'], color='blue', alpha=0.7)
```

```python
sns.scatterplot(data=df, x='X', y='Y', color='green', alpha=0.7)
```

```python
df.plot.scatter(x='X', y='Y', color='purple', alpha=0.7)
```

Correlation from scatter plots

Checking what a scatter plot looks like is another way of checking for outliers or anomalies, but we also might be looking to see if a line of best fit (or regression line) might be appropriate.



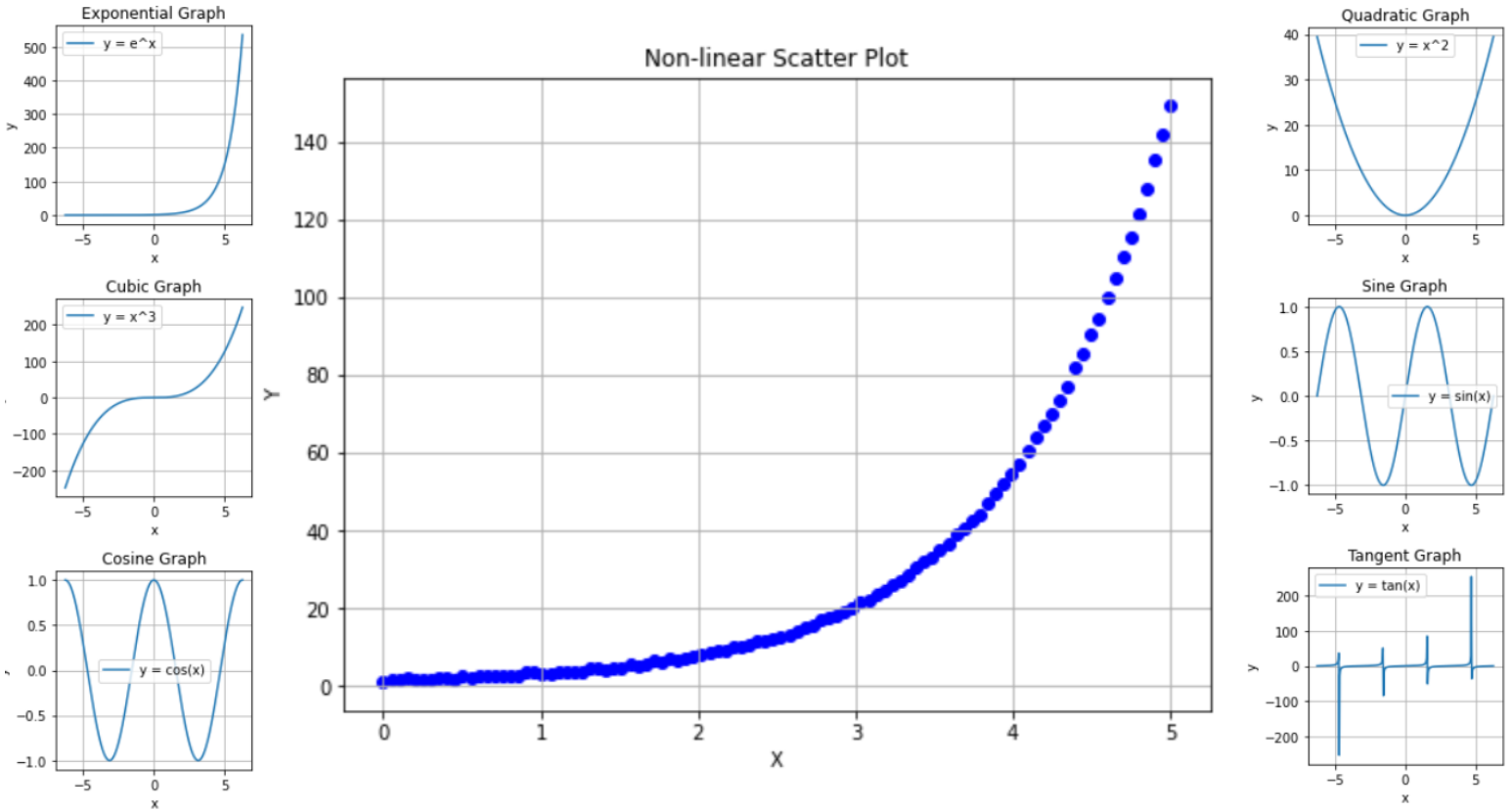
The more squashed the ‘ball’ of data is, the stronger the correlation.

Correlation or association can tell us if two data features are linked, but one hasn’t necessarily caused the other – there may be a third unseen feature that is a **causal link**.

THE PROBLEM WITH CURVES

If the shape of the data looks curved, there are an infinite number of equations that could be fitted!

This one appears exponential at first glance; however, it could be part of many other types of curve that might be stretched and shifted to fit the data.



A straight line, on the other hand, has only one equation!

Distributions

DISTRIBUTION

A **Distribution** is the location the values in a column of data on a number line in relation to the other values – what shape does it make?

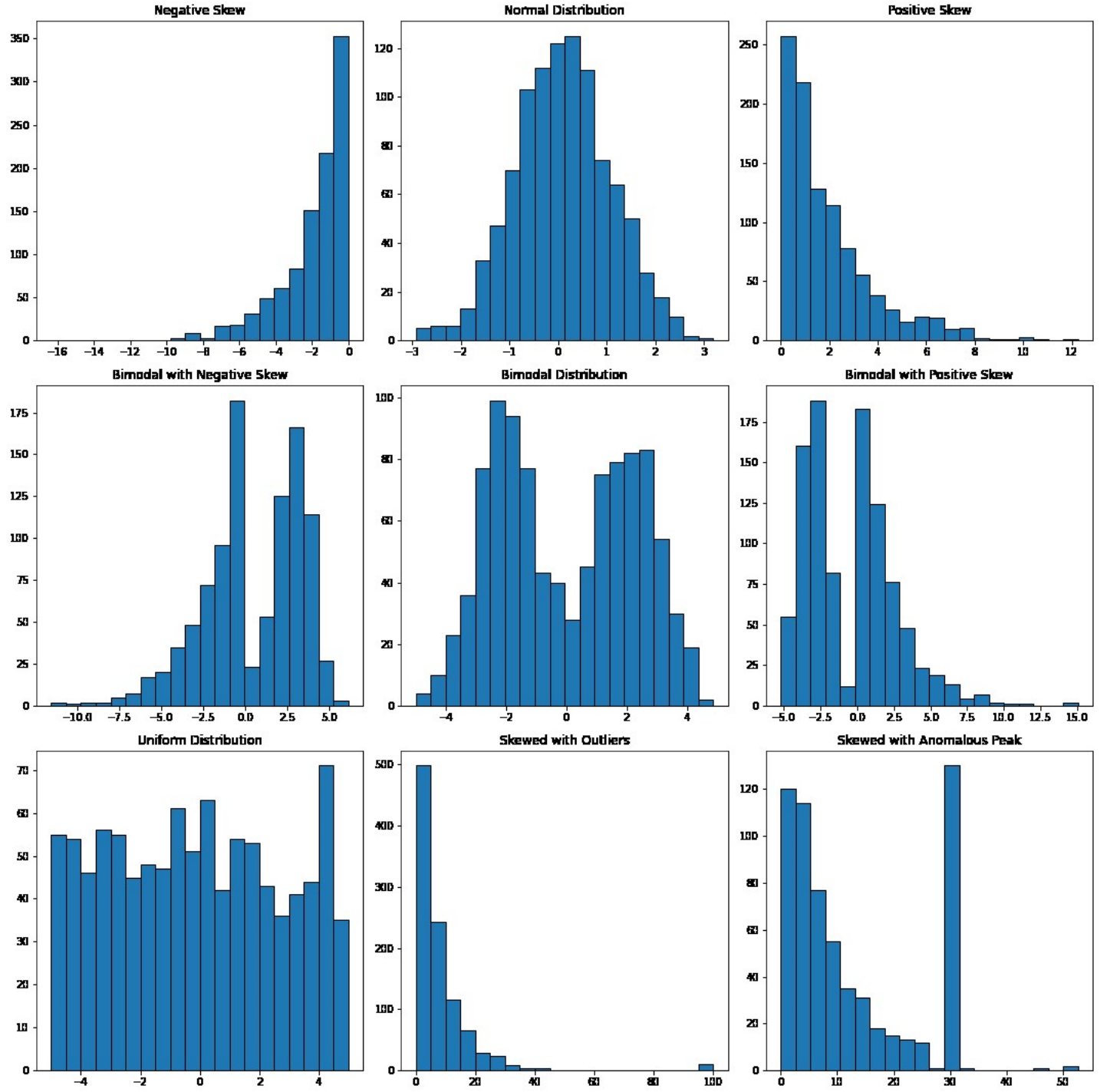
You might also hear the phrase ‘the **structure** of the data’, particularly when we begin talking about more than one column (with a slightly different meaning to **structured data**).

To see the distribution or structure of the data, we don’t really need to know the scale.

We can see the distribution using a **histogram** to display:

- raw or scaled data.
- probability of each value in the dataset.

DISTRIBUTIONS: RAW DATA USING HISTOGRAMS



Inferential statistics

STATISTICAL INFERENCE: STEPS TO TAKE WHEN TESTING HYPOTHESES

Summary of steps we will take:

- Define any notation or abbreviations.
- State the null and alternative hypotheses – how you think the **population** behaves: H_0 and H_1 with the value(s) of any parameters and a descriptive sentence to help you interpret the results later. Identify if the test is one or two tailed.
- Decide the **significance level**, α.
- State that you are **assuming the null hypothesis is true** and state any logical deductions from this (e.g., the assumed distribution of the population).
- Use a sample to calculate a **test statistic or p-value** based on the assumption that the null hypothesis is true.
- Check **if** the test statistic is extreme or if the p-value is low – if either is the case you can **reject the assumption** that H_0 is true; the result from the sample is **significant** and conclude (in context) that there **might** be enough evidence to suggest the alternative is true.

SIGNIFICANCE LEVEL

The **significance level**, α.

How unlikely does something have to be before we start to wonder if we have been making bad assumptions?

We can't set it at 0% or we would ***always*** stick with our initial assumption and there would be no point doing the test!

But it also represents the chance of switching to an alternative hypothesis incorrectly – it's the chance of a **false positive**.

The level should be decided **before** calculations on the sample – to avoid scientific bias.

T-TEST: AN EXAMPLE OF A/B TESTING

Basic idea: are the means of two samples similar or different? If they are similar, they could come from the same population.

Examples:

- Is the average sentiment value from before and after a marketing campaign the same or have we made a positive impression?

Note: This is an example of **1 tailed,** specifically 'have we made a positive impact' rather than 'is it different', which would be **2 tailed**.

- Is the average investment performance different if we change strategy?

A/B testing can be used to compare before vs after, or distinct groups in the same time frame.

T-TEST EXAMPLE SET UP

Group A: all the investments follow the standard strategy

Group B: all the investments use the new strategy

H_0: The population means of group A and B are the same – i.e., it makes no difference which strategy is used.

H_1: The population means of group A and B are different – i.e., it does make a difference which strategy is used.

This is 2 tailed.

Significance level: 5% = 0.05. As this is 2 tailed, the significance level is split when comparing to the p-value later: 0.025

Assume the null hypothesis is true – this is done automatically by the tool or programming language you will use.

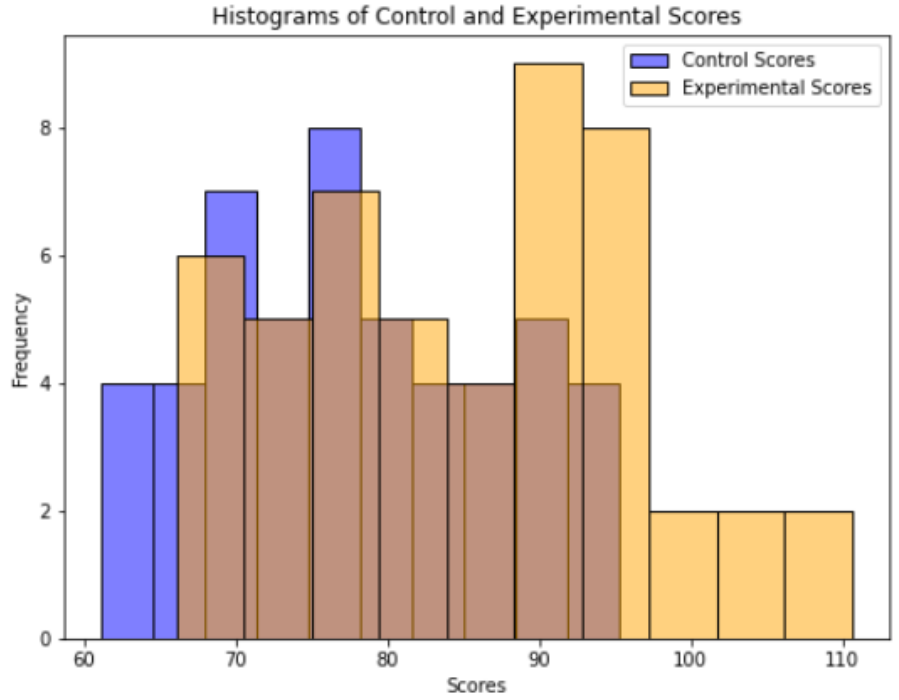
T-TEST VISUALISED AND IN CODE

Are the these different enough or could they have come from the same population?

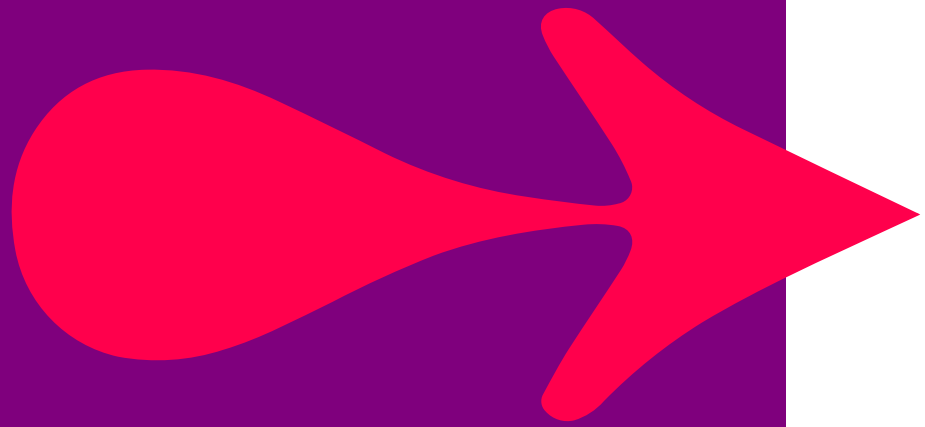
p-value: What's the chance the samples behave this way if the assumption is correct (that they come from the same population).

We can use a package like scipy.stats to calculate the p-value.

Underlying code often uses the normal distribution to calculate probabilities and work out overlapping areas.



CONDUCTING A T-TEST IN PYTHON



- Import pandas as pd and from scipy import stats.
- Use pd.read_csv() to save the data in investment_performance.csv as df, a pandas dataframe.
- Use stats.ttest_ind() with each of the two columns from the dataframe addressed in the brackets.
- Compare the p-value to half the significance level (since it's a two tailed test).
- Conclude: What to do about the assumption that was made? (That the null hypothesis is true).
- Conclude in context: What does that mean we have evidence for in this particular context?

Exploratory data analysis

EDA TASKS (NOT AN EXHAUSTIVE LIST!)

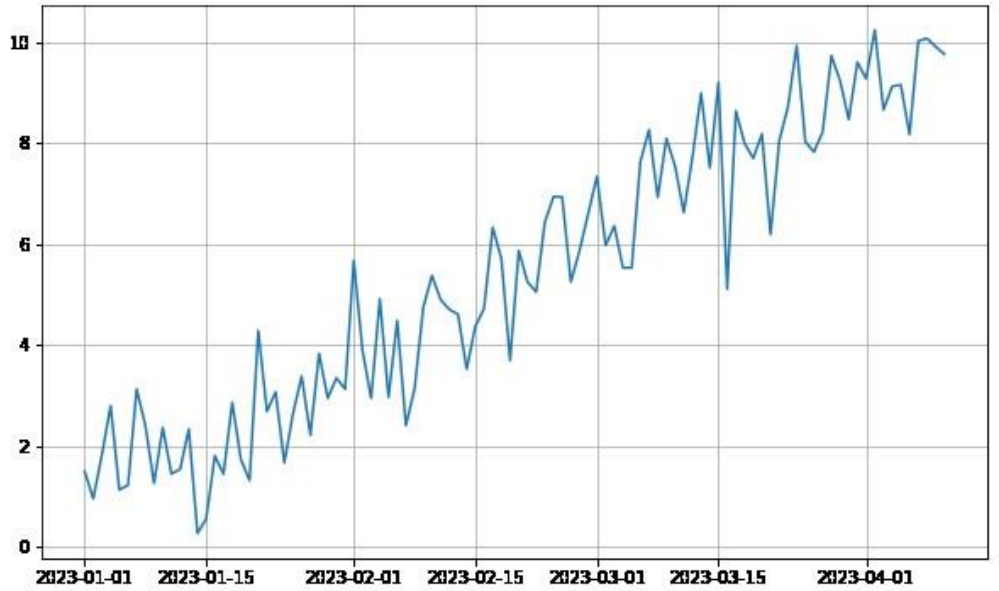
Categorical and Qualitative Data

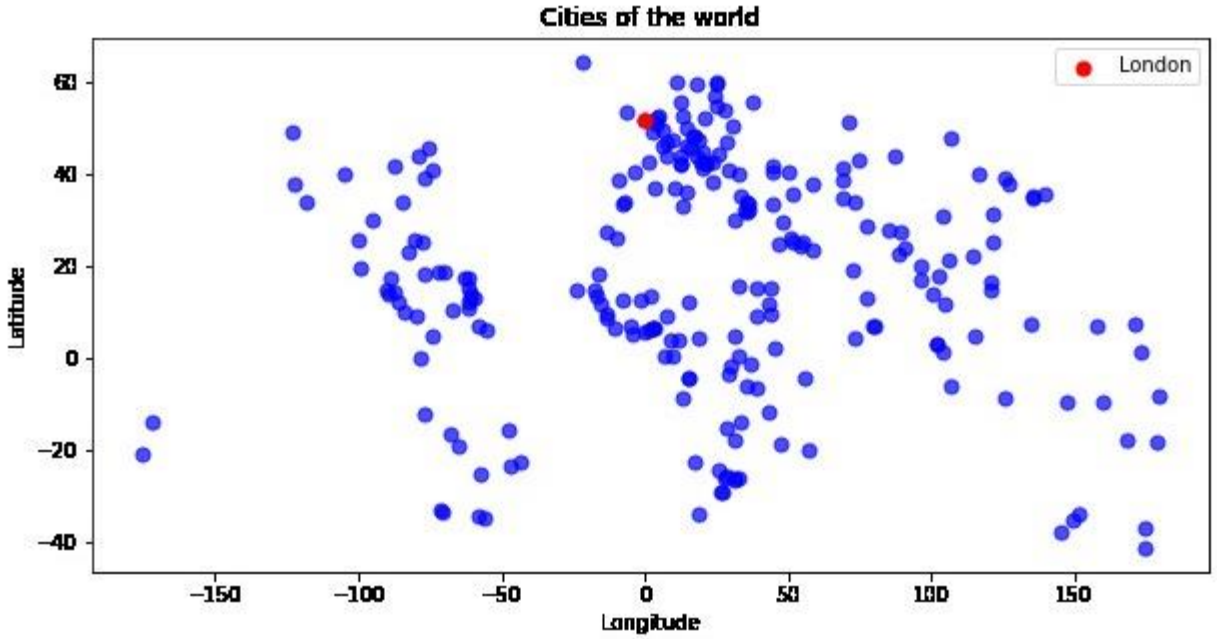
- Proportion of the data within each category
- Tests can be done to check that proportions in the data are representative of the population once a model is deployed

Numerical and Quantitative Data

- Summaries: Typical value (averages), Variation from the typical value, and the shape and structure of the data.
- Identify outliers, anomalies, and structures/clusters for separate investigations
- Tests for distribution types which might impact on model choices
- Tests for connections between data features (correlation or association)
- Tests for differences between sources at different locations or the same source over time

DIFFERENCES OVER TIME OR BETWEEN SECTIONS: VISUALLY





DIFFERENCES OVER SPACE/TIME:

A/B TESTING

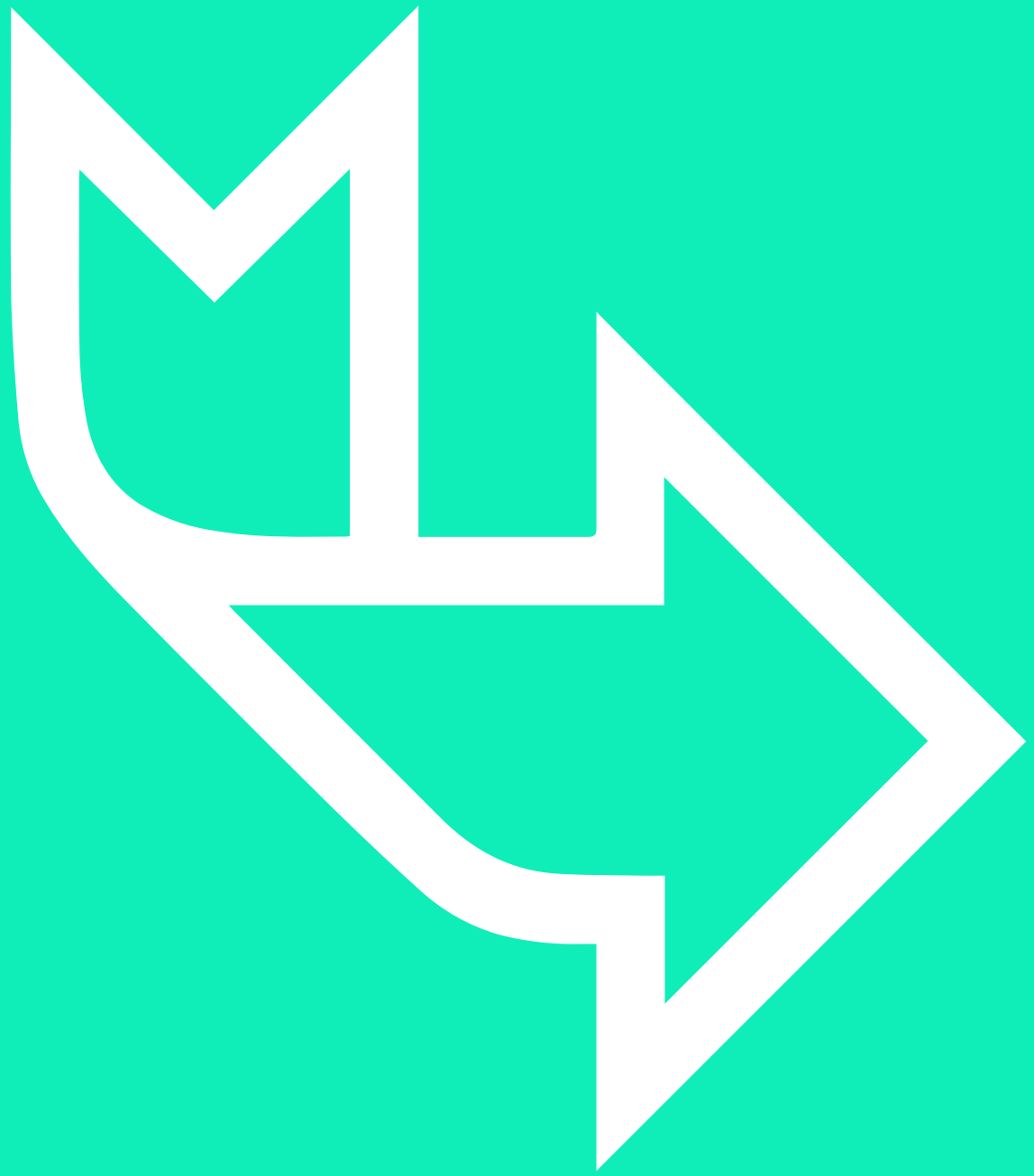
Key question: Are two or more groups different?

We can compare their averages to see if they are quite similar or significantly different.

Space: Compare the average sales between two stores in the same time frame.

Time: Compare the average customer rating before and after an advertising campaign.

Example: T-tests.



Activity

Using the Iris data set, perform a simple Exploratory Data Analysis:

- **Read in the csv and save it as df.**
- **Mean and median.**
- **Calculate correlation coefficients.**

QA

SOLUTIONS

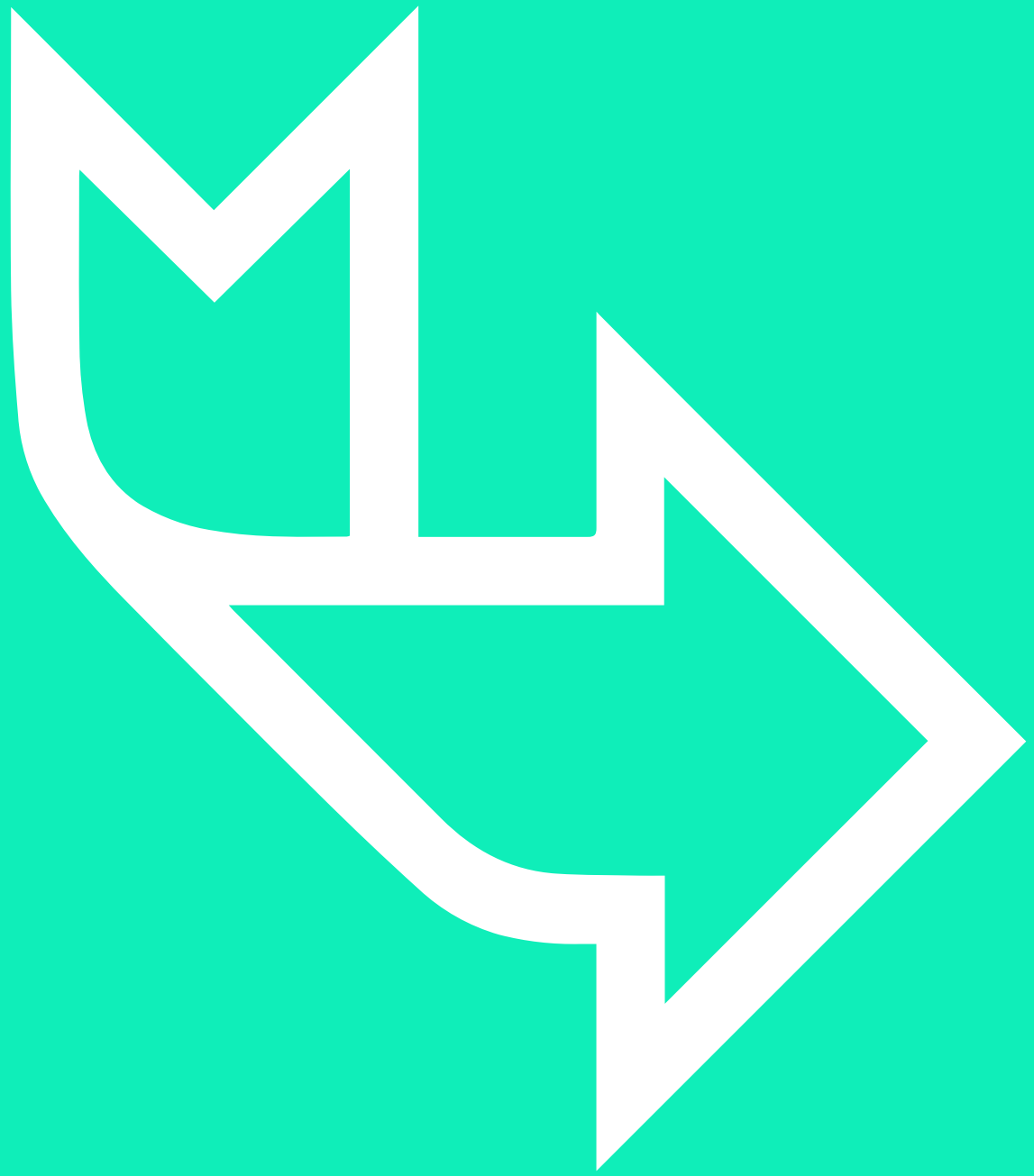
PYTHON

```
import pandas as pd

df = pd.read_csv('iris_dataset.csv')

df.describe()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.057333	3.758000	1.199333	1.000000
std	0.828066	0.435866	1.765298	0.762238	0.819232
min	4.300000	2.000000	1.000000	0.100000	0.000000
25%	5.100000	2.800000	1.600000	0.300000	0.000000
50%	5.800000	3.000000	4.350000	1.300000	1.000000
75%	6.400000	3.300000	5.100000	1.800000	2.000000
max	7.900000	4.400000	6.900000	2.500000	2.000000

```
df.corr()
```

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
sepal length (cm)	1.000000	-0.117570	0.871754	0.817941	0.782561
sepal width (cm)	-0.117570	1.000000	-0.428440	-0.366126	-0.426658
petal length (cm)	0.871754	-0.428440	1.000000	0.962865	0.949035
petal width (cm)	0.817941	-0.366126	0.962865	1.000000	0.956547
target	0.782561	-0.426658	0.949035	0.956547	1.000000



Discussion:

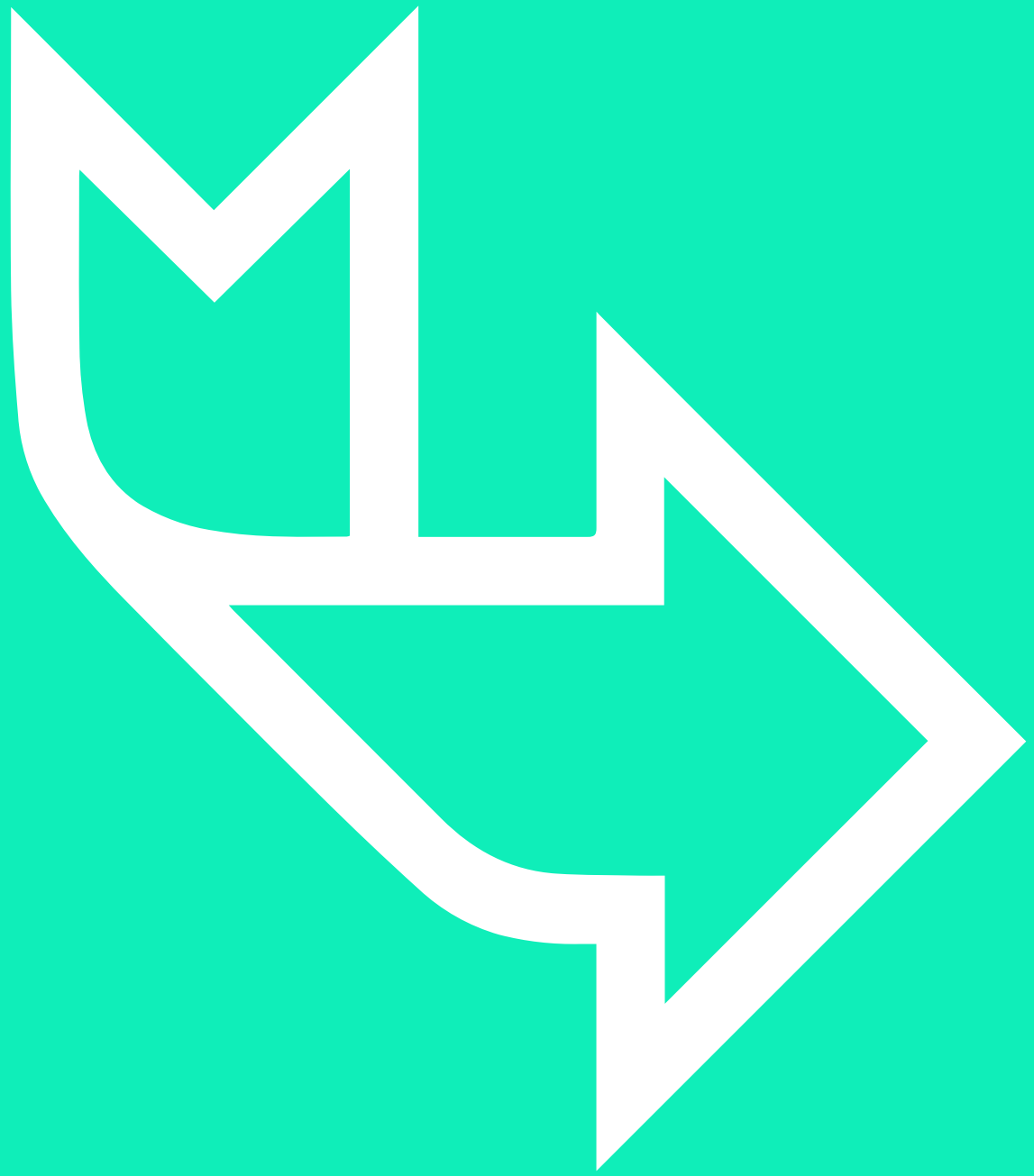
What does our analysis tell us?

What sort of model could be created and who might find it useful?

LEARNING CHECK

Think about your answers to these questions:

- What role do descriptive and inferential statistics play in Data Science?
- What do measures of central tendency, variation, and correlation tell us about our data?
- How can we use hypothesis tests?
- Which statistical visualisations could we use to understand data distributions?
- What is the the role of Exploratory Data Analysis in a Data Science project?



HOW DID YOU GET ON?

Learning objectives

• Understand the role that descriptive and inferential statistics play in Data Science.

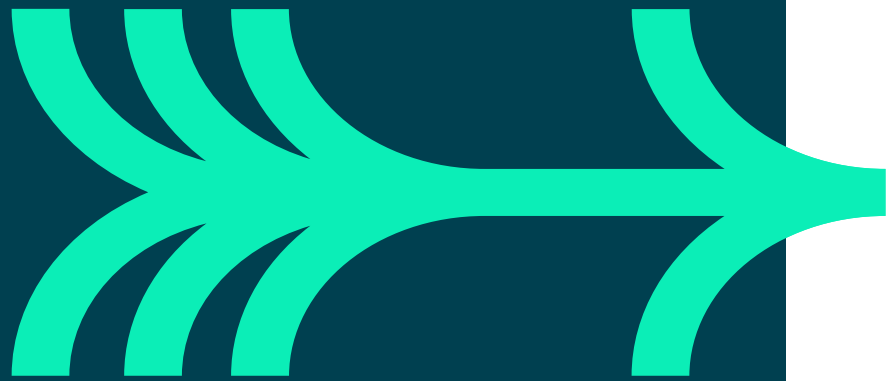
• Use measures of central tendency, variation, and correlation to understand data.

• Use hypothesis tests to establish the significance of effects.

• Use statistical visualisations to understand data distributions.

•Describe the role of Exploratory Data Analysis in a Data Science project.

PREPROCESSING DATA FOR ANALYSIS



Learning objectives

- Appropriately process duplicated data, missing values, and outliers.
- Understand the importance of scaling, encoding, and feature selection.
- Describe the importance of training, testing, and validation sets.
- Know which engineer novel features to analyse.

Expected prior knowledge

- Nothing is assumed about your background.

PRE-PROCESSING

Duplicates
Data sets can contain records which have been duplicated for one reason or another.

Null Values
Missing or NULL values can be problematic, depending on the situation.

Outliers
Outliers within your data set can skew predictions, leading to an inaccurate or unreliable model.

PRE-PROCESSING

Scaling
Most datasets contain multiple numeric features expressed in different units of measurement.

Encoding
Models require data to be encoded numerically, and there is more than one way of doing this.

Feature selection and engineering
Ideally, we want to use as few features to yield as much information as possible. This can be achieved by selecting, or creating, better ones.

Duplicates

Duplicates

```
# Oversampled DF
X_train2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1062 entries, 0 to 1061
Data columns (total 7 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   Income           1062 non-null   float64
 1   Term_Long Term   1062 non-null   int64  
 2   Term_Short Term  1062 non-null   int64  
 3   Balance          1062 non-null   float64
 4   Debt             1062 non-null   float64
 5   Score            1062 non-null   float64
 6   Default          1062 non-null   int64  
dtypes: float64(4), int64(3)
memory usage: 58.2 KB
```

```
# Oversampled DF
X_train2.duplicated()
```

```
]: 0       False
   1       False
   2       False
   3       False
   4       False
           ...
   1057     True
   1058     True
   1059     True
   1060     True
   1061     True
   Length: 1062, dtype: bool
```

```
# Oversampled DF
X_train2.duplicated().sum()
```

```
20]: 477
```

Identifying duplicates

```
:
1 #identify duplicate rows
2 X_train2[X_train2.duplicated()]
3
```

[21]:

	Income	Term_Long Term	Term_Short Term	Balance	Debt	Score	Default
585	19900.0	1	0	1560.0	0.0	273.0	1
586	17500.0	0	1	1040.0	1695.0	89.0	1
587	21100.0	1	0	610.0	0.0	196.0	1
588	14600.0	0	1	1470.0	1345.0	97.0	1
589	21500.0	0	1	810.0	1752.0	138.0	1
...	...	...	...	...	...	...	...
1057	23400.0	0	1	1050.0	0.0	115.0	1
1058	18600.0	0	1	890.0	0.0	291.0	1
1059	21300.0	1	0	1020.0	2156.0	83.0	1
1060	11800.0	1	0	1070.0	0.0	193.0	1
1061	18100.0	1	0	1180.0	0.0	407.0	1

477 rows × 7 columns

Removing duplicates

```
1  # Drop duplicate rows
2  X_train2.drop_duplicates()
3
```

]:

	Income	Term_Long Term	Term_Short Term	Balance	Debt	Score	Default
0	45300.0	1	0	1430.0	422.0	646.0	0
1	25800.0	1	0	510.0	0.0	337.0	0
2	23500.0	0	1	650.0	0.0	348.0	0
3	20100.0	1	0	650.0	0.0	216.0	0
4	22900.0	1	0	910.0	284.0	256.0	0
...	...	...	...	...	...	...	...
580	58500.0	1	0	1710.0	4835.0	804.0	0
581	76500.0	0	1	4860.0	0.0	1000.0	0
582	24200.0	0	1	1160.0	0.0	341.0	0
583	21500.0	0	1	810.0	1752.0	138.0	1
584	23700.0	0	1	880.0	703.0	316.0	1

585 rows × 7 columns

Missing data

MISSING DATA

How do we deal with missing data?
There are three main options:

1) Removal.

2) Imputation – requires skill.

3) Leave as is; some models can deal with missing values.

TYPES OF MISSING DATA

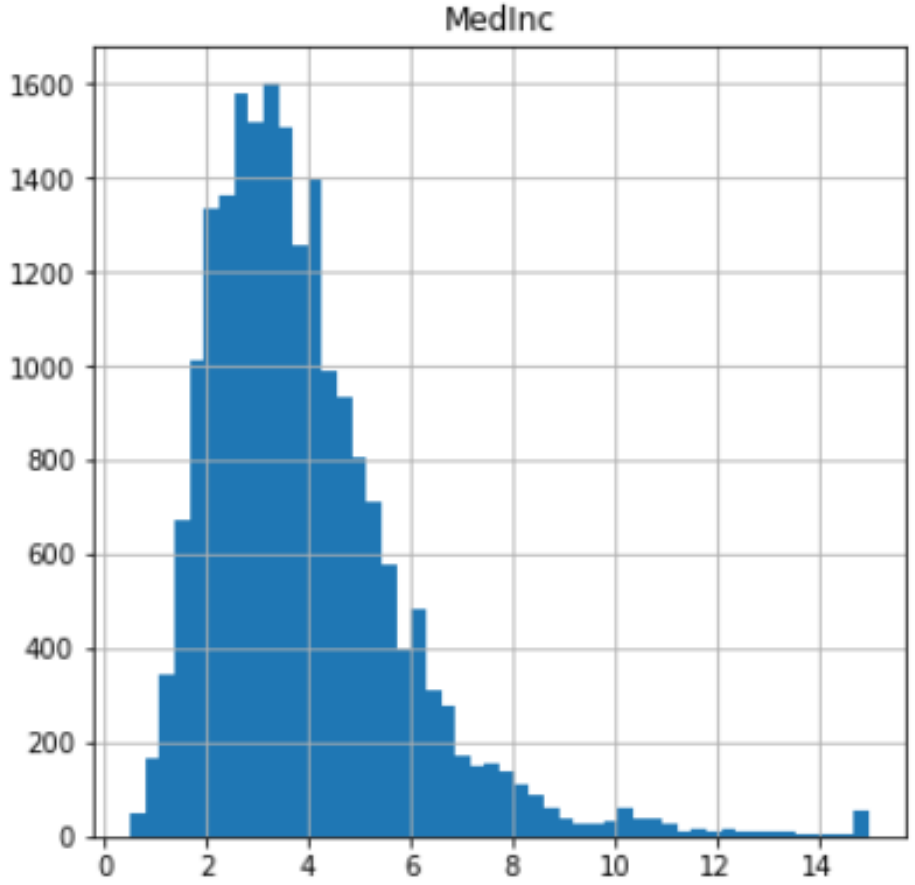
Missing completely at random (MCAR)
Data are missing independently of both observed and unobserved data.

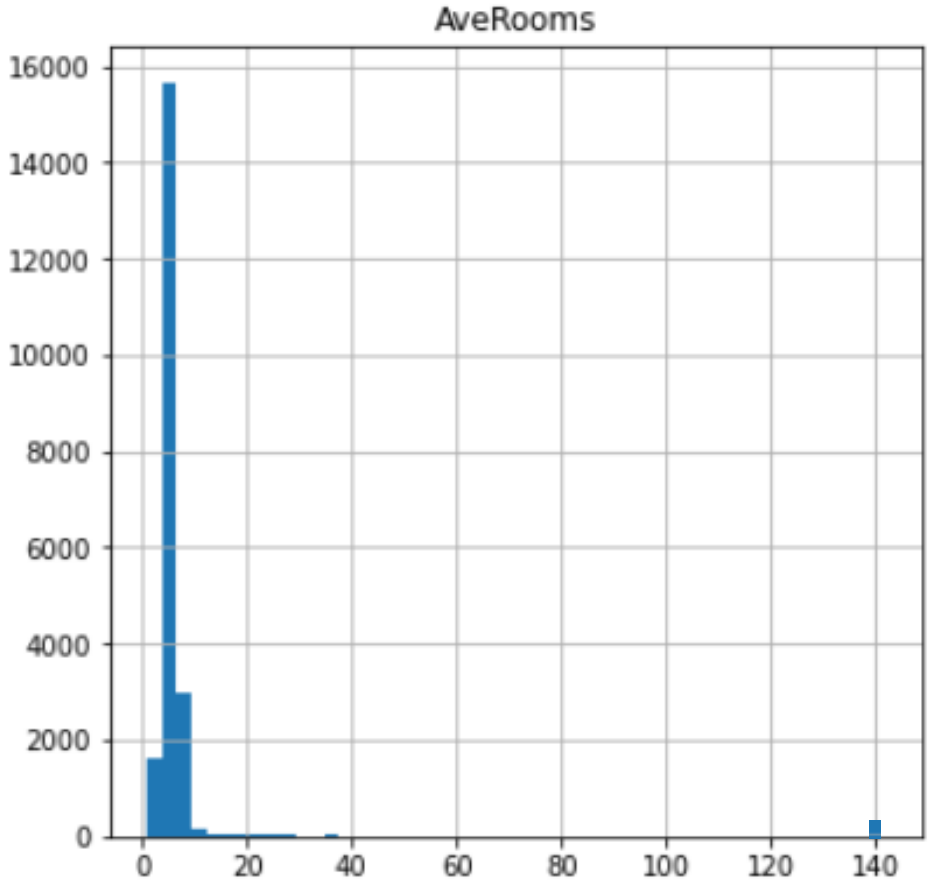
Missing at random (MAR)
Given the observed data, data are missing independently of unobserved data.

Missing Not at Random (MNAR)
Missing observations related to values of unobserved data.

Outliers

What are outliers?





OUTLIERS: CALCULATED

e.g.:
2, 3, 4, 6, 7, 7, 8, 19

These two methods are designed to attempt to create boundaries which capture around 95% of the data typically – leaving just 5% plus outliers to be investigated if your data was normally distributed.

$$\bar{x} \pm 2s$$

= -3.58, 17.6 => outlier: 19

$$Q_1 - 1.5IQR$$
$$Q_3 + 1.5IQR$$

= -2.5, 13.5 => outlier: 19

Either could be used to automatically detect or highlight outliers.

OUTLIERS: PYTHON

```python
import numpy as np

data = [2, 3, 4, 6, 7, 7, 8, 19]

# Method 1: Mean plus or minus 2 standard deviations
mean = np.mean(data)
stddev = np.std(data)

lower_bound_method1 = mean - 2 * stddev
upper_bound_method1 = mean + 2 * stddev

outliers_method1 = [x for x in data if x < lower_bound_method1 or x > upper_bound_method1]

# Method 2: Q3 + 1.5 * IQR and Q1 - 1.5 * IQR
q1 = np.percentile(data, 25)
q3 = np.percentile(data, 75)
iqr = q3 - q1

lower_bound_method2 = q1 - 1.5 * iqr
upper_bound_method2 = q3 + 1.5 * iqr

outliers_method2 = [x for x in data if x < lower_bound_method2 or x > upper_bound_method2]

print("Data:", data)
print("Method 1 Outliers:", outliers_method1)
print("Method 2 Outliers:", outliers_method2)
```

```
Data: [2, 3, 4, 6, 7, 7, 8, 19]
Method 1 Outliers: [19]
Method 2 Outliers: [19]
```

Scaling

SCALING

To minimise the effect of magnitude, we often scale data. Two common approaches are:

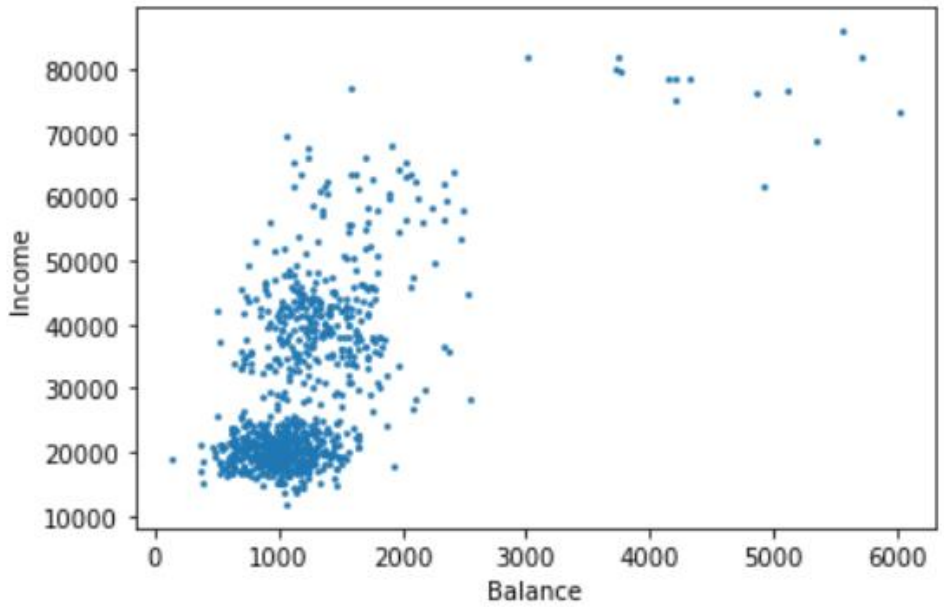
Normalisation

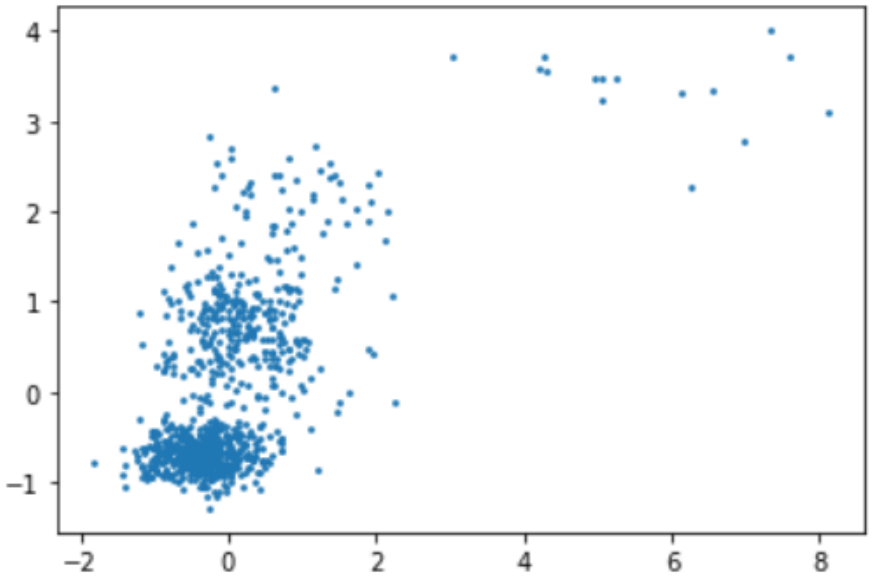
- Data is rescaled to have a minimum of 0 and a maximum of 1.
- Also referred to as **min-max scaling**.
- $x_{sc} = \frac{x - x_{min}}{x_{\max} - x_{min}}$

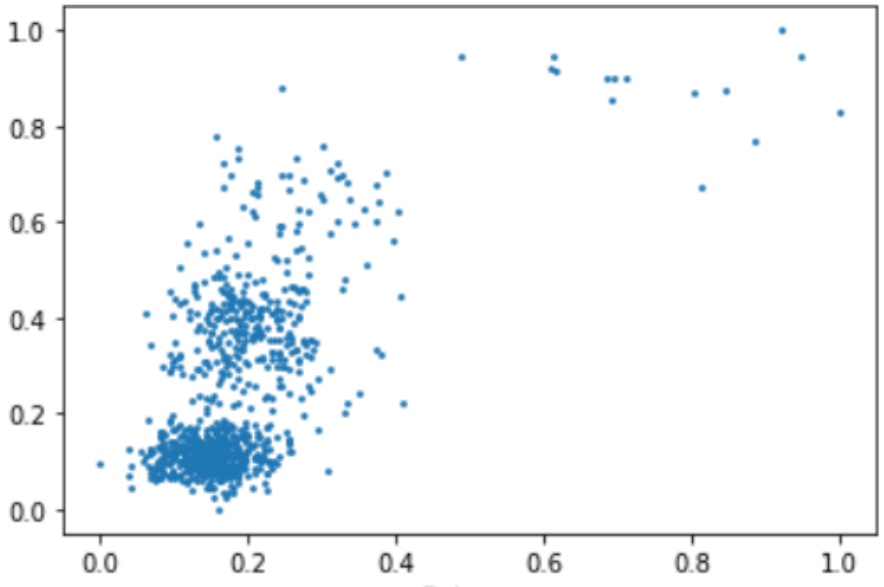
Standardisation

- Data is rescaled to have a mean of 0 and a standard deviation of 1.
- Also referred to as the **Z-score**.
- $z = \frac{x - \mu}{\sigma}$

Scaling

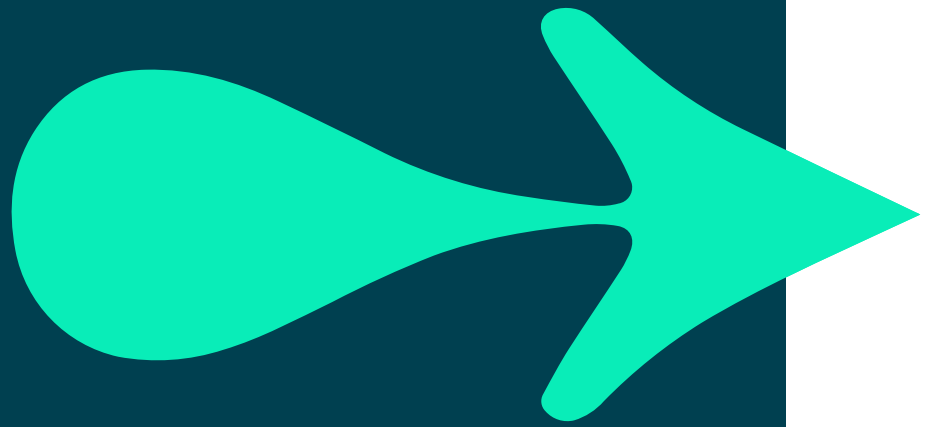

Income
80000
70000
60000
50000
40000
30000
20000
10000
0
1000
2000
3000
4000
5000
6000
Balance


4
3
2
1
0
−1
−2
0
2
4
6
8


1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0

Encoding

ORDINAL ENCODING VS. ONE-HOT



	colour	top_speed	price
0	red	fast	50000
1	black	medium	30000
2	silver	fast	45000
3	red	slow	35000
4	blue	slow	40000
5	black	medium	60000

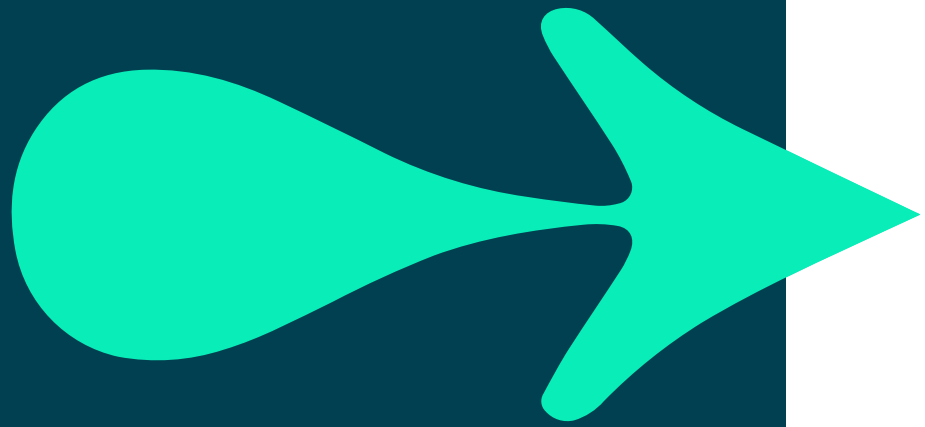
```
encoder = OrdinalEncoder()
cars['top_speed'] = encoder.fit_transform(cars[['top_speed']])
cars
```

	colour	top_speed	price
0	red	0.0	50000
1	black	1.0	30000
2	silver	0.0	45000
3	red	2.0	35000
4	blue	2.0	40000
5	black	1.0	60000

```
encoder = OneHotEncoder(sparse_output=False)
cars[encoder.get_feature_names_out()] = encoder.fit_transform(cars[['colour']])
cars.drop('colour', axis=1, inplace=True)
cars
```

	top_speed	price	colour_black	colour_blue	colour_red	colour_silver
0	fast	50000	0.0	0.0	1.0	0.0
1	medium	30000	1.0	0.0	0.0	0.0
2	fast	45000	0.0	0.0	0.0	1.0
3	slow	35000	0.0	0.0	1.0	0.0
4	slow	40000	0.0	1.0	0.0	0.0
5	medium	60000	1.0	0.0	0.0	0.0

FEATURE SELECTION AND ENGINEERING



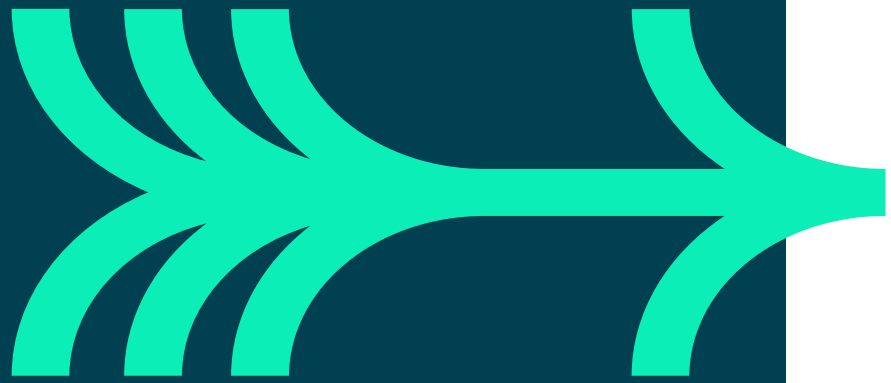
Feature engineering can dramatically improve model performance. It typically involves combining, transforming, and enriching features.

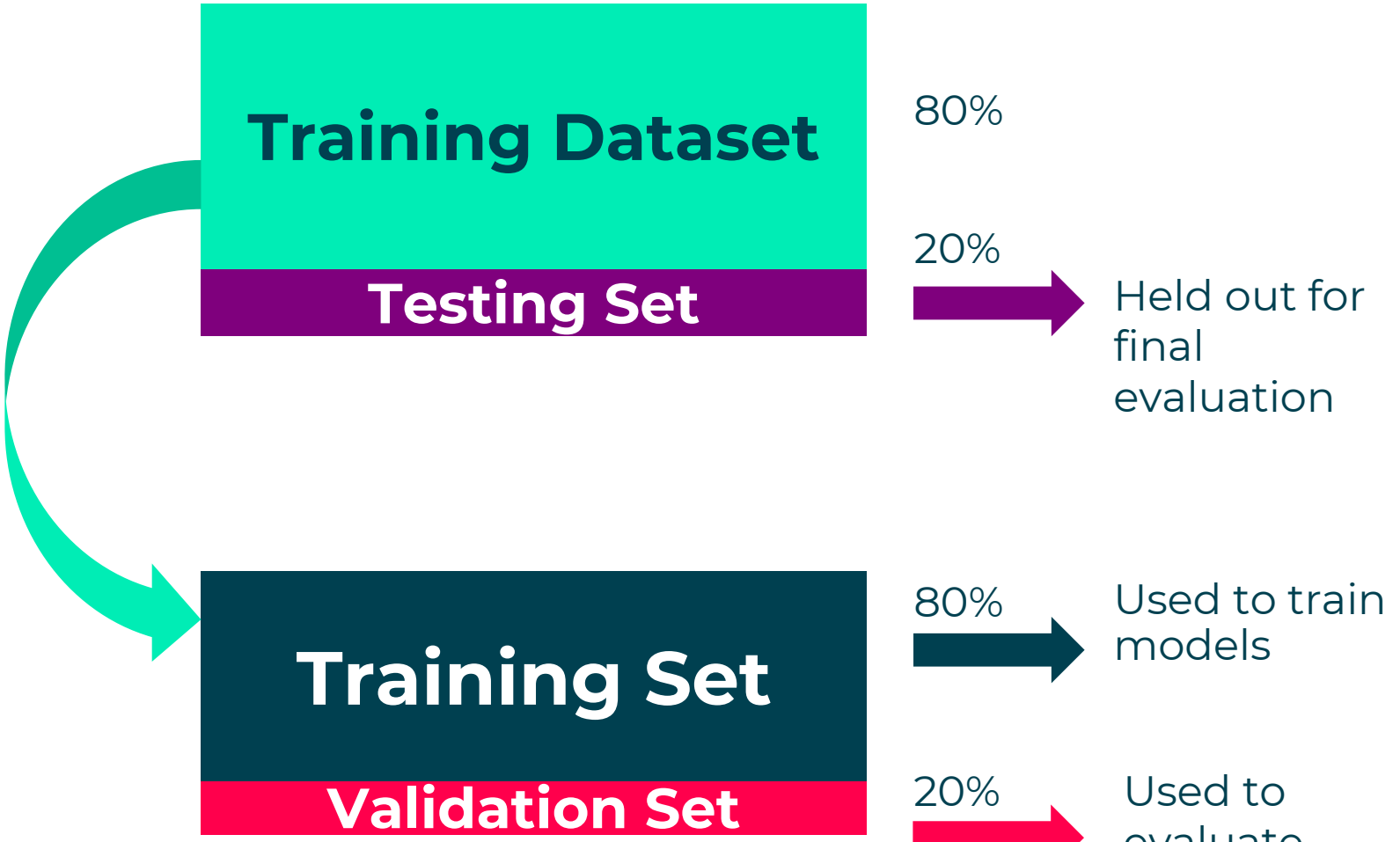
For example, advertisement clicks and views could be combined into a Clickthrough Rate (CTR).

Feature selection involves choosing the most explanatory or predictive features for your model

Training, testing, and cross validation

TRAINING, TESTING, AND VALIDATION





Cross validation

5-Fold Validation

Fold 1

Training Set
Training Set
Training Set
Training Set
Validation Set

Fold 2

Training Set
Training Set
Training Set
Validation Set
Training Set

Fold 3

Training Set
Training Set
Validation Set
Training Set
Training Set

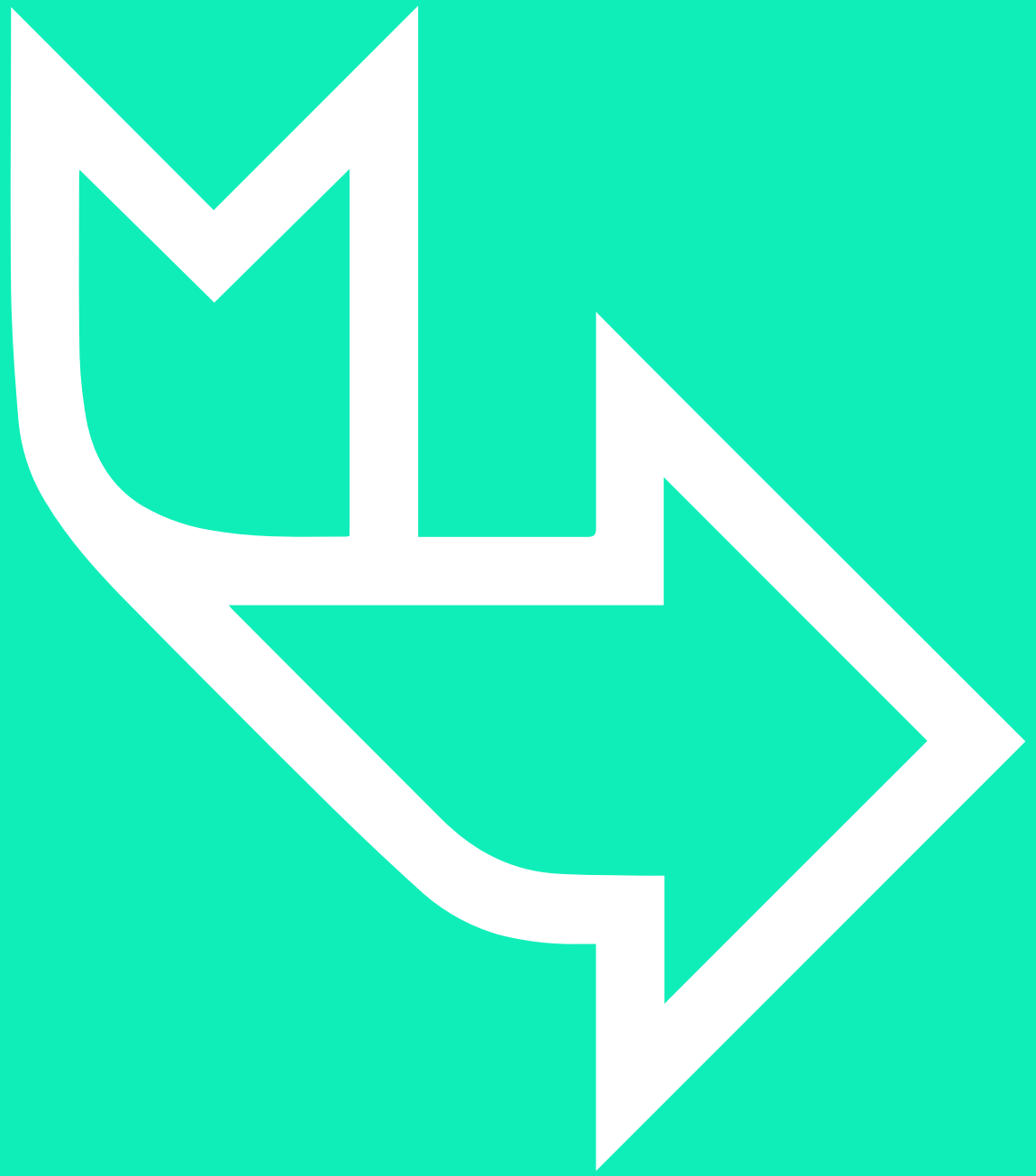
Fold 4

Training Set
Validation Set
Training Set
Training Set
Training Set

Fold 5

Validation Set
Training Set
Training Set
Training Set
Training Set

Each segment represents 20% of the dataset.



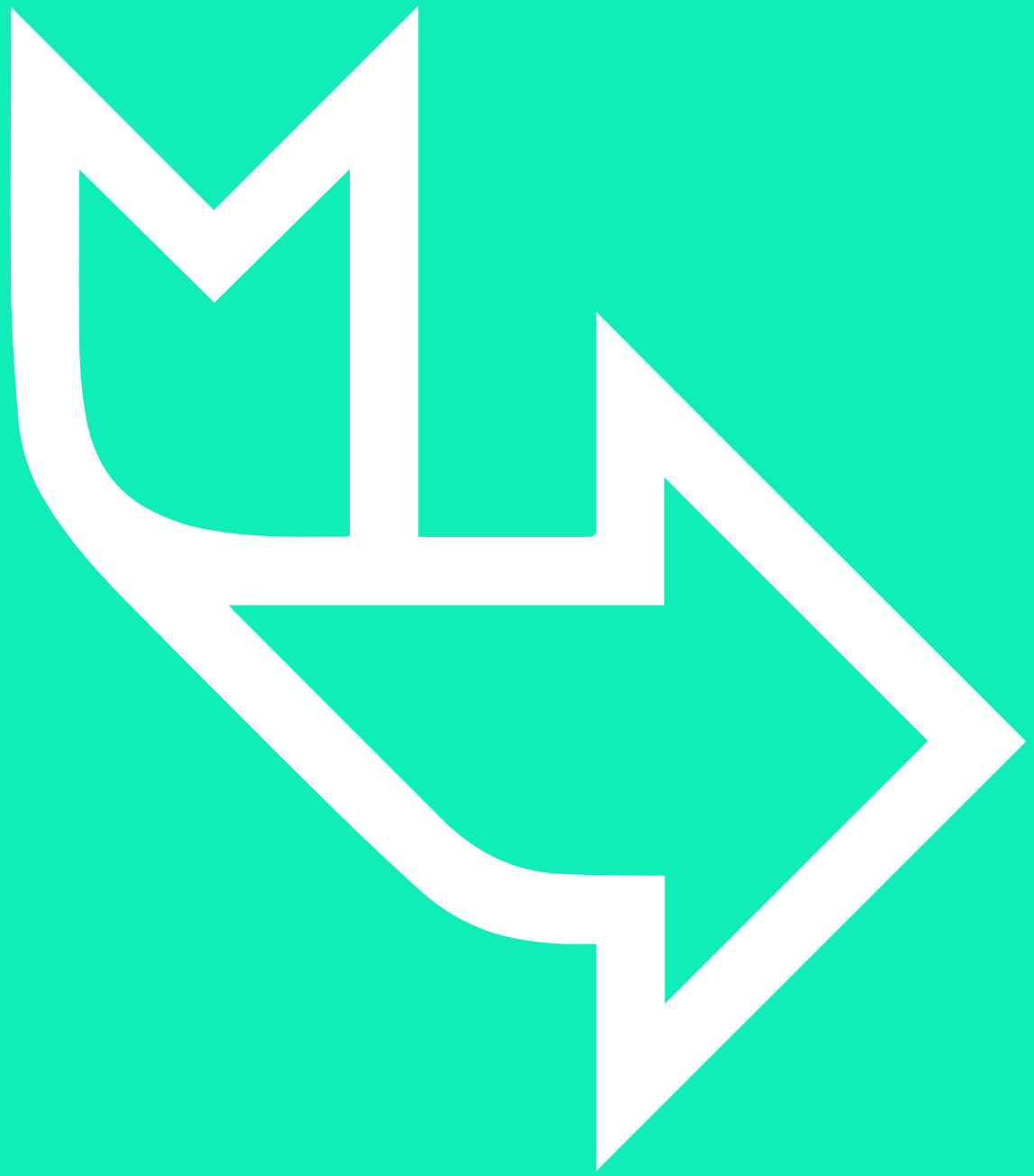
Exercise

Work though Module 4 Exercises: Preprocessing Data for Analysis

LEARNING CHECK

Think about your answers to these questions:

- What do we need to consider when working with duplicated data, missing values, or outliers?
- Why are scaling, encoding, and feature selection important?
- What is the importance of training, testing, and validation sets?
- Why do we often engineer novel features?

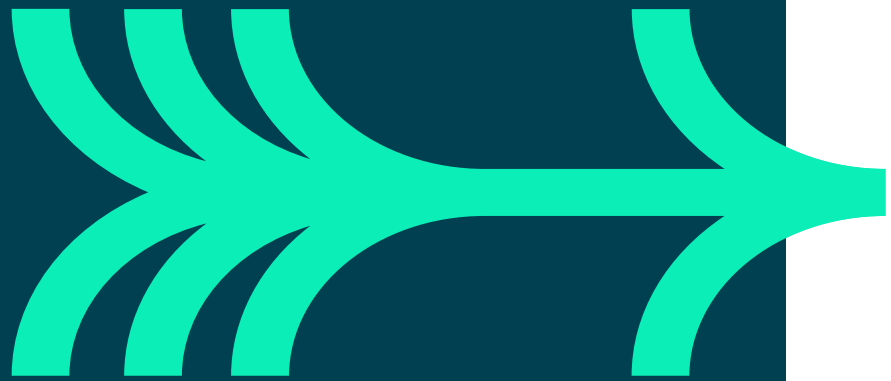


HOW DID YOU GET ON?

Learning Objectives

- Appropriately process duplicated data, missing values & outliers
- Understand the importance of scaling, encoding, and feature selection
- Describe the importance of training, testing & validation sets
- Engineer novel features to analyse

SUPERVISED LEARNING: REGRESSION



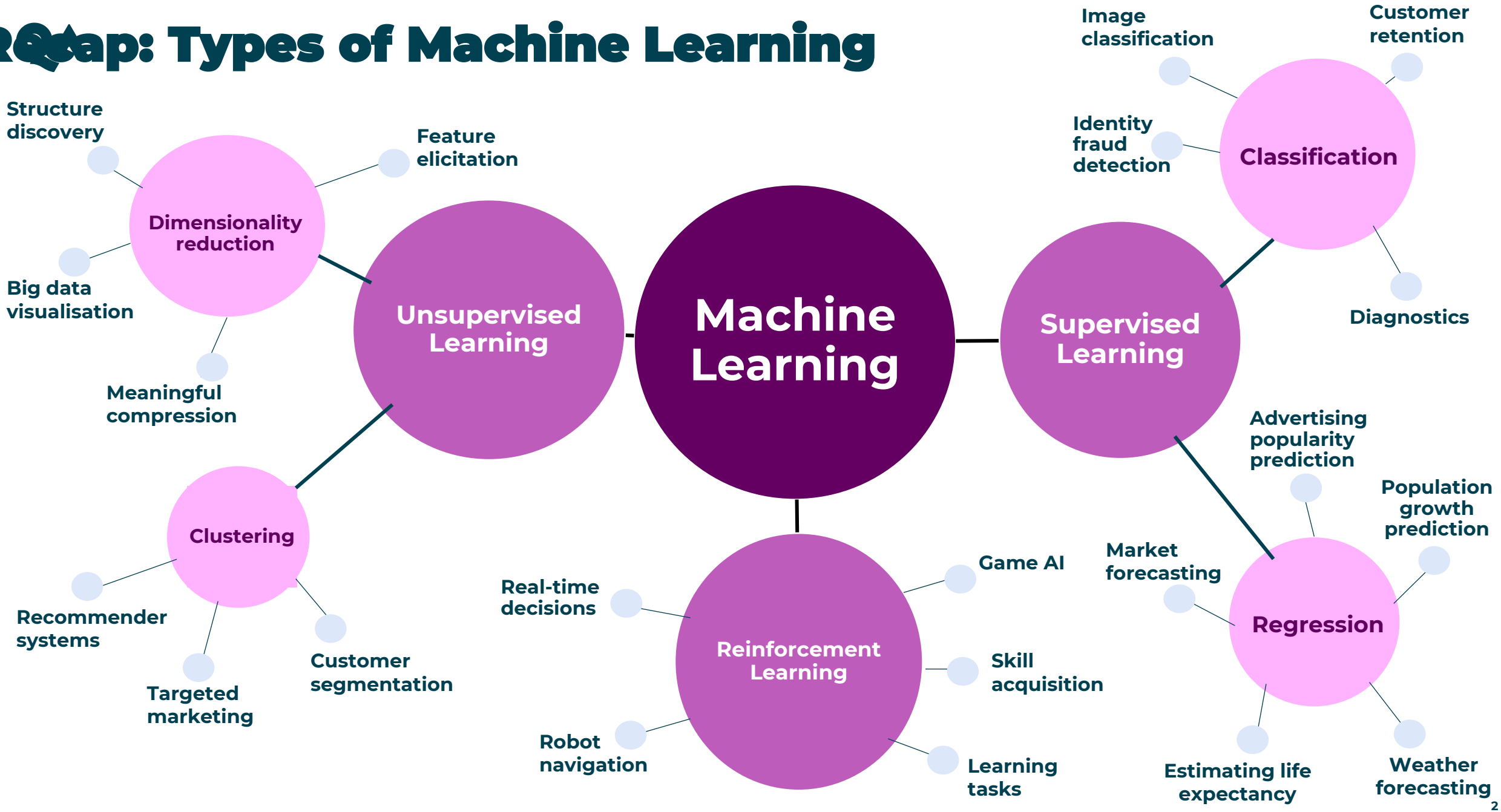
Learning objectives

- Describe regression in the context of machine learning.
- Build simple and multiple linear regression models.
- Understand non-linear regression approaches.
- Evaluate and compare regression models.

Expected prior knowledge

- Nothing is assumed about your background.

Recap: Types of Machine Learning



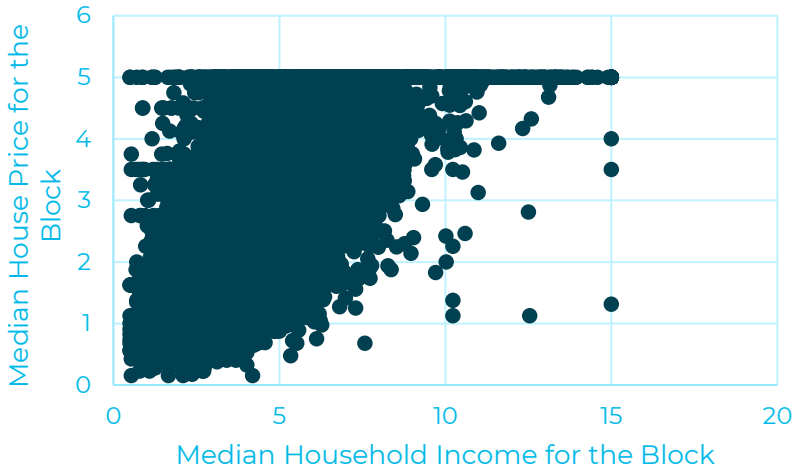
Regression

THE GOAL

Regression is used for **predicting a continuous variable**, y. We ask the question: what variable might it be useful to predict?

This variable can be placed on the vertical axis of a scatter plot, and we can try different variables on the x axis to work towards predicting y accurately.

Example: Predicting the average house price for a ‘block’ in California:



POSSIBLE MODELS

- Linear Regression
- Polynomial Regression
- Splines
- Decision Tree
- Random Forest
- XGBoost

And more!

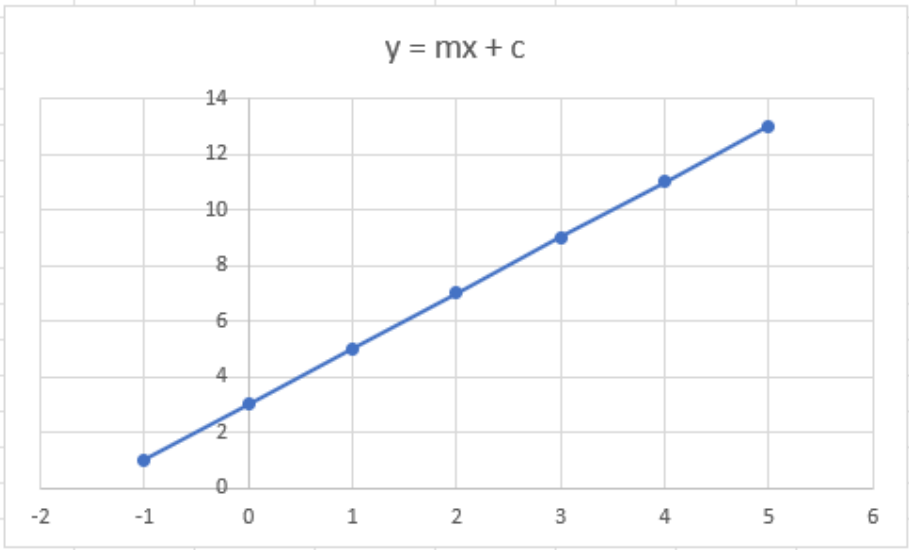
Linear Regression

All straight lines have an equation that can be written in the form **y = mx + c**

Equation of a line and terminology

x	y = mx + c
-1	1
0	3
1	5
2	7
3	9
4	11
5	13

m:	2
c:	3



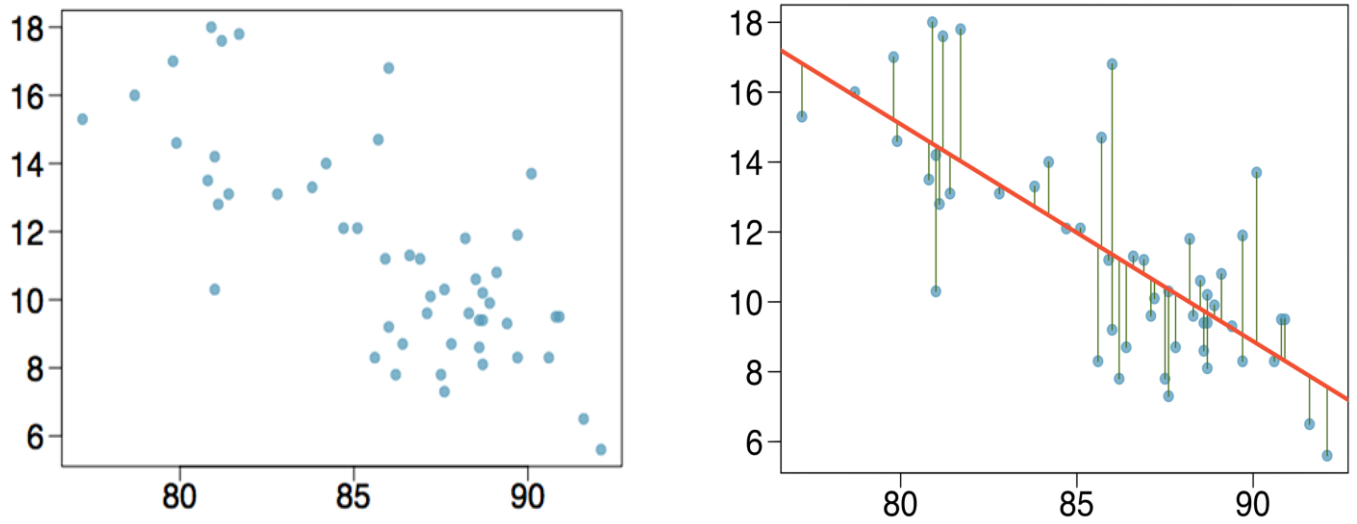
m is referred to as the **coefficient of x**. On a plot, we see it's effect as the **gradient** of the line. For Data Analytics, the symbol $\boldsymbol{\beta_1}$ might be used instead. We can interpret it as the impact that changing x has on the predicted y value – if x increases by 1 unit, m is the amount that the y prediction changes by.

c is a **constant**. On a plot it controls the height of the graph – when x=0 we see that it is where the line **intercepts the y-axis**. For Data Analytics, the symbol $\boldsymbol{\beta_0}$ might be used instead.

Residuals (or Loss)

RESIDUALS

Real data is rarely located in perfect straight lines!



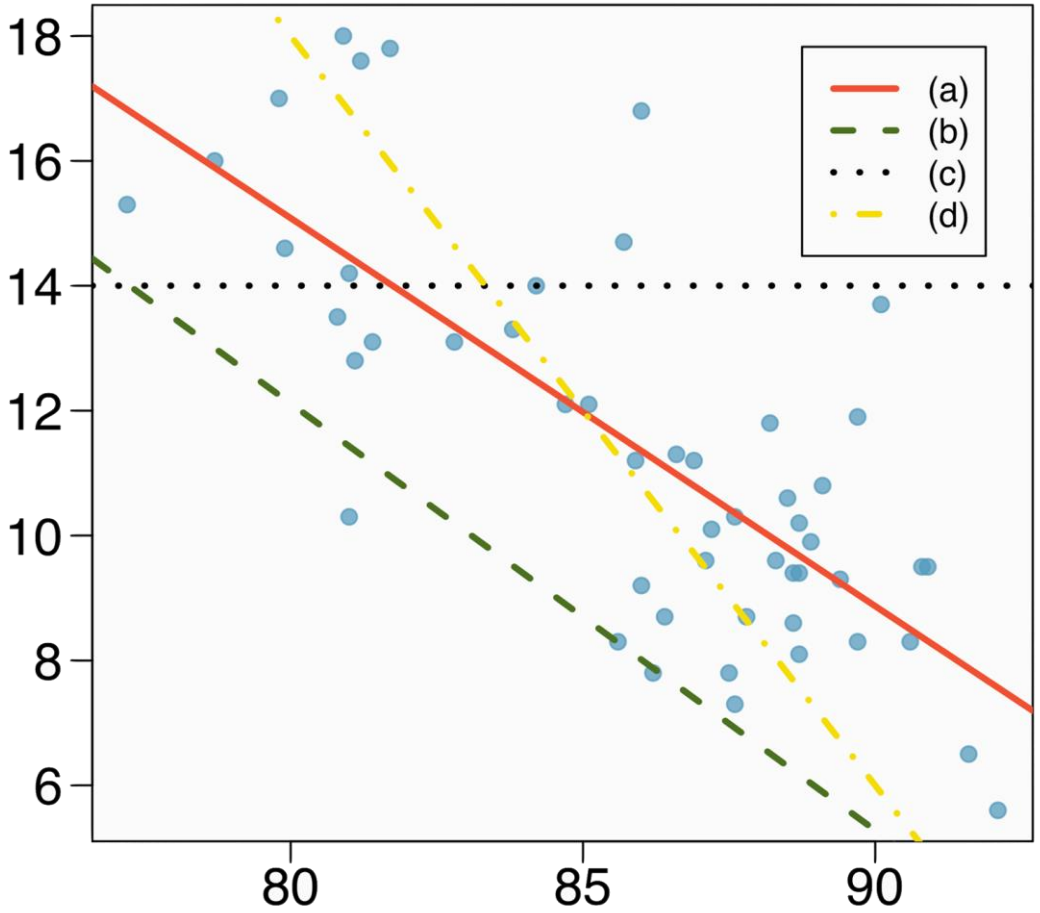
If we fit a line through the data, the **vertical differences** between the data and the line are called **residuals**.

Each one of these residuals shows the difference between the real y-coordinate and what the equation of the line predicts from the x coordinate.

Residuals represent the **error** that a prediction would have. **Linear Regression** is a method that involves finding the best $\boldsymbol{\beta_1}$ and $\boldsymbol{\beta_0}$ (i.e. the best equation y = mx + c) to minimise the error.

EYEBALLING THE LINE OF BEST FIT

Which of the following appears to be the line that best fits the data?



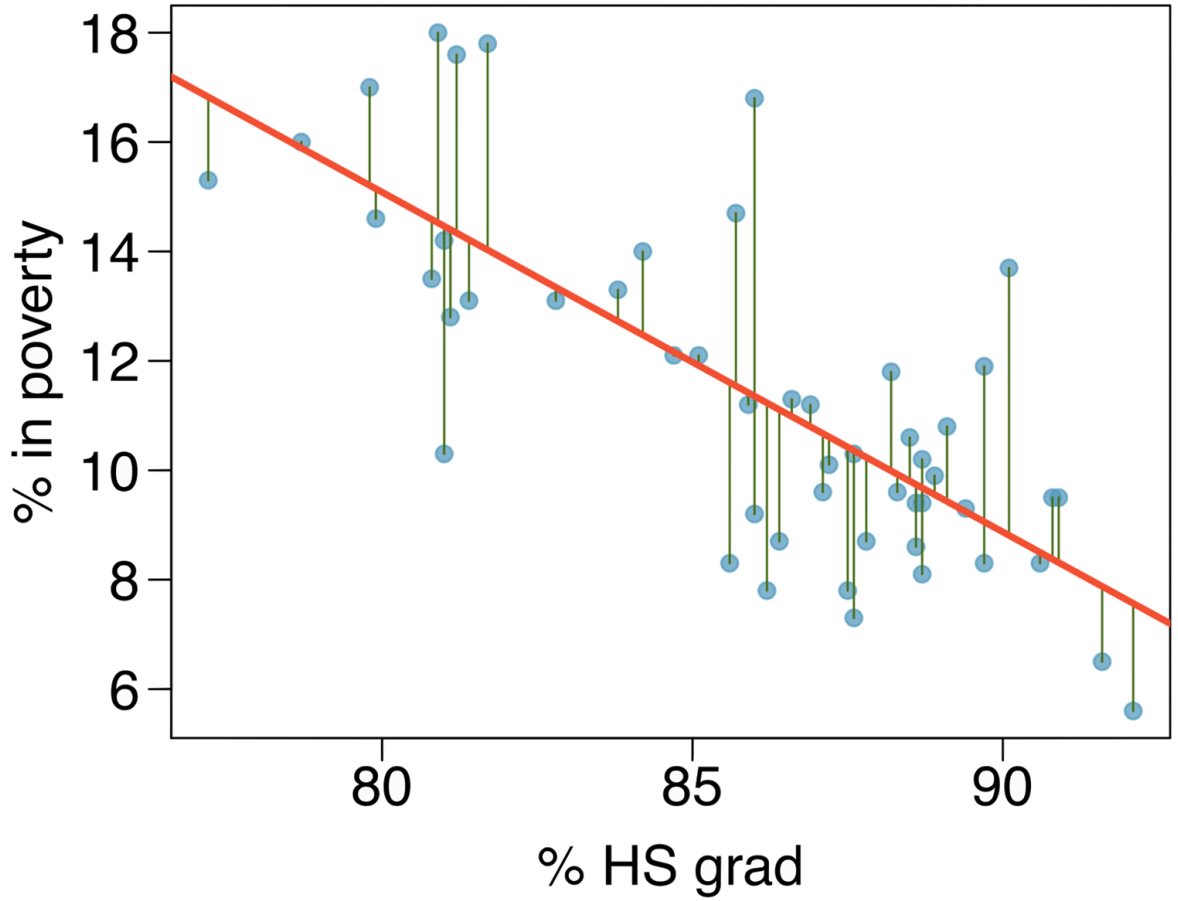
RESIDUALS

Leftovers from the model fit:

Data = Fit + Residual

Predicted: $\hat{y}_i = mx_i + c$

Observed: $y_i = mx_i + c + e_i$

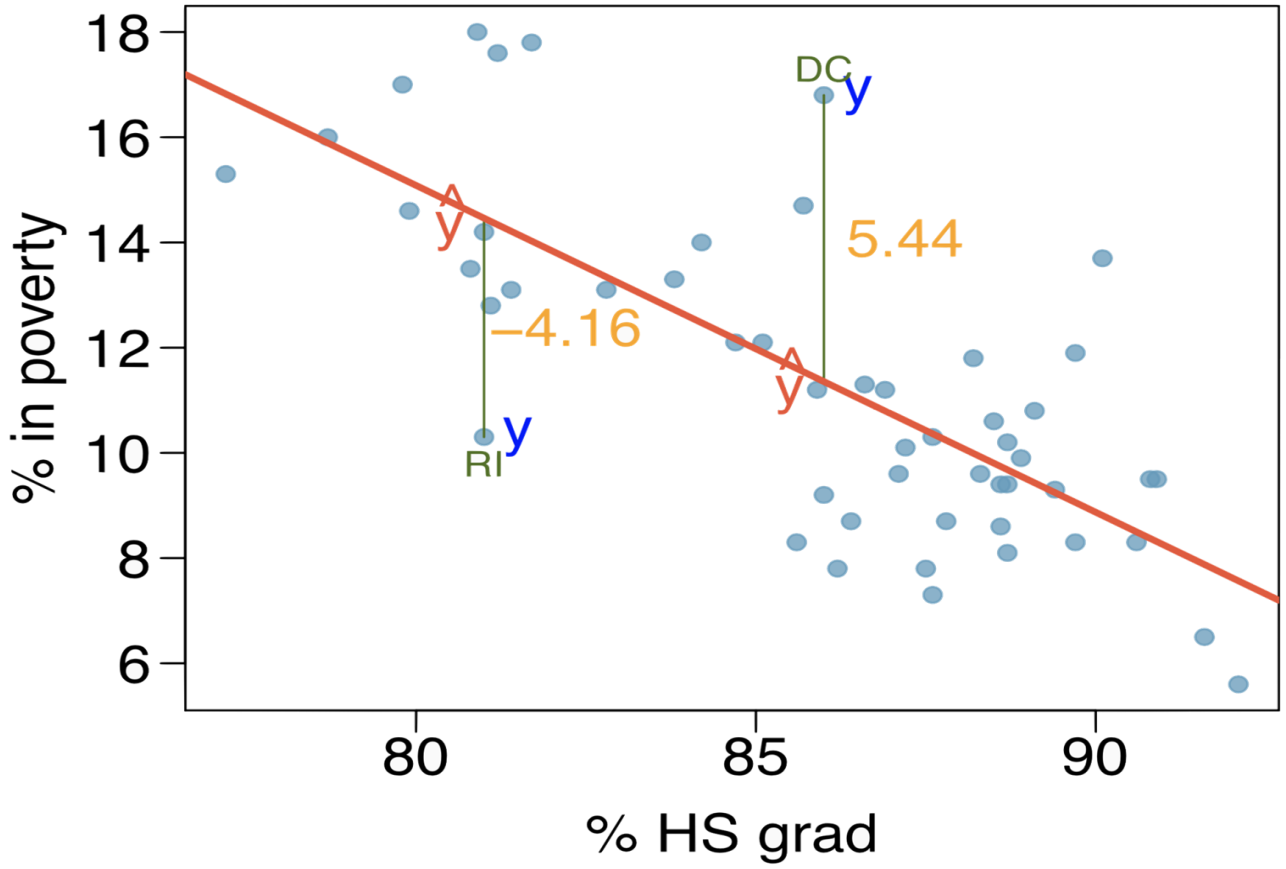


RESIDUALS

Difference between the observed (y_i) and predicted $\hat{y}_i$

$$e_i = y_i - \hat{y}_i$$

% living in poverty in RI is 4.16% less than predicted



Fitting a Regression Model

FITTING

We want a line that has small residuals...

Option 1: Minimise the sum of magnitudes (absolute values) of residuals

$$|e_1| + |e_2| + \ldots + |e_n|$$

Option 2: Minimise the sum of squared residuals – 'least squares'

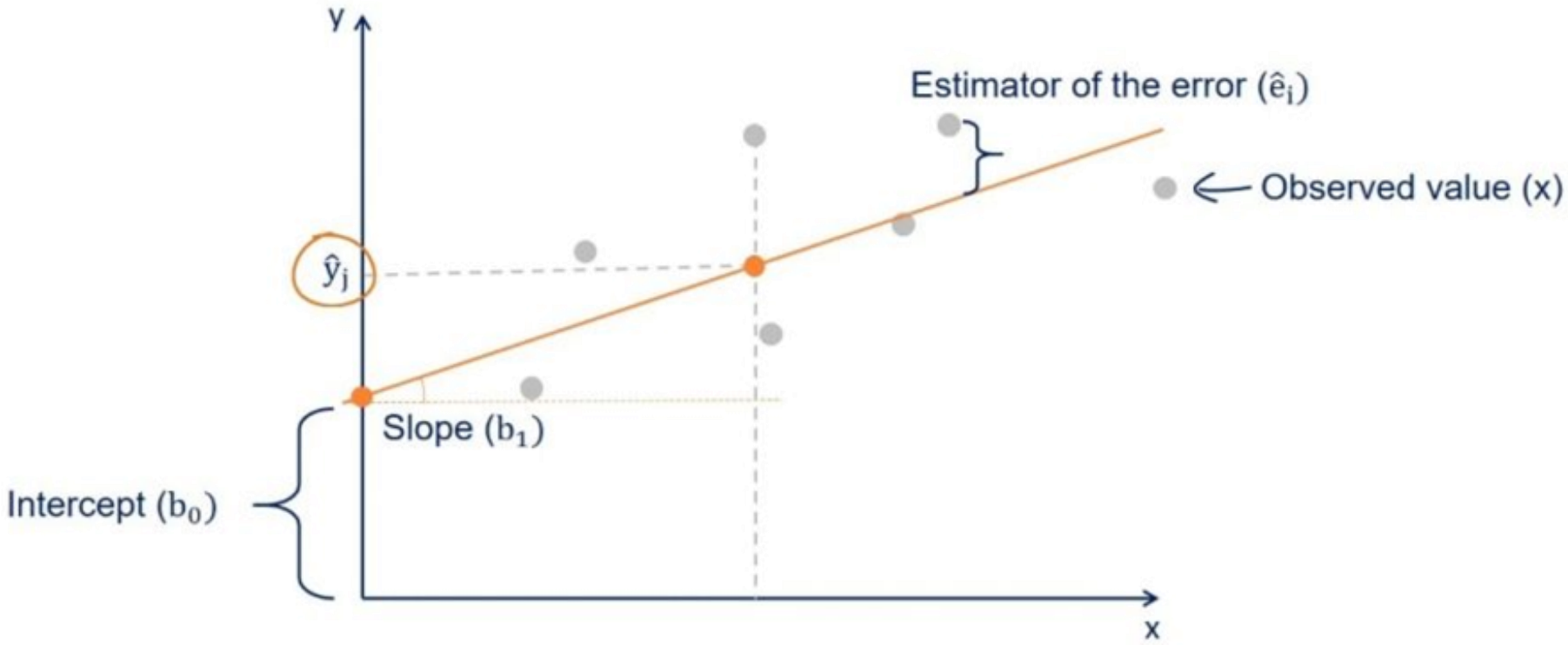
$$e_1^2 + e_2^2 + \ldots + e_n^2$$

Why least squares?

1. Most commonly used.
2. Easier to compute by hand and using software.
3. In many applications, a residual twice as large as another is usually more than twice as bad.

Regression coefficients

$$Y_i = \beta_0 + \beta_1 X_i + \hat{e}_i$$



LEAST SQUARES LINE: CONDITIONS

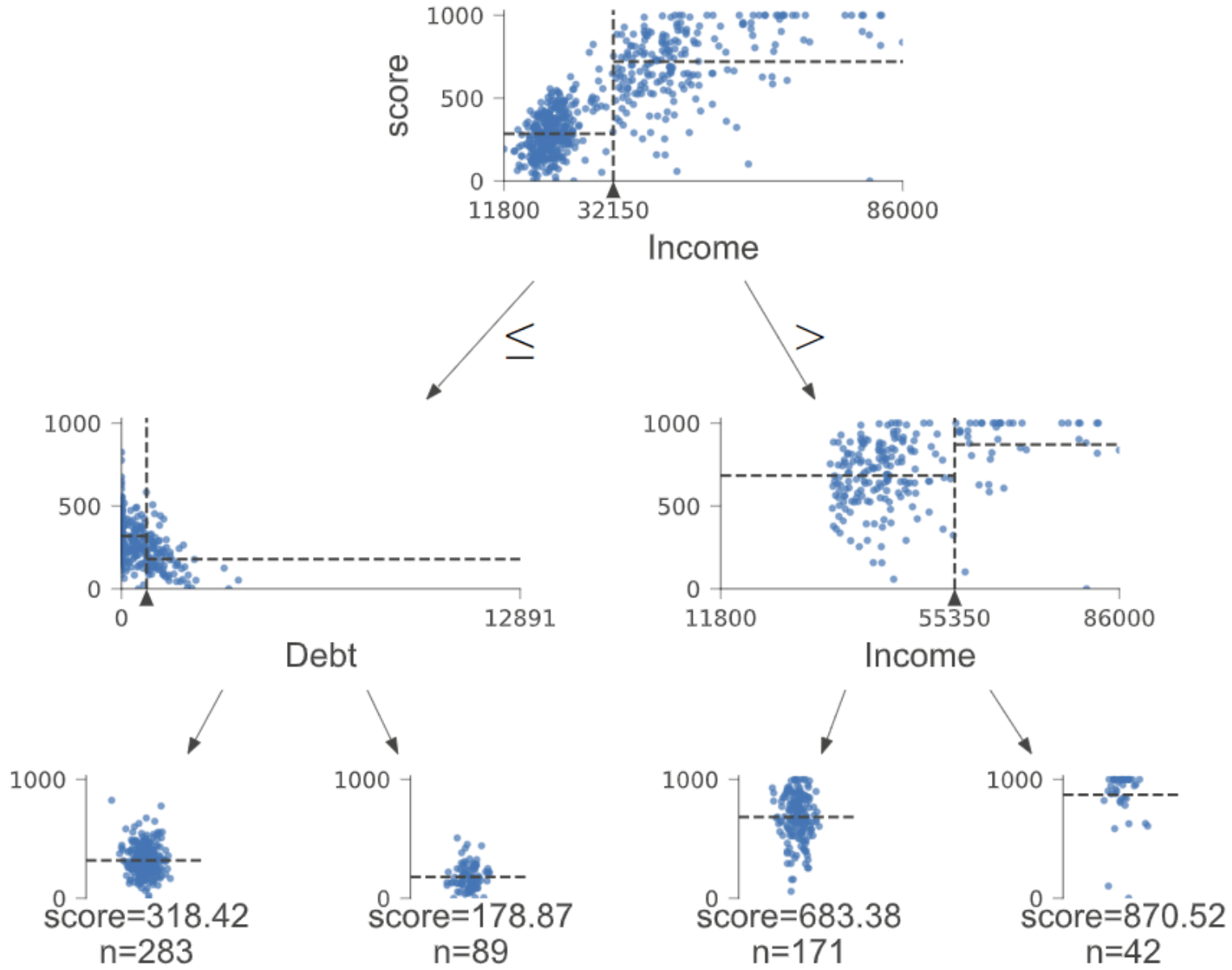
1. Linearity
2. Nearly normal residuals
3. Constant variability

Alternatives to Linear Regression

DECISION TREE FOR REGRESSION

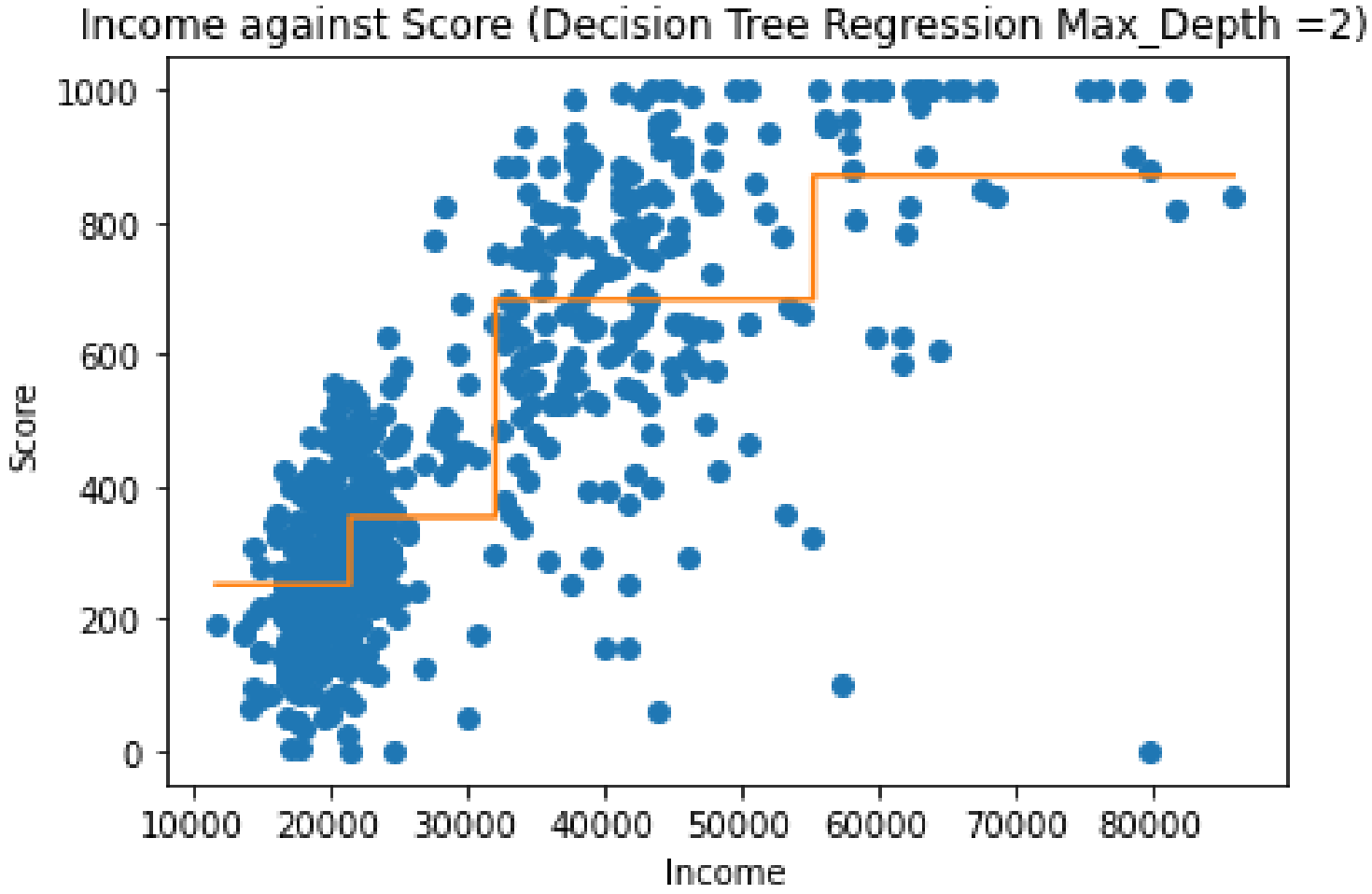
An illustration of a regression tree model to predict credit scores from the loans data using income, balance and debt as predictors.

MaxDepth = 2



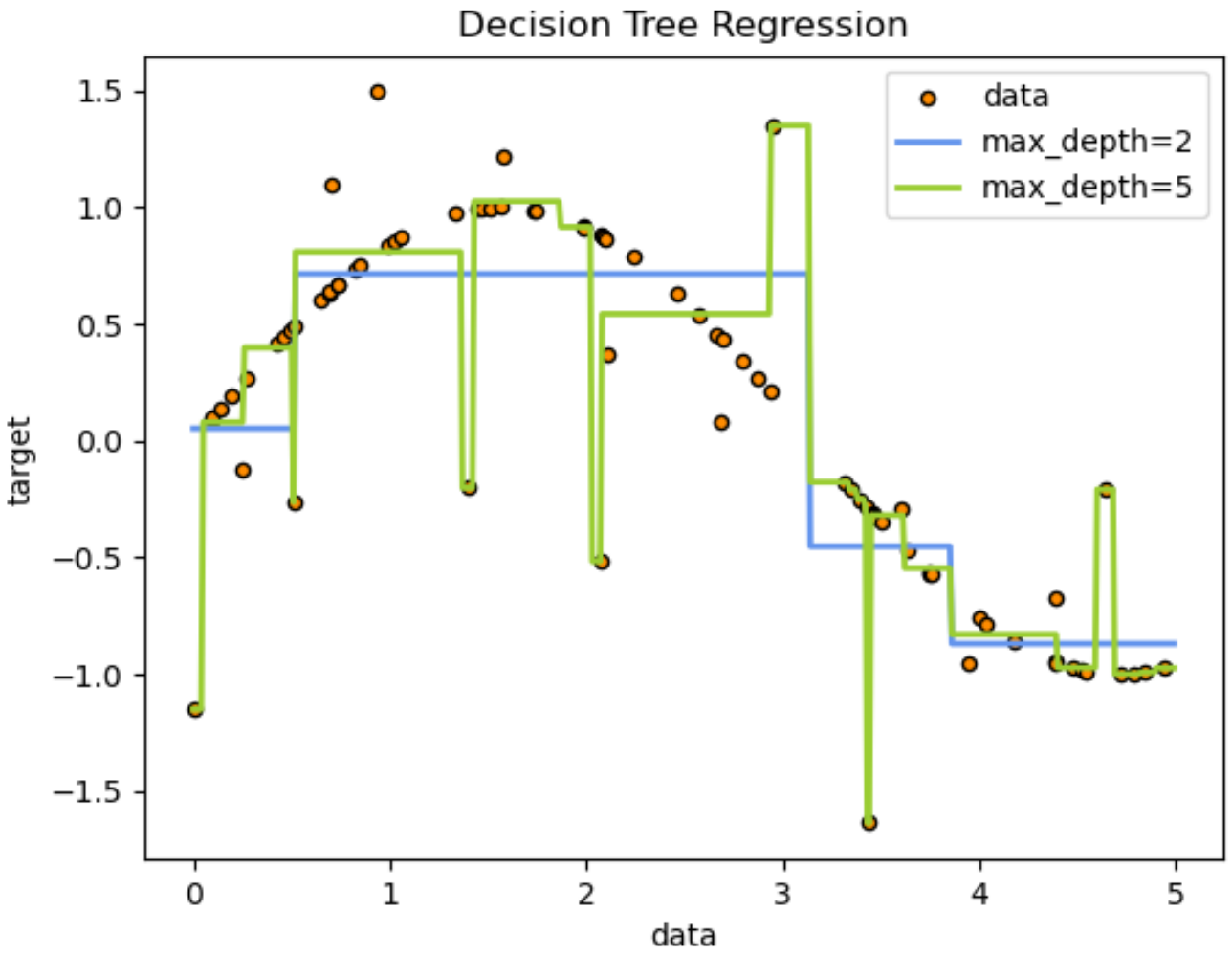
DECISION TREE FOR REGRESSION

Decision tree based regression. Note the stepwise nature of the resulting model.



MAX-DEPTH HYPERPARAMETER

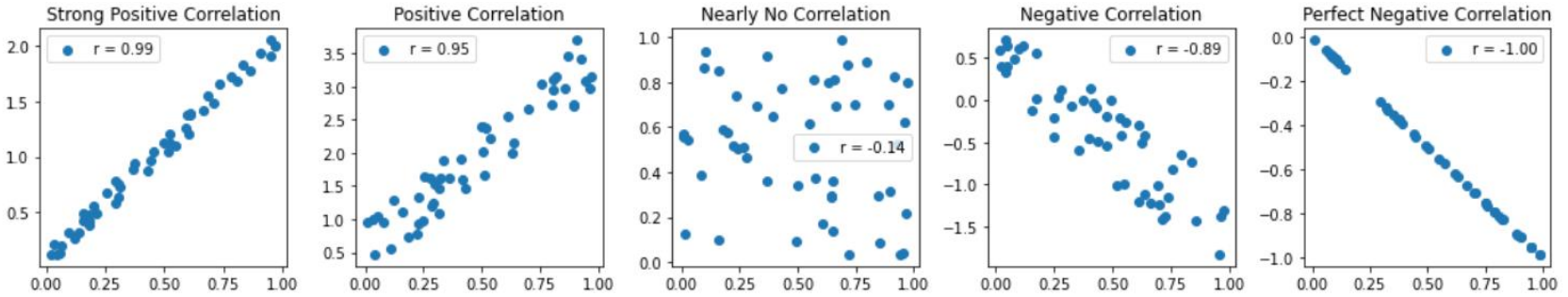
With the depth too low or high, the model will under or overfit. Below is an example of a regression tree trying to model a periodic function with some noise. With MaxDepth 5, it is too susceptible to the noise.



Using Regression Models

GOOD LINEAR MODELS

Even if the scatter plot appears elliptical, a regression line may not give accurate predictions.



If correlation is strong, the ellipse will be more 'flat' and residuals will be small – this indicates that predicted values from new data will probably have lower errors.

The data used to create the regression line is a **sample** from a population which includes new data that we would like to predict values for.

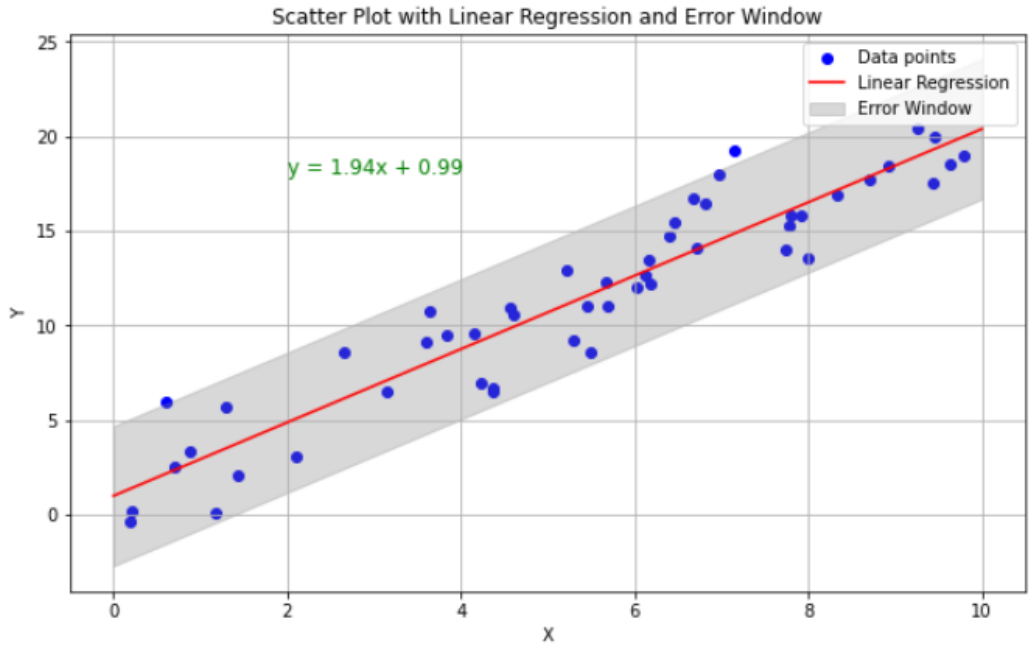
We can **test for correlation** based on the existing data to give us some evidence towards whether to use the linear model or not.

PREDICTING FROM A LINEAR MODEL

The equation for the regression line is calculated using existing data.

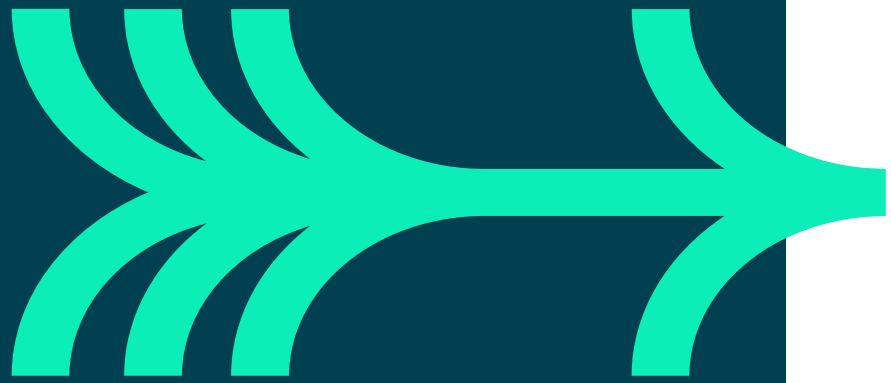
Suppose a new value of x is requested and we want to predict the value of y that goes with it.

If x=6, the equation gives us y=12.63, but we can see from the error window that it could be as big as about 17 or as small as about 8.5.



Evaluating Regression

REGRESSION METRICS: ERRORS



Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=0}^{N}(y_i - \hat{y}_i)^2}$$

Mean Absolute Error (MAE)

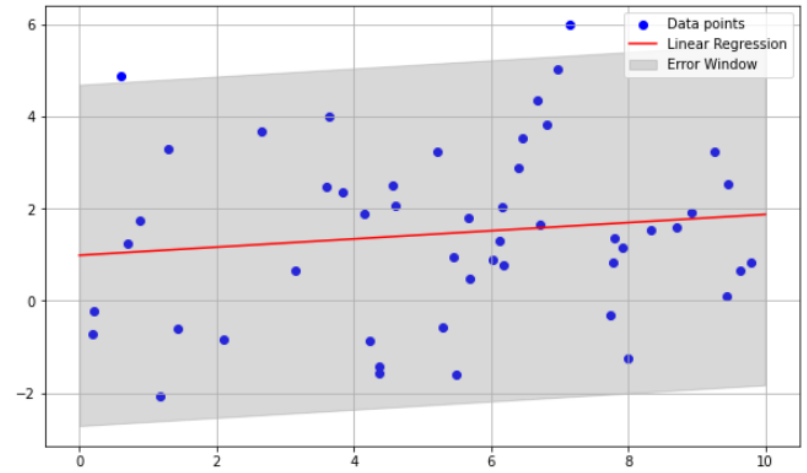
$$MAE = \frac{1}{N}\sum_{i=0}^{N}|y_i - \hat{y}_i|$$

There are others!

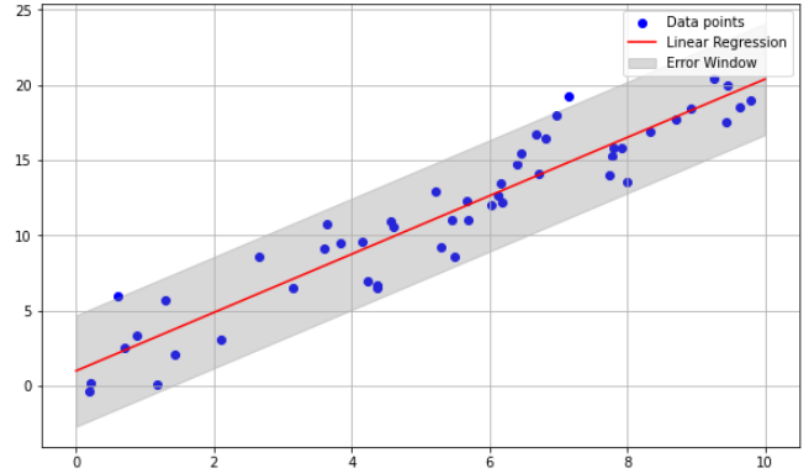
REGRESSION METRICS: USEFULNESS

R^2 is the **coefficient of determination**. It can be used as a measure of **usefulness** of the model. It's a measure of the variance of the dependent variable, y.

It's on a scale of 0 (bad) to 1 (good).

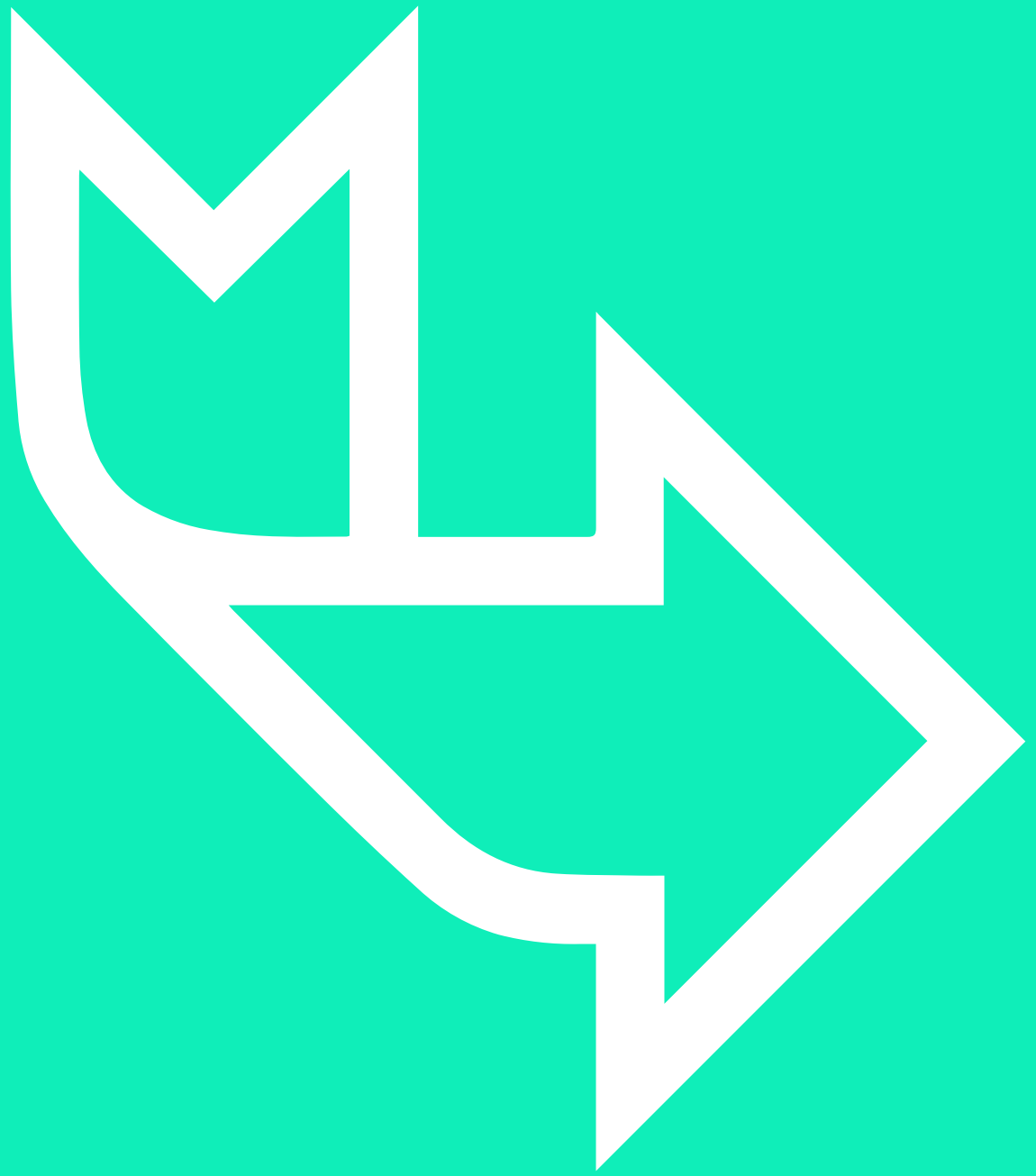


R Squared = 0.02



R Squared = 0.89

Consider the predicted y values in these cases – if you predict from different x inputs, does it make much difference to what is predicted?



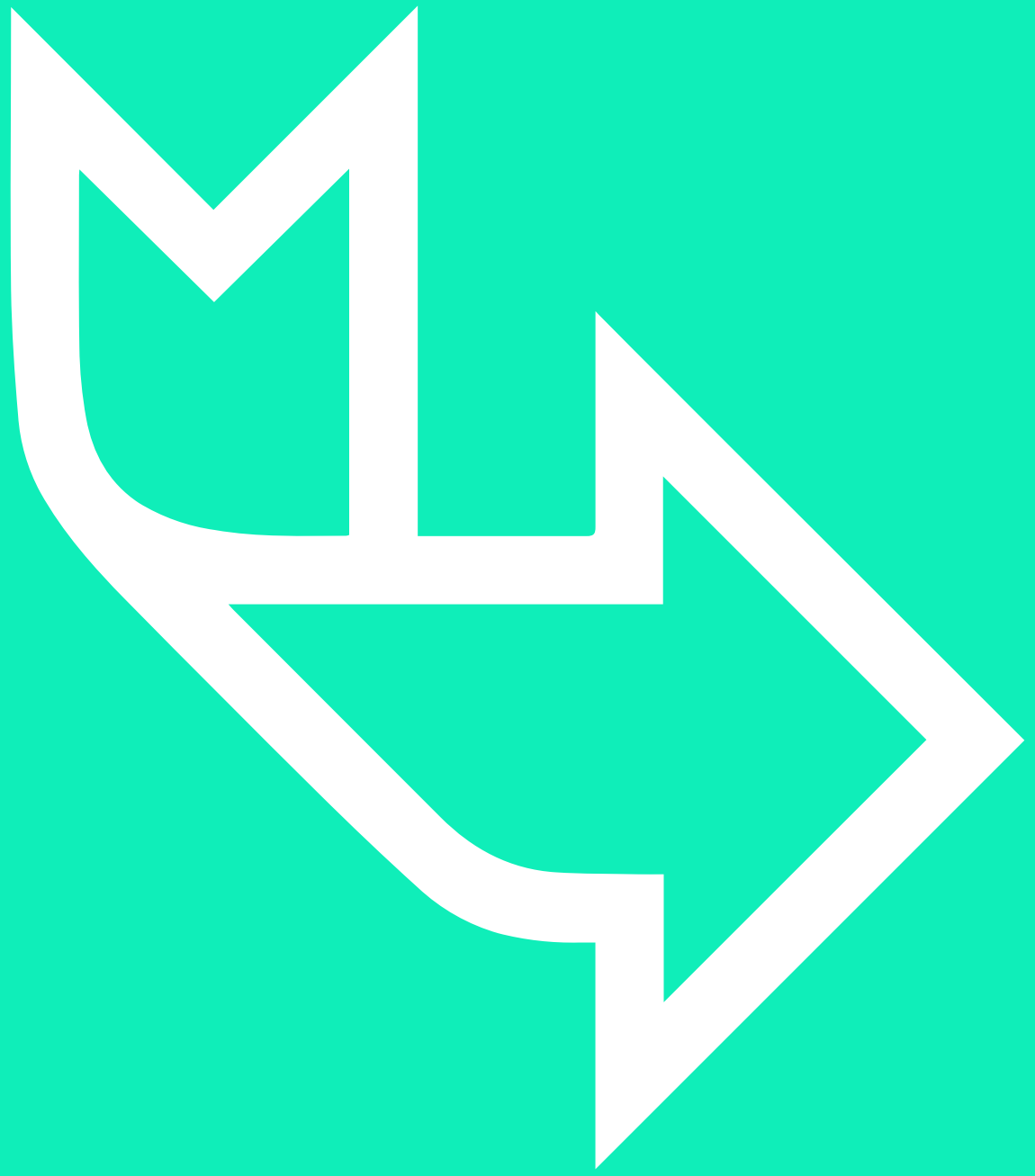
Exercise

Work though Module 5 Exercises: Regression

LEARNING CHECK

Think about your answers to these questions:

- What is regression?
- What is the difference between simple and multiple linear regression models?
- What are some non-linear regression approaches?
- How can we evaluate regression models?

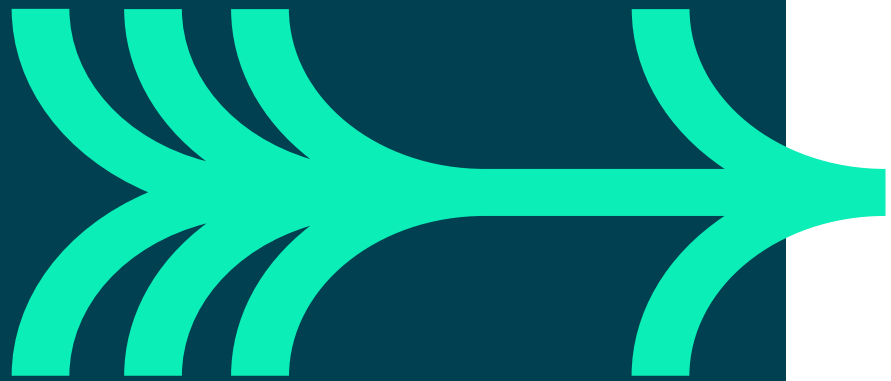


HOW DID YOU GET ON?

Learning objectives

- Describe regression in the context of machine learning.
- Build simple and multiple linear regression models.
- Understand non-linear regression approaches.
- Evaluate and compare regression models.

SUPERVISED LEARNING: CLASSIFICATION



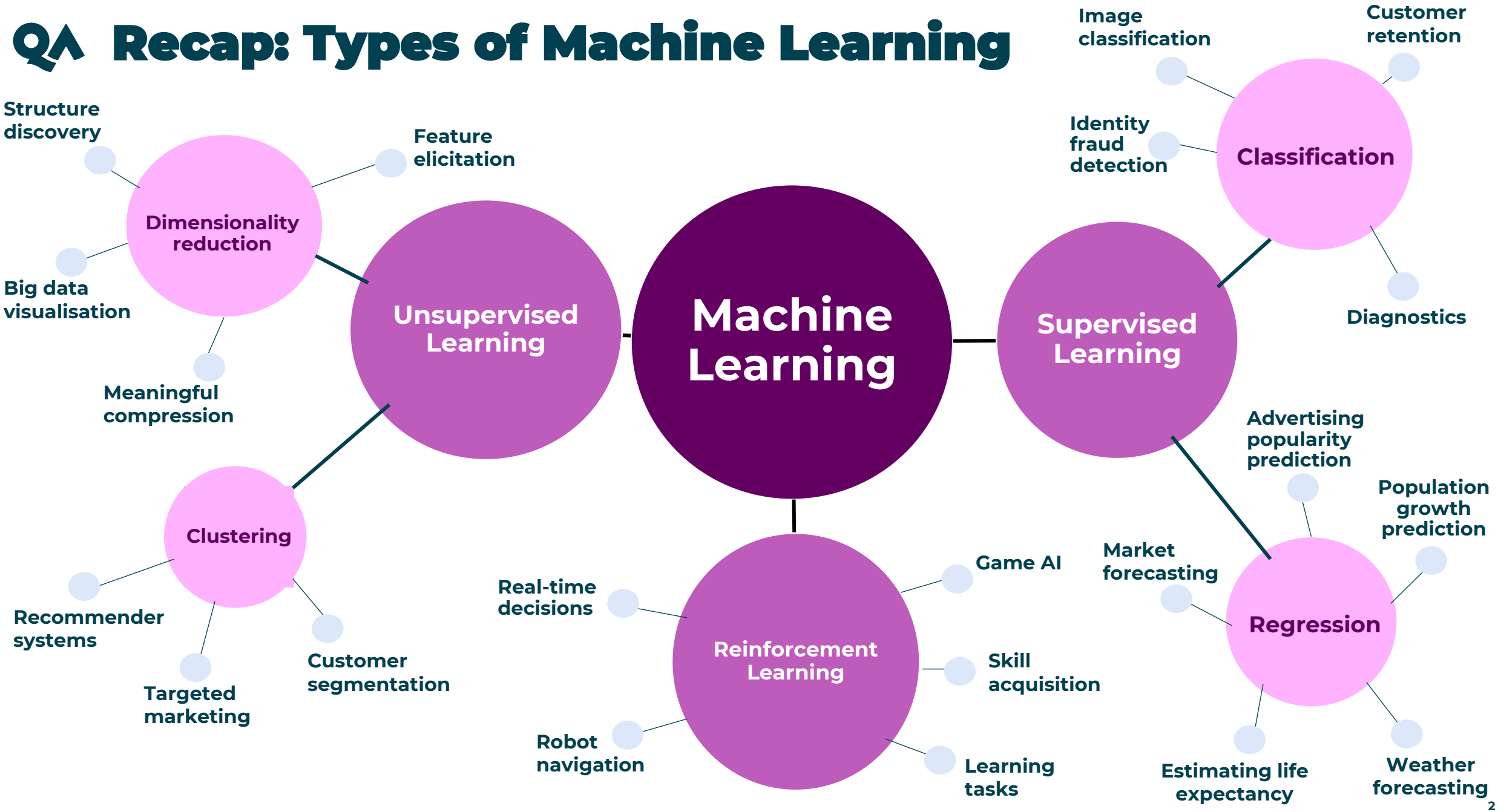
Learning objectives

- Describe classification in the context of machine learning.
- Build simple and multiple logistic regression models for classification.
- Build Decision Tree and Random forest models for classification.
- Evaluate and compare classification models.

Expected prior knowledge

- Nothing is assumed about your background.

Recap: Types of Machine Learning



Classification

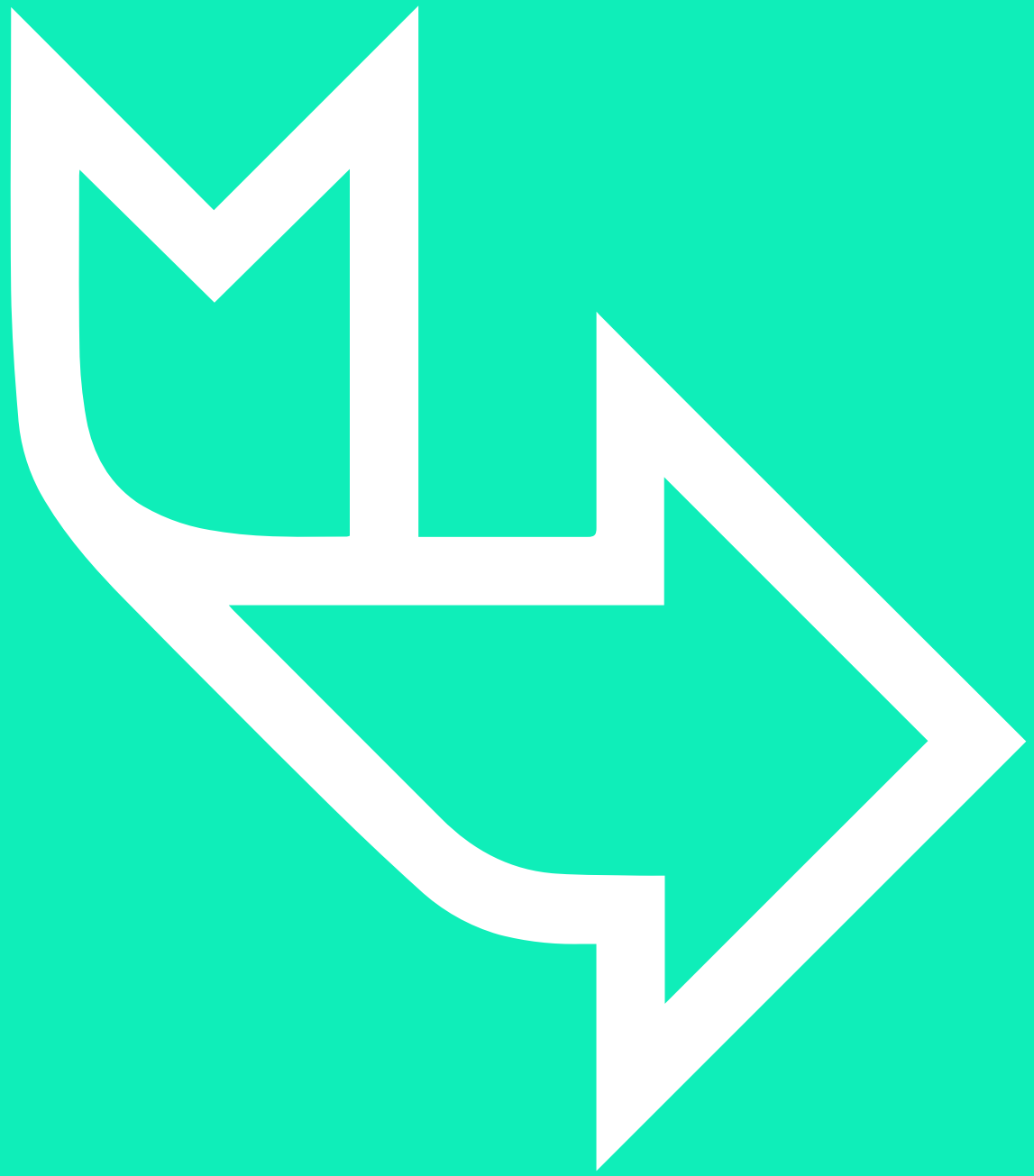
Exercise scenario

You work for a bank as a data scientist. One of the products offered is personal loans.

The bank has data on previous loans sold. While most people repaid the loan, some defaulted.

The bank obviously wants to minimise the amount of loans they agree to that are defaulted.

You have been tasked with creating a model to help in this process.



EXERCISE

Consider the following, then write down a few ideas to each of the points below, after which we will discuss as a group:

- **How could you frame this as a Machine Learning problem?**
- **What data do you think you would require for this?**
- **What would be the success criteria for your solution?**
- **How would you assess if you have met this criteria?**

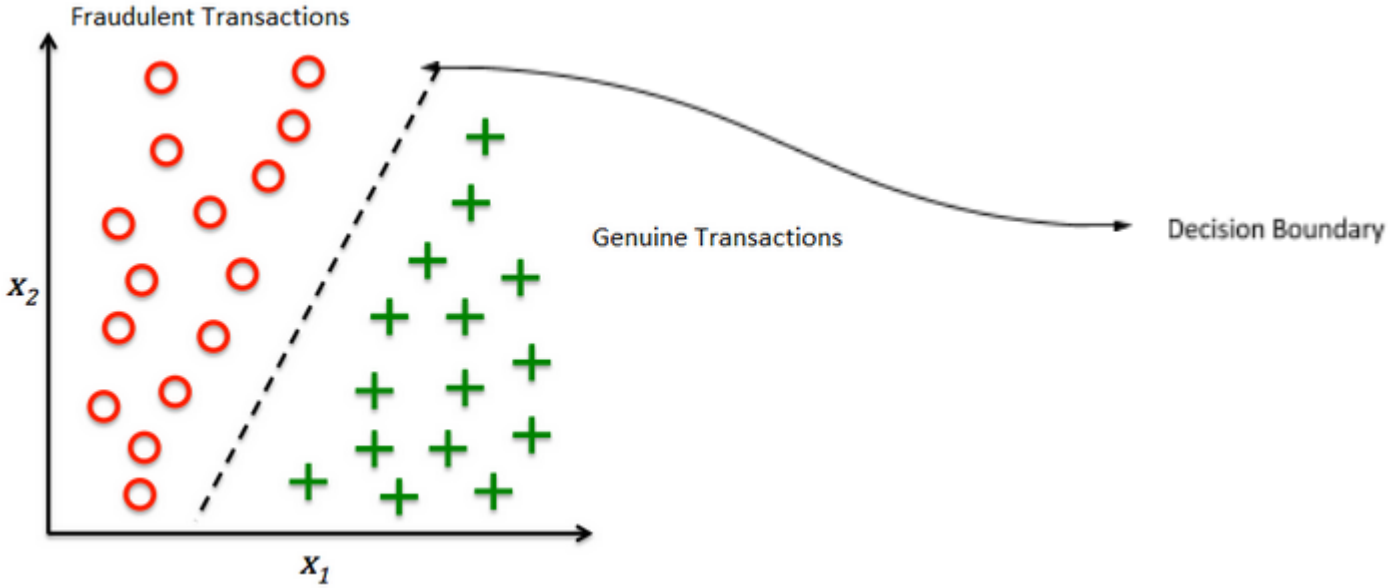
Logistic Regression

WHAT IS LOGISTIC REGRESSION?

- Logistic regression is an extremely popular **machine learning** technique.
- Despite the name, logistic regression is a **classification** algorithm.
- It is also an example of **supervised learning.**
- The output of a logistic regression model is the **probability** of an occurrence belonging to a **particular class**, e.g., classifying emails as being spam or not.
- In our case study, we will be using it as a **binary** classifier, although it can also be applied to **multinomial** and **ordinal** classes

ASSUMPTIONS

One of the key assumptions for standard logistic regression is that the **binary groups** are l**inearly separable**, or **mostly linearly separable**, i.e., not perfectly separable, as shown below:



Above is an example of perfectly linearly separable and balanced data, meaning we have an **equal distribution of classes.**

Source

ASSUMPTIONS

Multi-collinearity:
The solutions we have for this in logistic regression are the same as the ones we use for standard regression cases.

In the demonstrations following, we will investigate this with the support variables.

Independence:
As with linear regression, all samples must be independent of each other.

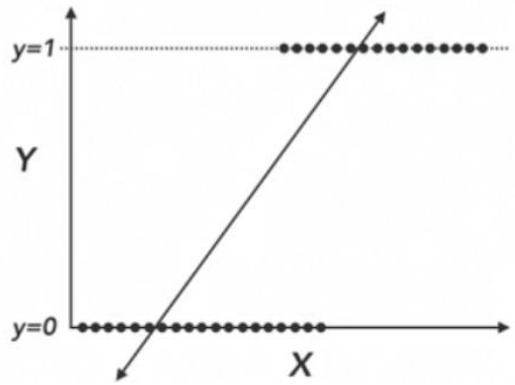
THE MODEL

Here is a simple example with one **support variable** (hours studying) and a **binary target** (pass or fail an exam).

As opposed to **linear regression**, which builds a **straight line**, **logistic regression** builds a **sigmoid function** that gives us the **probability** of a **data point** belonging to a particular class.

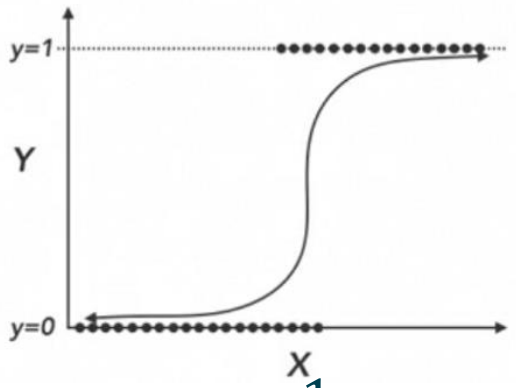
$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

Linear Regression



$$Y = \beta_0 + \beta_1 x$$

Logistic Regression



$$Y = \frac{1}{1+e^{-(\beta_0 + \beta_1 x)}}$$

Source

THE MATHS

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

As the output of the logistic regression is a probability, it will always be between 0 and 1.

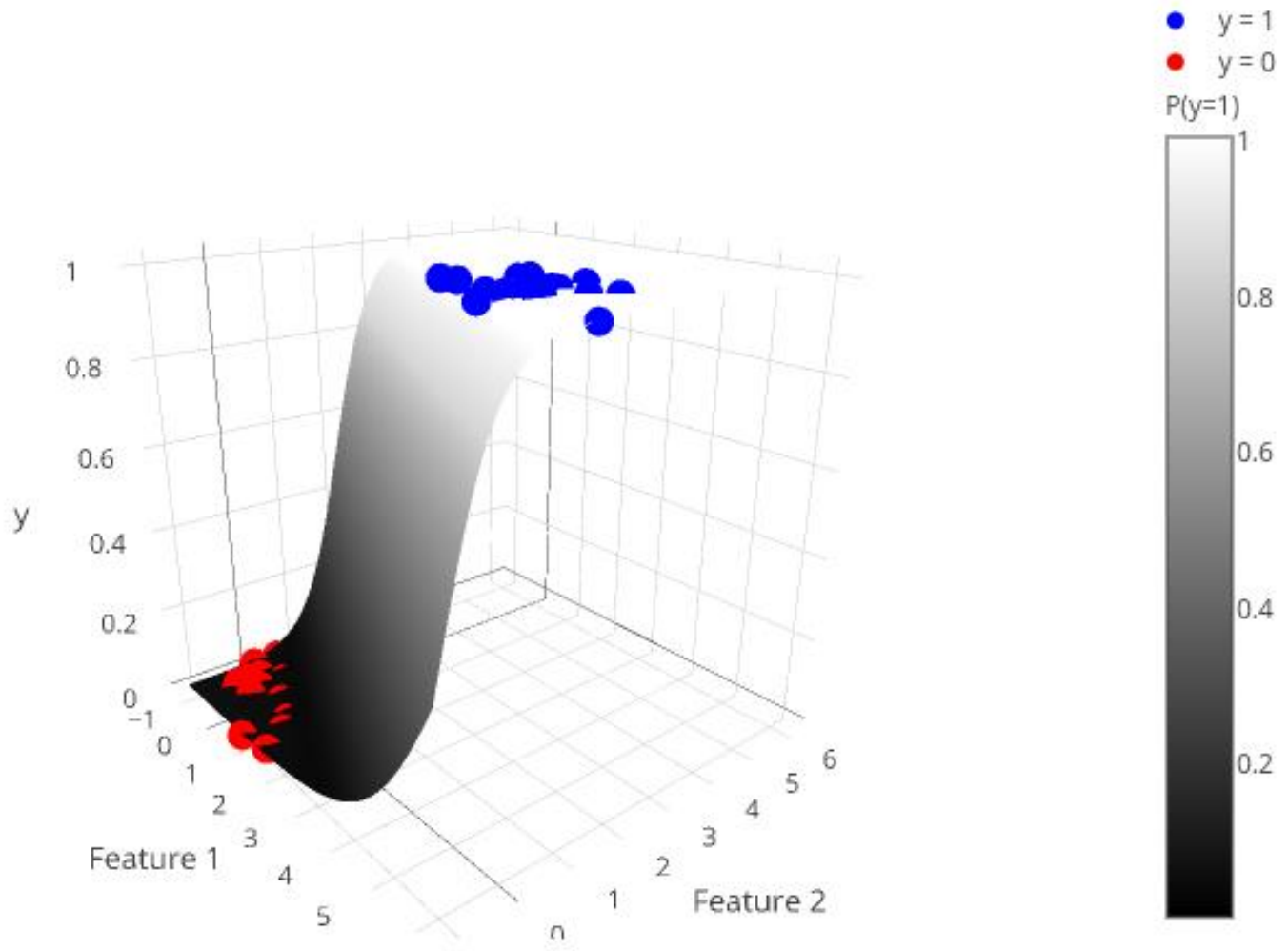
As $(\beta_0+\beta_1 x) \to \infty$, $1 + e^{-(\beta_0+\beta_1 x)} \to 1$, $p(\vec{x}) \to 1$

As $(\beta_0+\beta_1 x) \to -\infty$, $1 + e^{-(\beta_0+\beta_1 x)} \to \infty$, $p(\vec{x}) \to 0$

$\beta_0 + \beta_1$ are discovered during the machine learning process and are the result of finding the parameters of the **hyperplane** that best separates the data.

Logistic regression is regarded as a **linear model,** as the parameters for the model are a linear combination.

LOGISTIC REGRESSION WITH TWO SUPPORT VARIABLES



Decision Trees

DECISION TREE

Decision trees can be used for both classification and regression.

A decision tree is a flowchart-like structure in which each internal node represents a ‘test’ on an attribute (e.g., whether a coin flip comes up heads or tails).

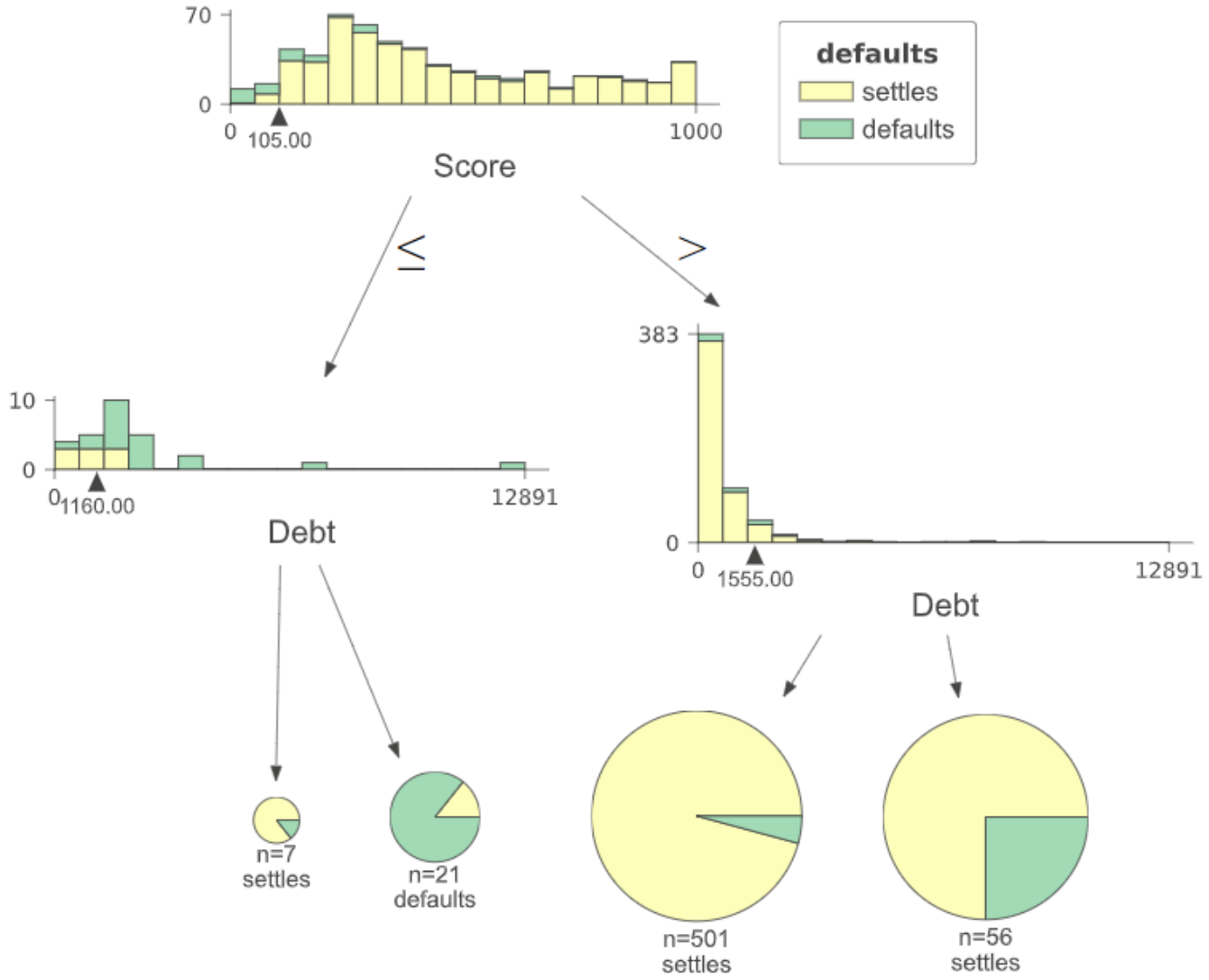
Each branch represents the outcome of the test, and each leaf node represents a class label or an interval, if using for regression.

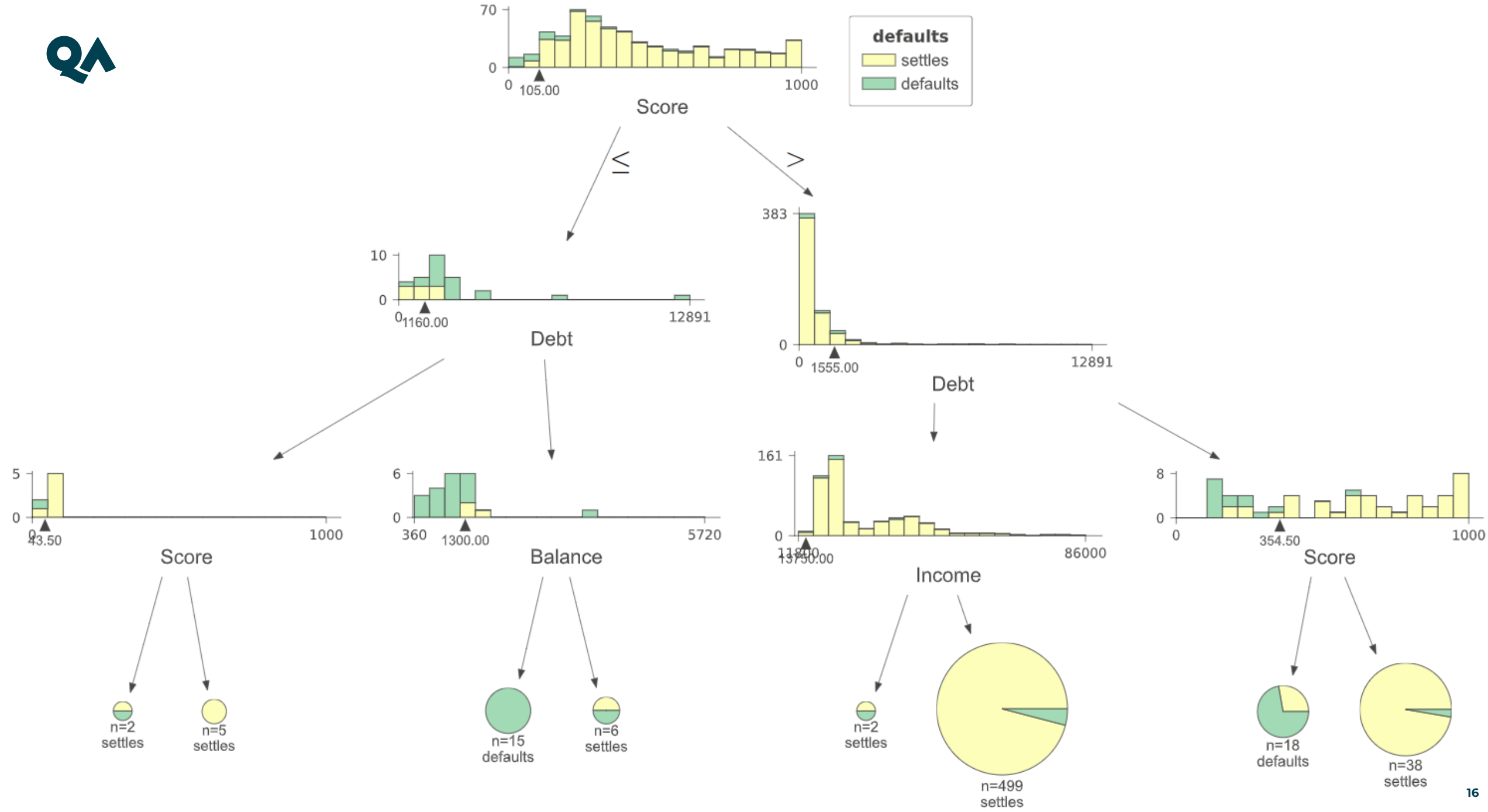
Decision tree for classification

An illustration of a decision tree on the loans data.

This tree has a max depth of 2.

Max depth is the amount of ‘questions’ we can ask of the data.




defaults
settles
defaults
70
0
0
105.00
1000
Score
≤
>
10
0
0
1160.00
12891
Debt
383
0
0
1555.00
12891
Debt
5
0
0
43.50
1000
Score
6
0
360
1300.00
5720
Balance
161
0
118
13750.00
86000
Income
8
0
0
354.50
1000
Score
n=2
settles
n=5
settles
n=15
defaults
n=6
settles
n=2
settles
n=499
settles
n=18
defaults
n=38
settles

THE MATHS

We will look at how the Gini Impurity is calculated for decision trees, but there are other metrics that can be used, for example:

Classification:

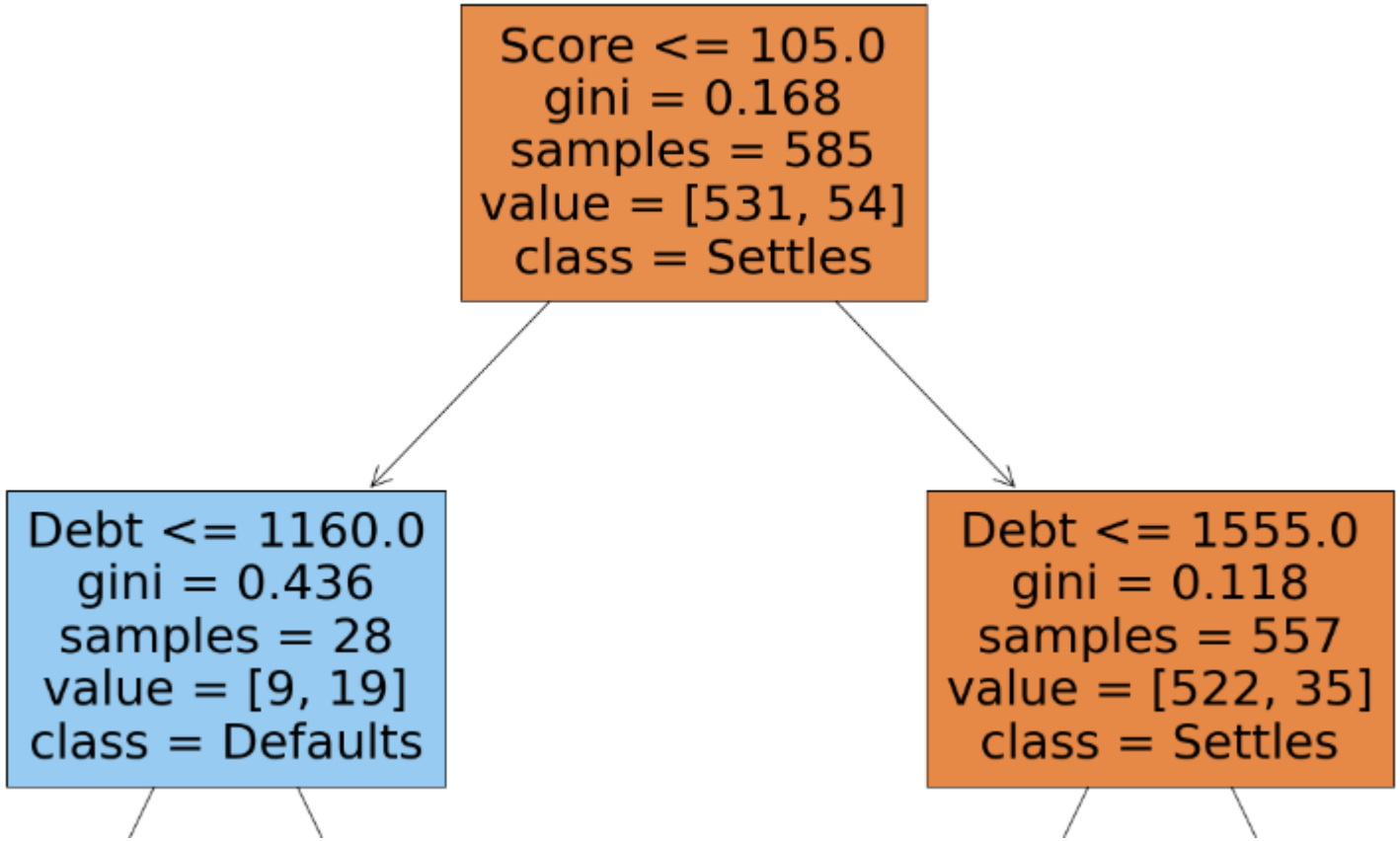
- Gini Impurity <- Default in Scikit Learn
- Entropy or Information Gain
- Log Loss

Regression:

- Squared Error <- Default in Scikit Learn
- Friedman Mean Squared Error
- Absolute Error

Further reading:
Scikit-Learn

Application in loans data

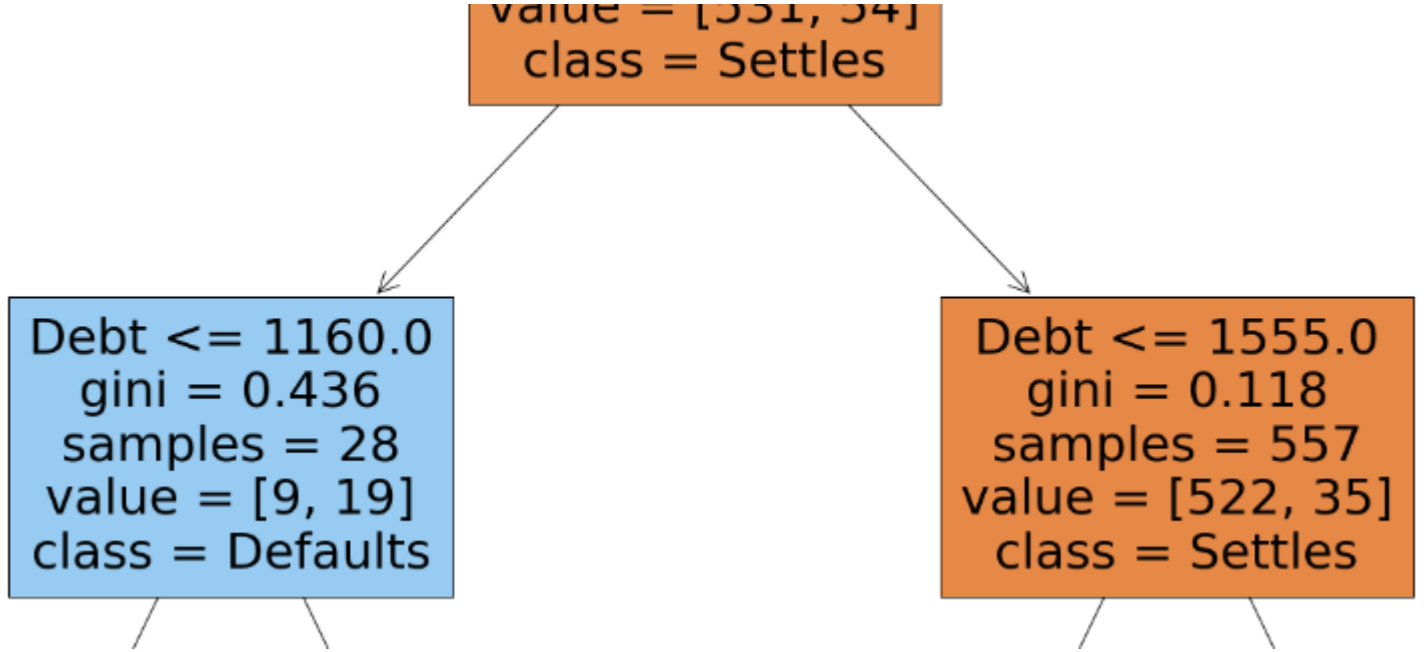


The Gini for the first node is easily calculated:

Gini = 2*(54/585)(531/585) = 0.168.

This is our base line.

Application in loans data



The Weighted Average Gini for the split is:

$$Weighted\ Gini = \frac{28(0.436) + 557(0.118)}{585}$$

$$= 0.1332$$

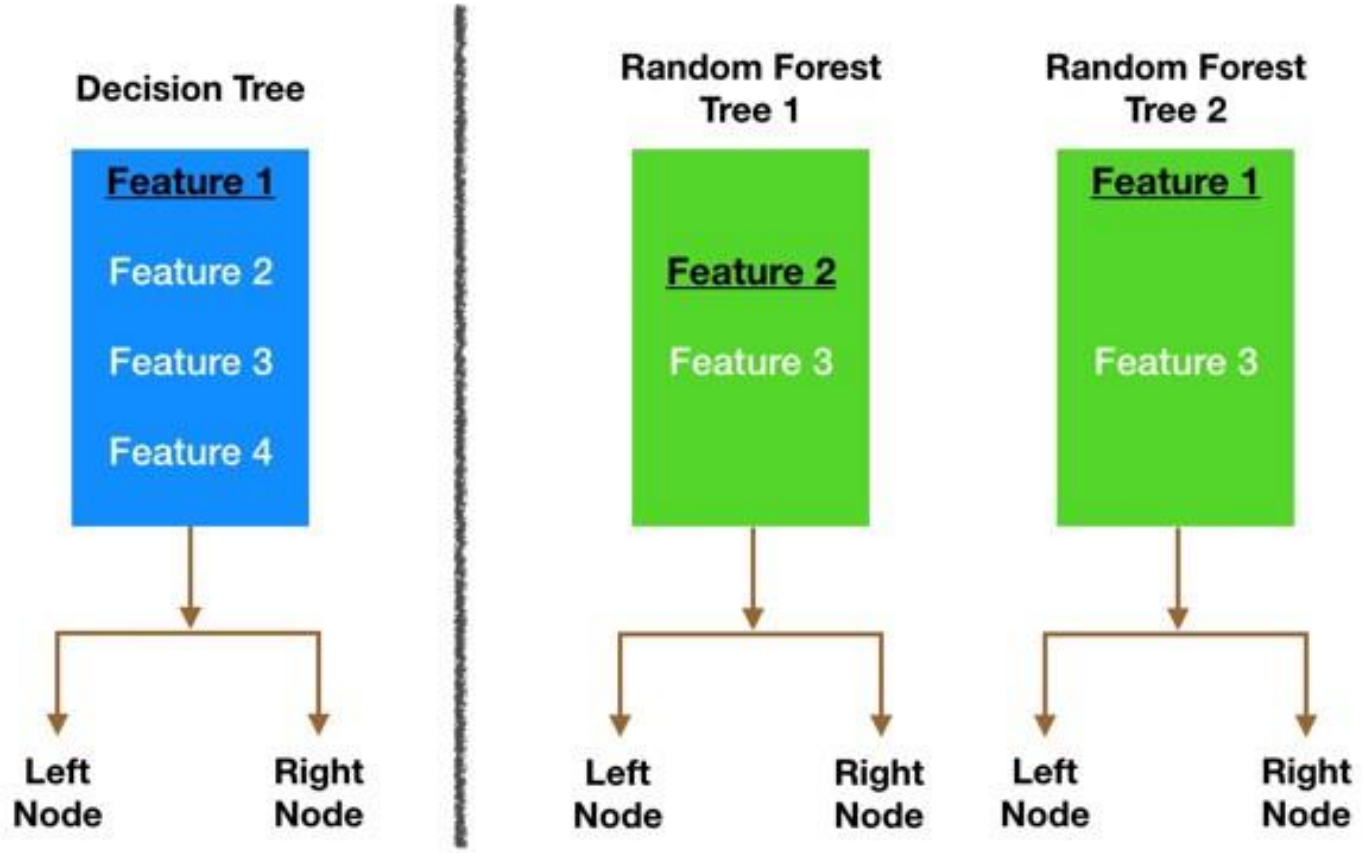
Extending Decision Trees

Random forests

A random forest takes some inherent disadvantages of decision trees and turns them into strengths.

It is an ensemble model built from a number of decision trees, hence its name.

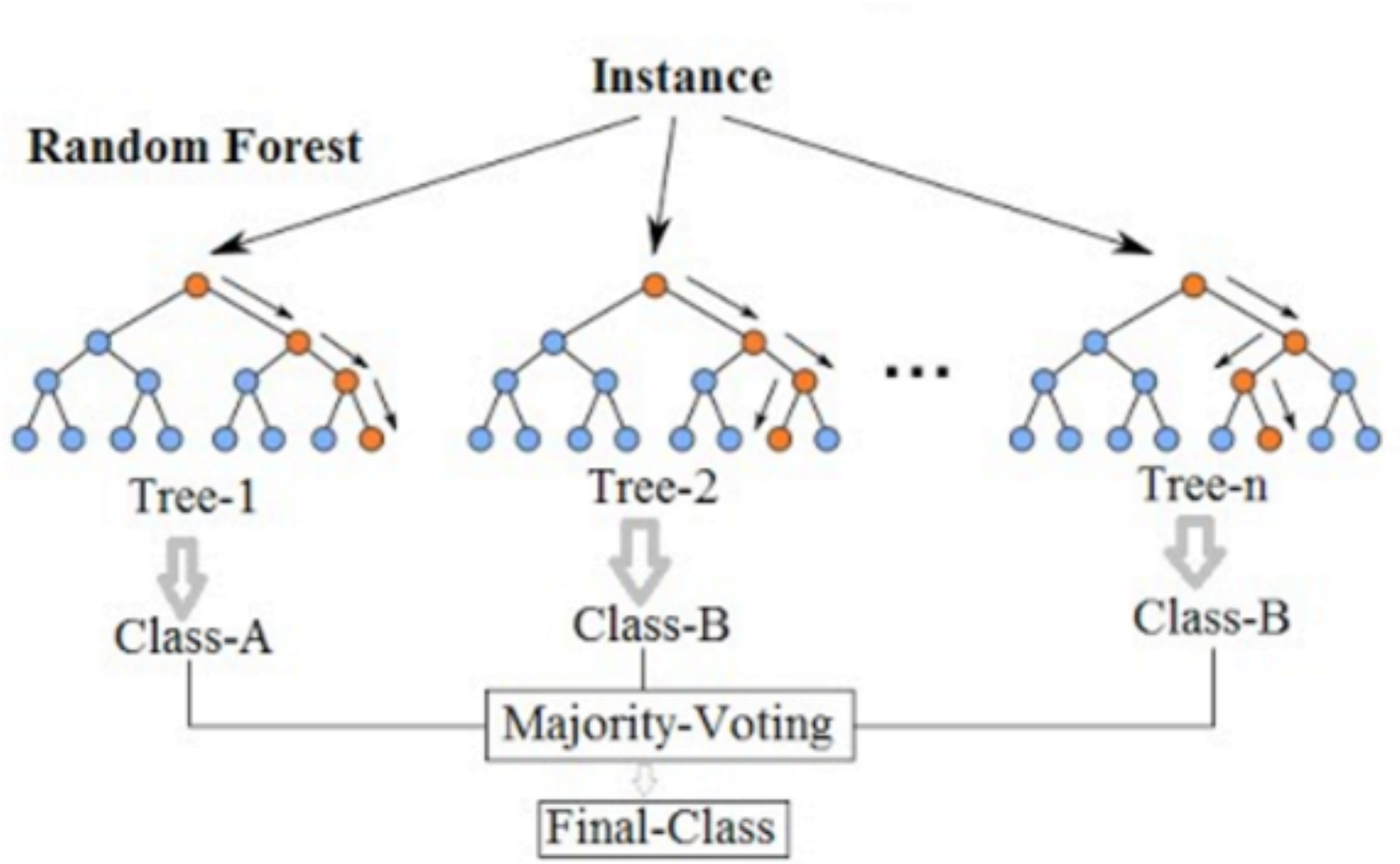
The fundamental concept behind a random forest is the ‘wisdom of crowds’; this is achievable by ensuring a diverse, or uncorrelated, forest of decision trees.



Random forests

Each tree in the random forest trains on a subset of the data, using bagging, with random features.

Then, when making a prediction, each tree votes, and the majority decision is the final class.



Source

Evaluating classification

CLASSIFICATION METRICS: RIGHTS AND WRONGS

True Positive (TP): A correct guess of the positive class

False Positive (FP): An incorrect guess of the positive class

True Negative (TN): A correct guess of the negative class

False Negative(FN): An incorrect guess of the negative class

Confusion Matrix

		Predicted Value	
		Positive	Negative
Actual Value	Postitive	True Positive	False Negative
	Negative	False Positive	True Negative

		Predicted Value	
		Positive	Negative
Actual Value	Postitive	100	30
	Negative	2	70

CLASSIFICATION METRICS: EVALUATION

Accuracy

$$Accuracy = \frac{\#\ of\ correct\ predictions}{\#\ of\ predictions} = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

$$Precision = \frac{\#\ of\ correctly\ predicted\ positives}{\#\ of\ predicted\ positives} = \frac{TP}{TP + FP}$$

Recall

$$Recall = \frac{\#\ of\ correctly\ predicted\ positives}{\#\ of\ actual\ positives} = \frac{TP}{TP + FN}$$

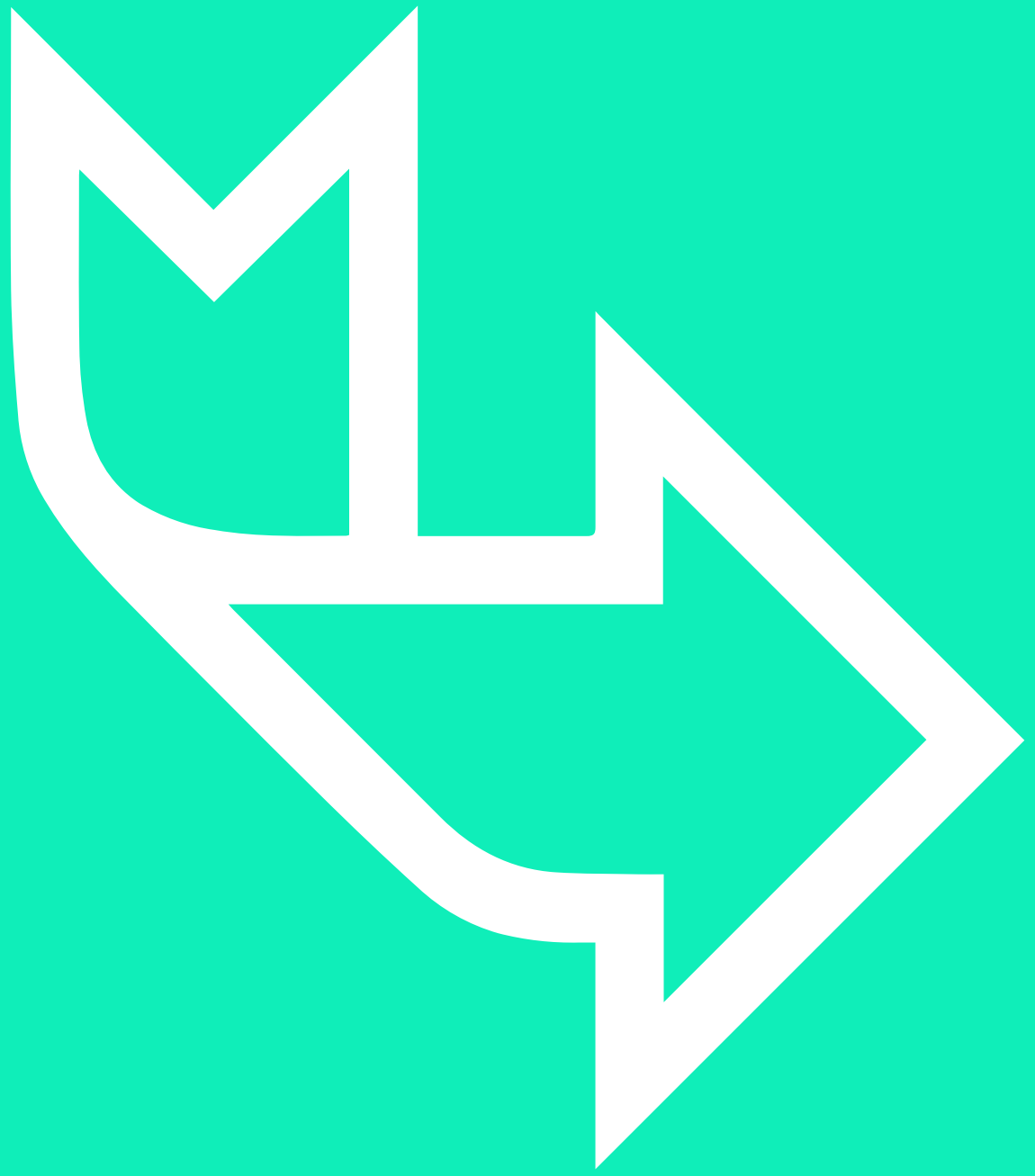
F1 Score

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

CLASSIFICATION METRICS: EVALUATION EXAMPLE

RandomForestClassifier(min_samples_leaf=10, n_estimators=30)

	precision	recall	f1-score	support
Settle	0.95	0.99	0.97	236
Default	0.73	0.38	0.50	21
accuracy			0.94	257
macro avg	0.84	0.68	0.73	257
weighted avg	0.93	0.94	0.93	257



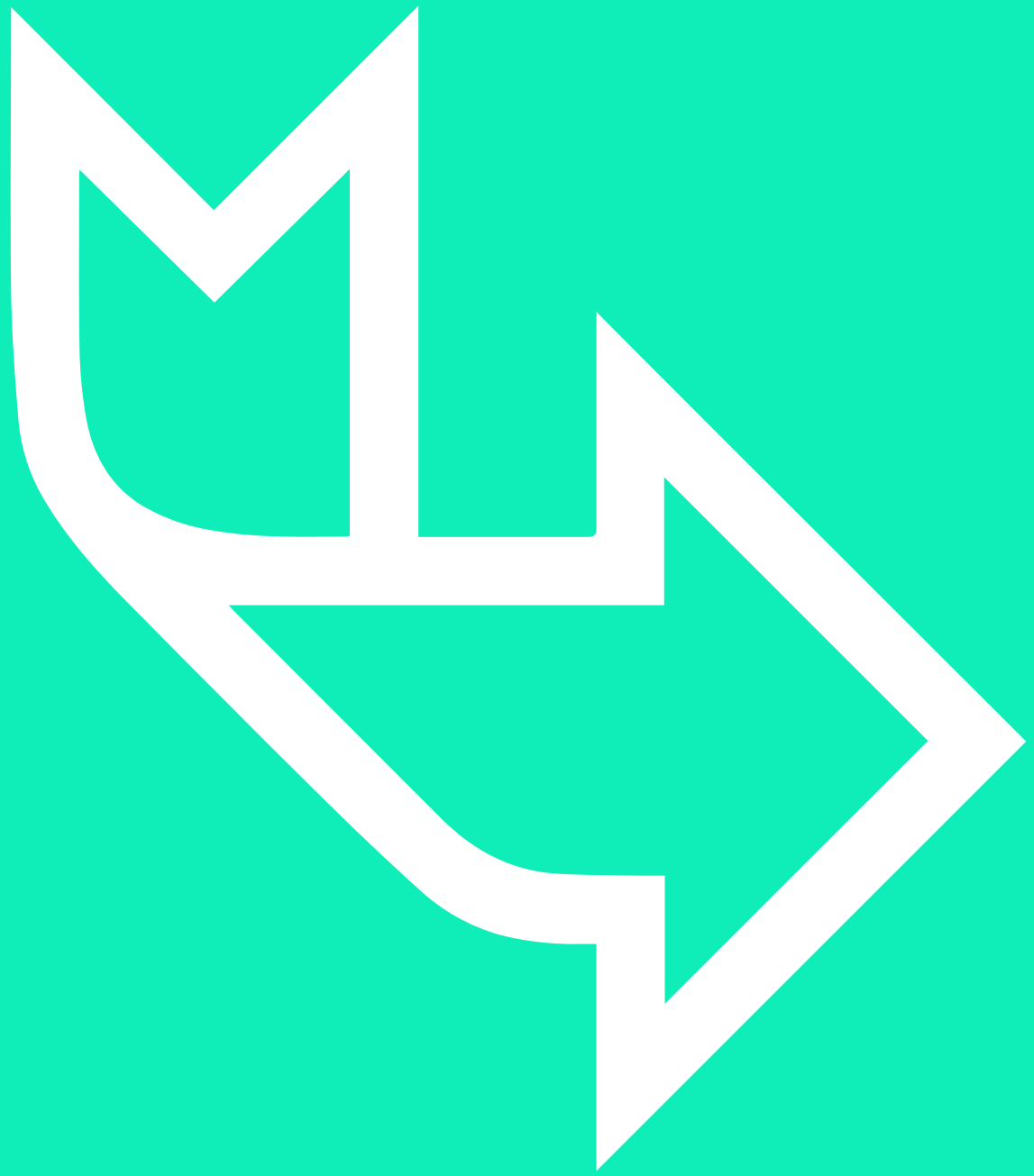
EXERCISE

Work though Module 6 Exercises: Classification

LEARNING CHECK

Think about your answers to these questions:

- What is classification?
- How do logistic regression models classify data?
- How do decision tree and random forest models classify data?
- How can we evaluate classification models?

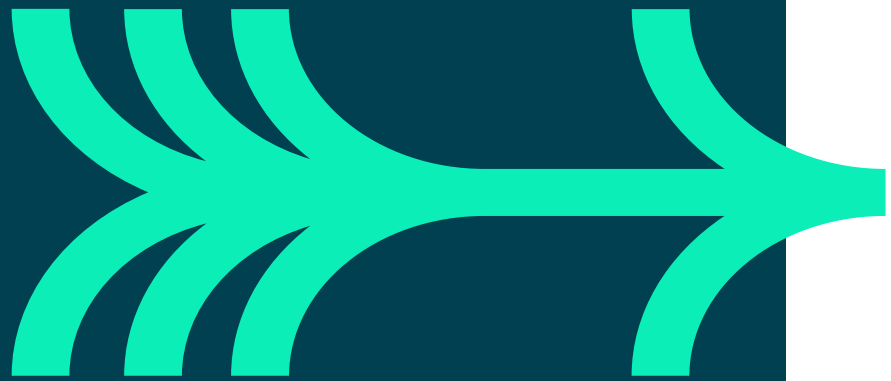


HOW DID YOU GET ON?

Learning objectives

- Describe classification in the context of machine learning.
- Build simple and multiple logistic regression models for classification.
- Build Decision Tree and Random forest models for Classification.
- Evaluate and compare classification models.

MODEL SELECTION AND EVALUATION



Learning objectives

- Understand how to choose the best model for regression and classification problems.
- Consider tests and baselines that can be used to evaluate model performance and behaviour.
- Evaluate ‘how good is good enough’.

Expected prior knowledge

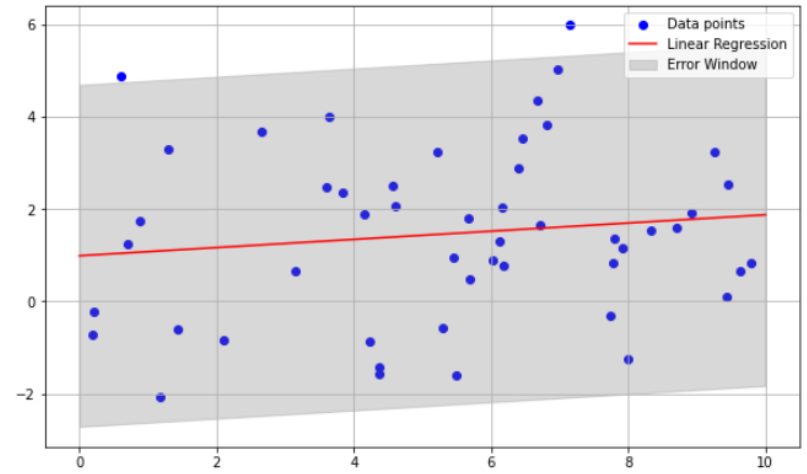
- Nothing is assumed about your background.

Evaluating model performance

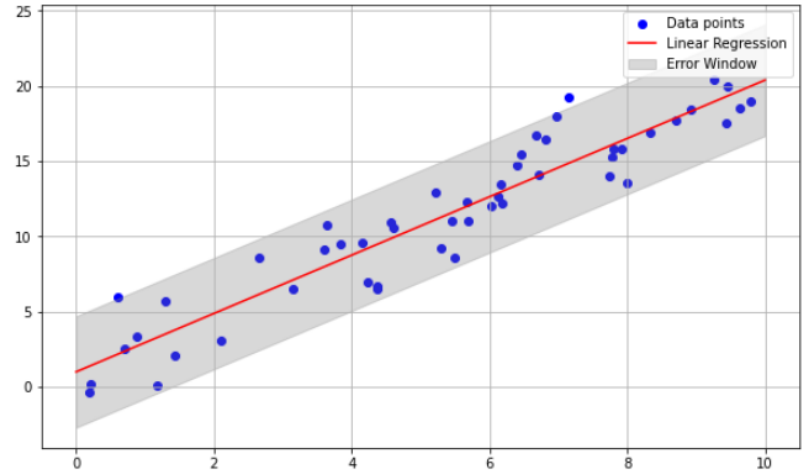
REGRESSION METRICS: USEFULNESS

R^2 is the **coefficient of determination**. It can be used as a measure of **usefulness** of the model. It's a measure of the variance of the dependent variable, y.

It's on a scale of 0 (bad) to 1 (good).



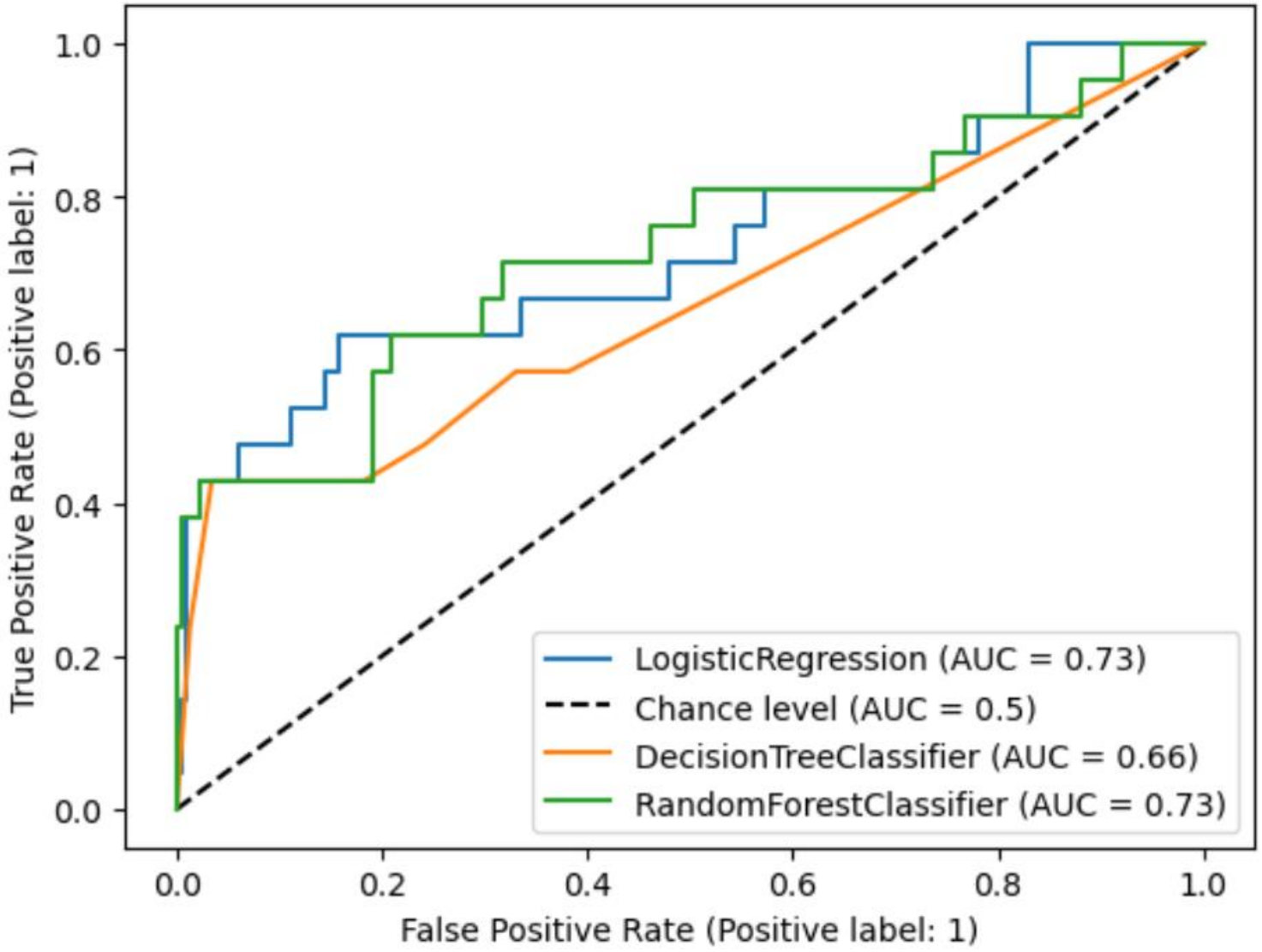
R Squared = 0.02



R Squared = 0.89

Consider the predicted y values in these cases – if you predict from different x inputs, does it make much difference to what is predicted?

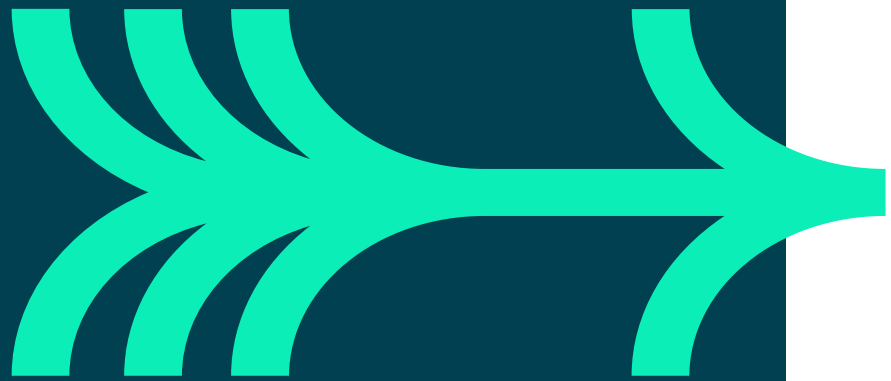
CLASSIFICATION METRICS: EVALUATION



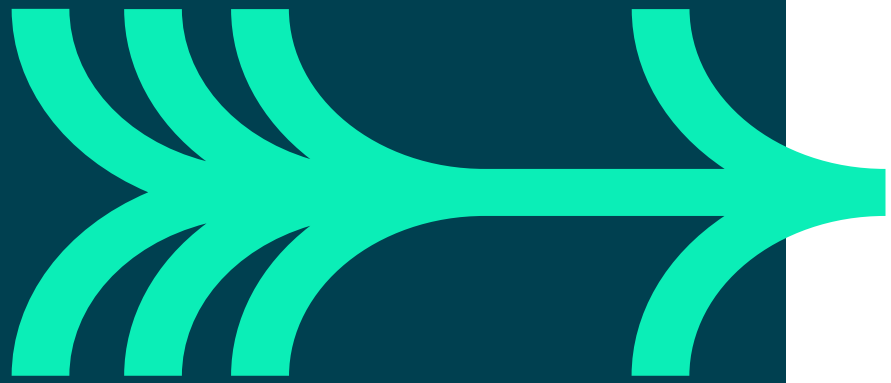
How good is good enough?

BASELINES

- Random
- Simple Heuristic
- Zero Rule
- Human
- Existing Solution



EVALUATING MODELS IN PRODUCTION



Perturbation Tests

- Evaluate how noise might impact model.
- Useful if high noise expected when deployed.

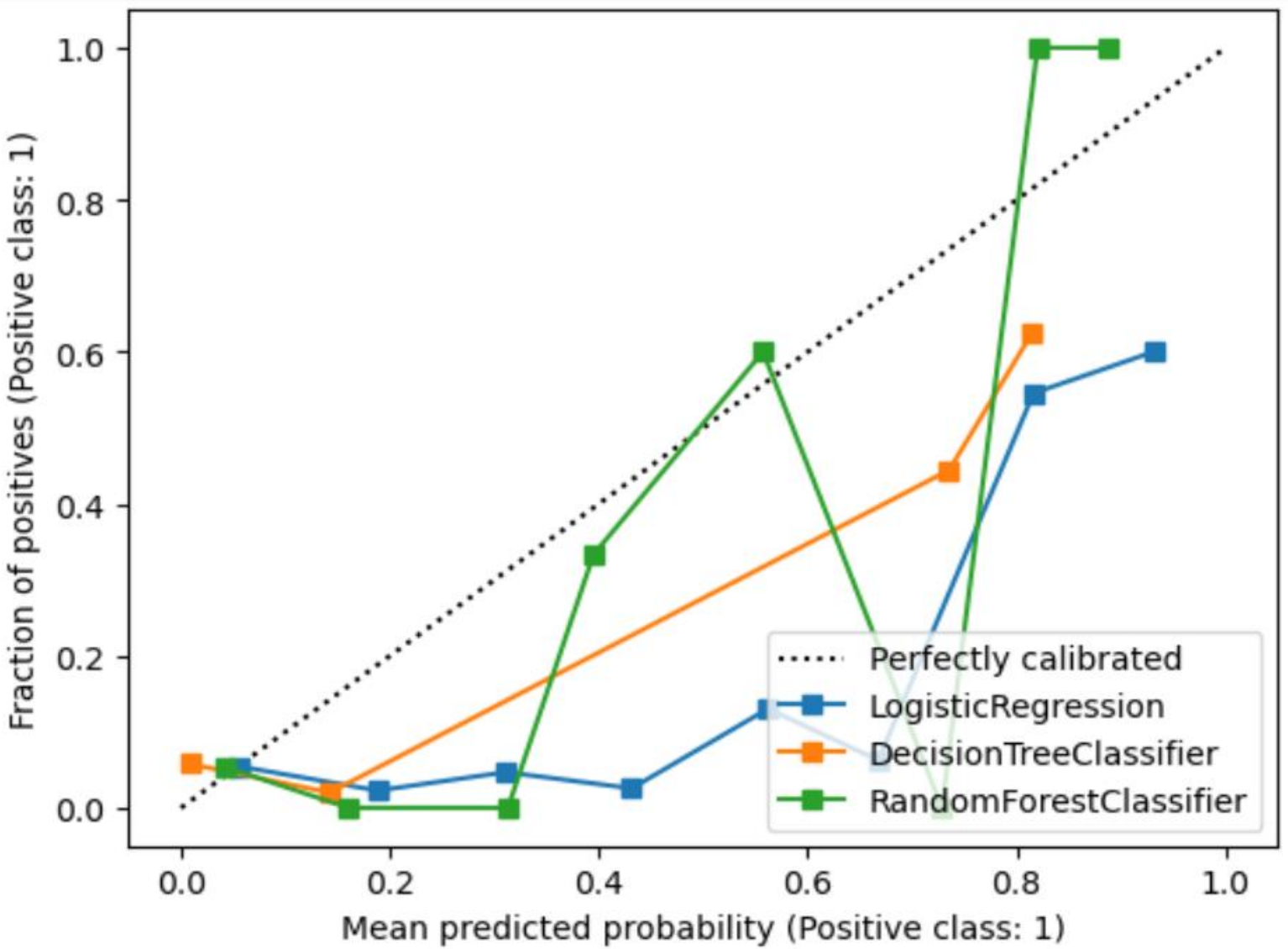
Invariance Tests

- Identify whether features that should not influence model in fact do.
- Especially useful if ethical bias possible.

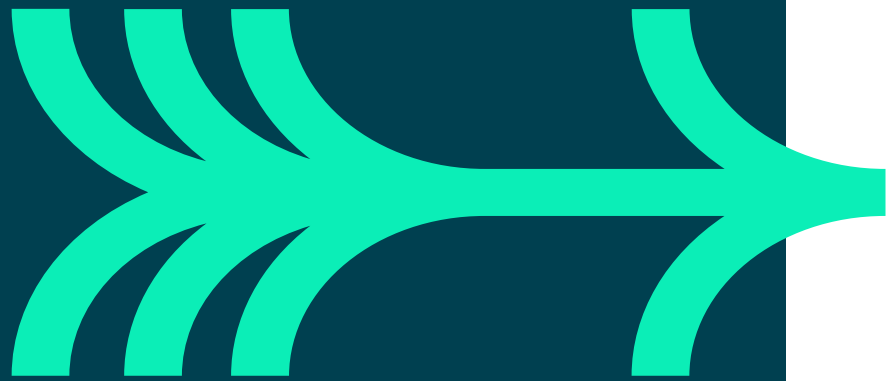
Directional Expectation Tests

- Check model behaves as expected.

MODEL CALIBRATION



ARE ALL PREDICTIONS EQUAL?

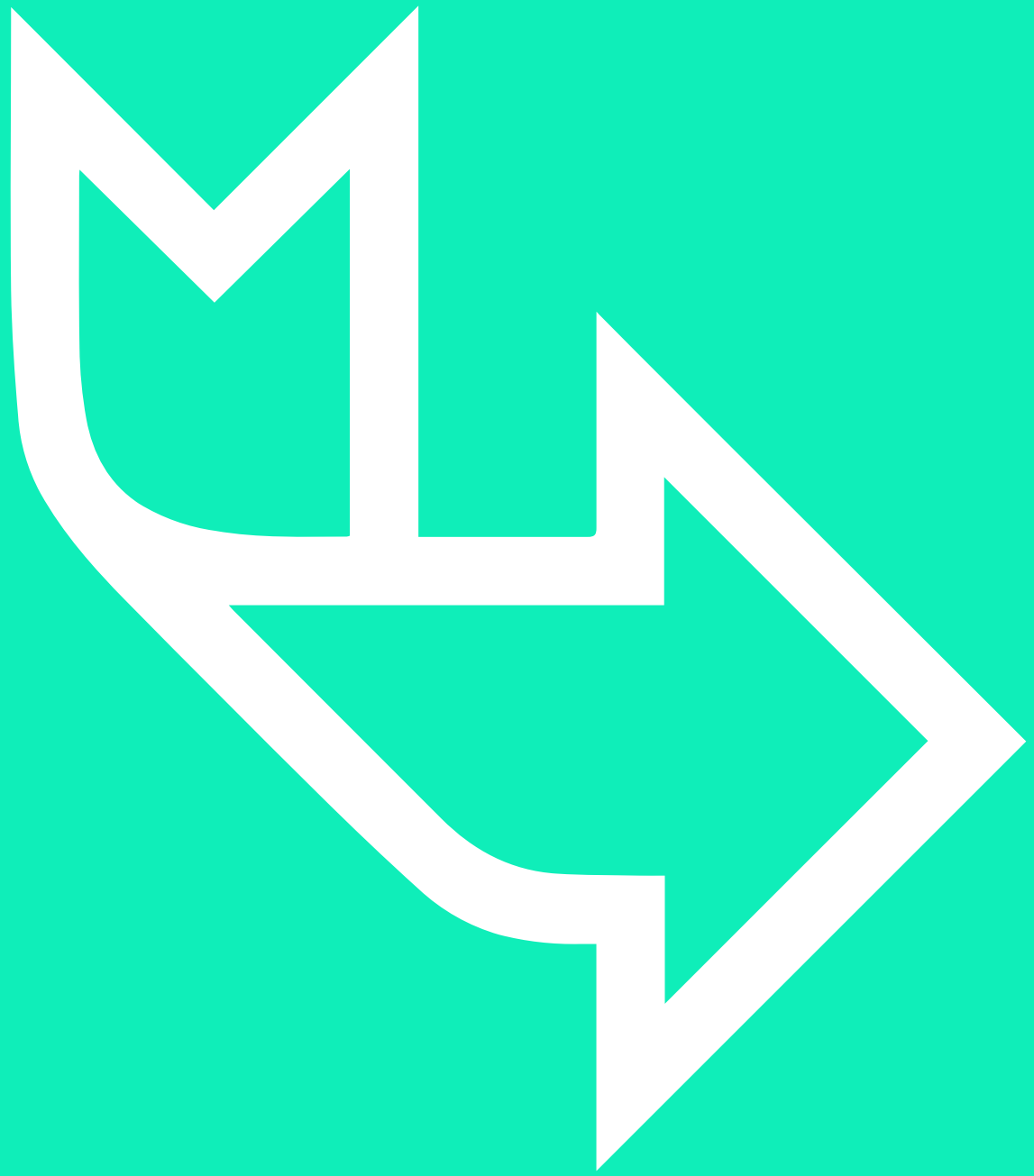


Confidence Measurement

- Are low confidence predictions worth giving to user?

Slice-Based Evaluation

- Are subgroups being predicted equally?
- A better model for subgroups could appear worse overall (Simpson).



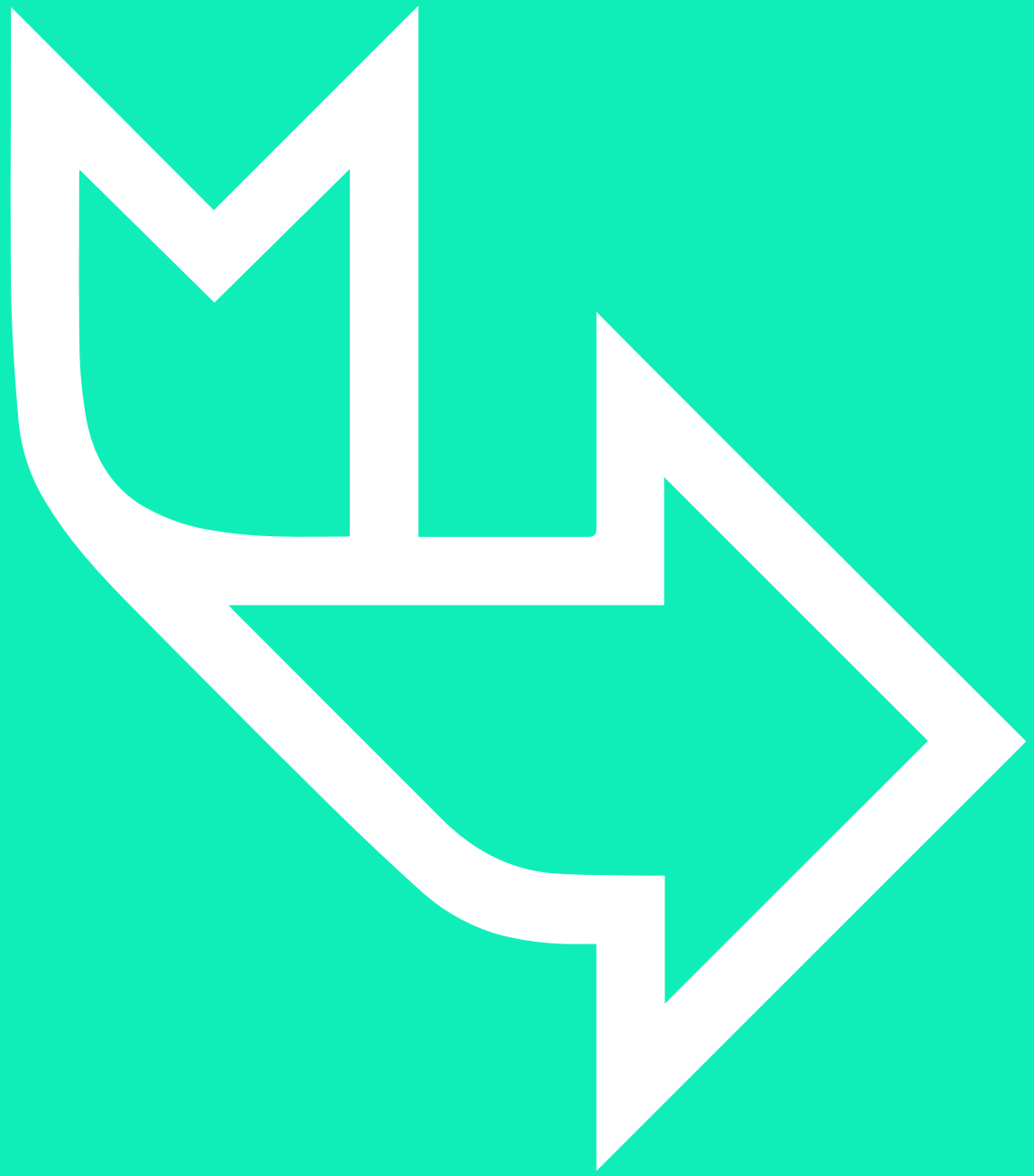
EXERCISE

Work though Module 7 Exercises: Model Selection and Evaluation

LEARNING CHECK

Think about your answers to these questions:

- How do we choose the best model for regression and classification problems?
- Which tests and baselines can be used to evaluate model performance and behaviour?
- How good is good enough?

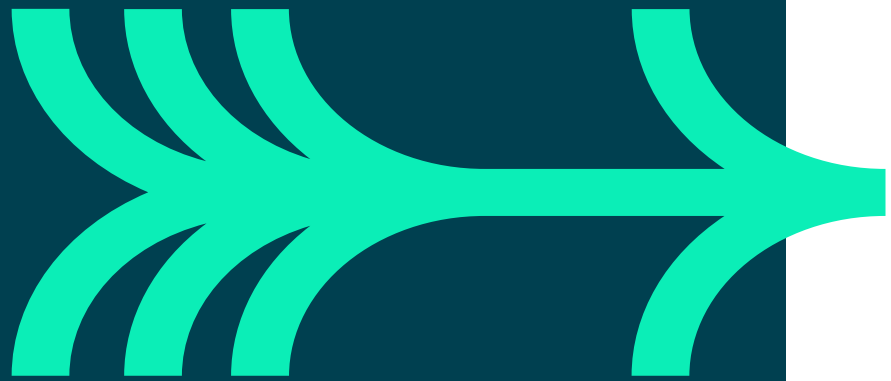


HOW DID YOU GET ON?

Learning objectives:

- Understand how to choose the best model for regression and classification problems.
- Consider tests and baselines that can be used to evaluate model performance and behaviour
- Evaluate ‘how good is good enough’.

UNSUPERVISED LEARNING



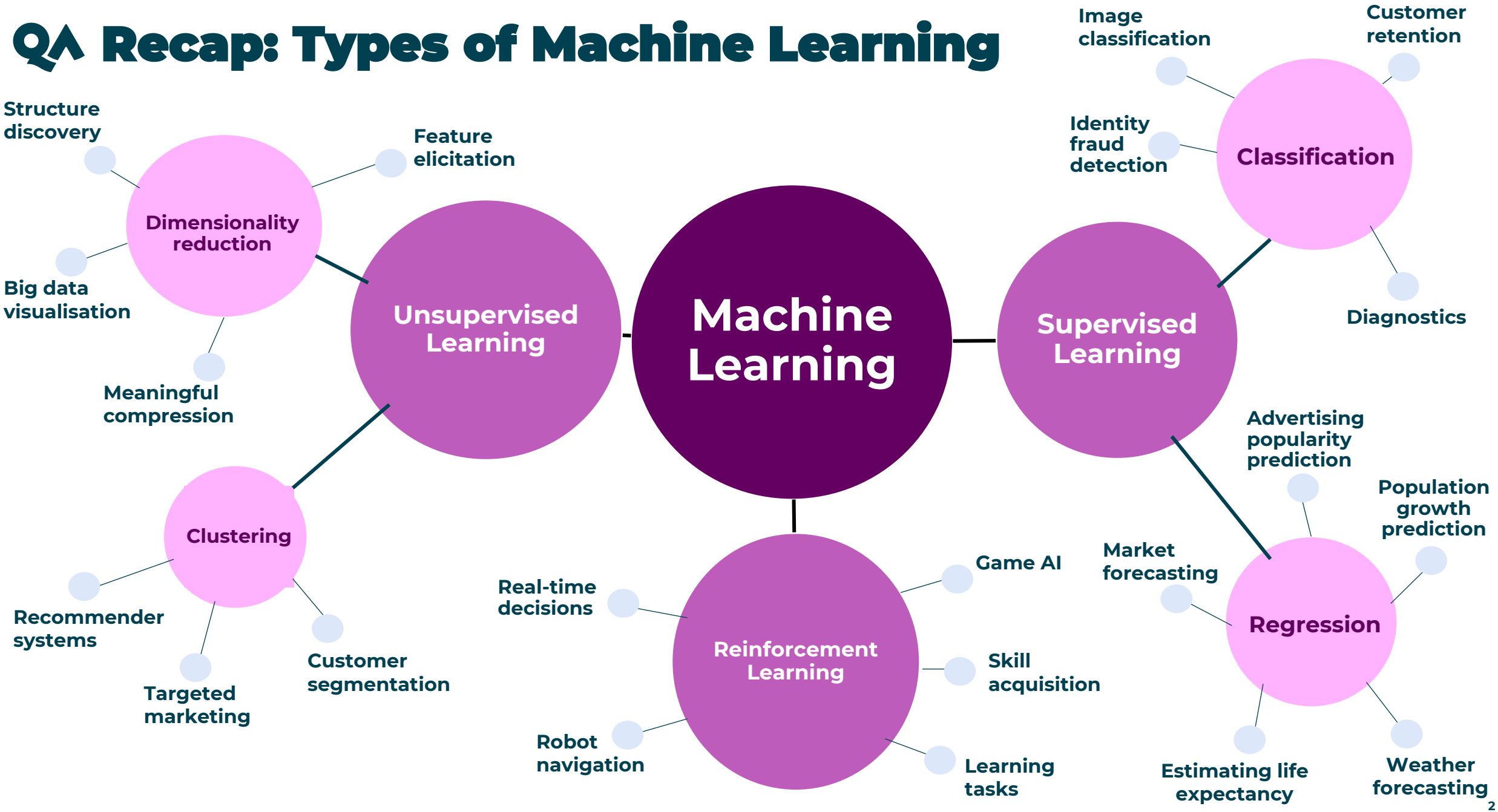
Learning objectives

- Describe clustering and dimensionality reduction in the context of machine learning.
- Apply and evaluate KMeans clustering.
- Apply and evaluate dimensionality reduction techniques.

Expected prior knowledge

- Nothing is assumed about your background.

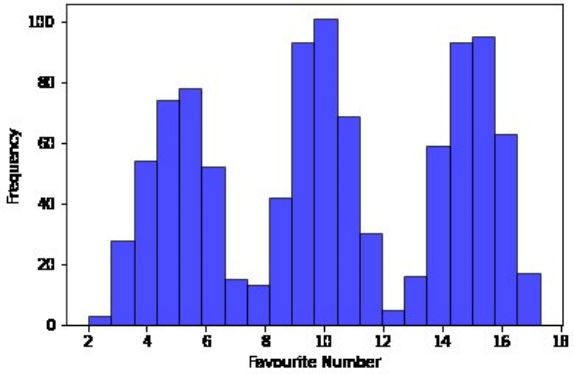
QA Recap: Types of Machine Learning



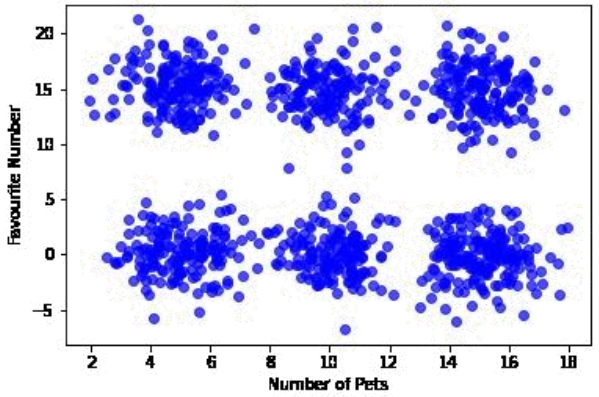
Clustering

CLUSTERING: WHAT STRUCTURE IS THERE?

One column of data:



Two columns of data:



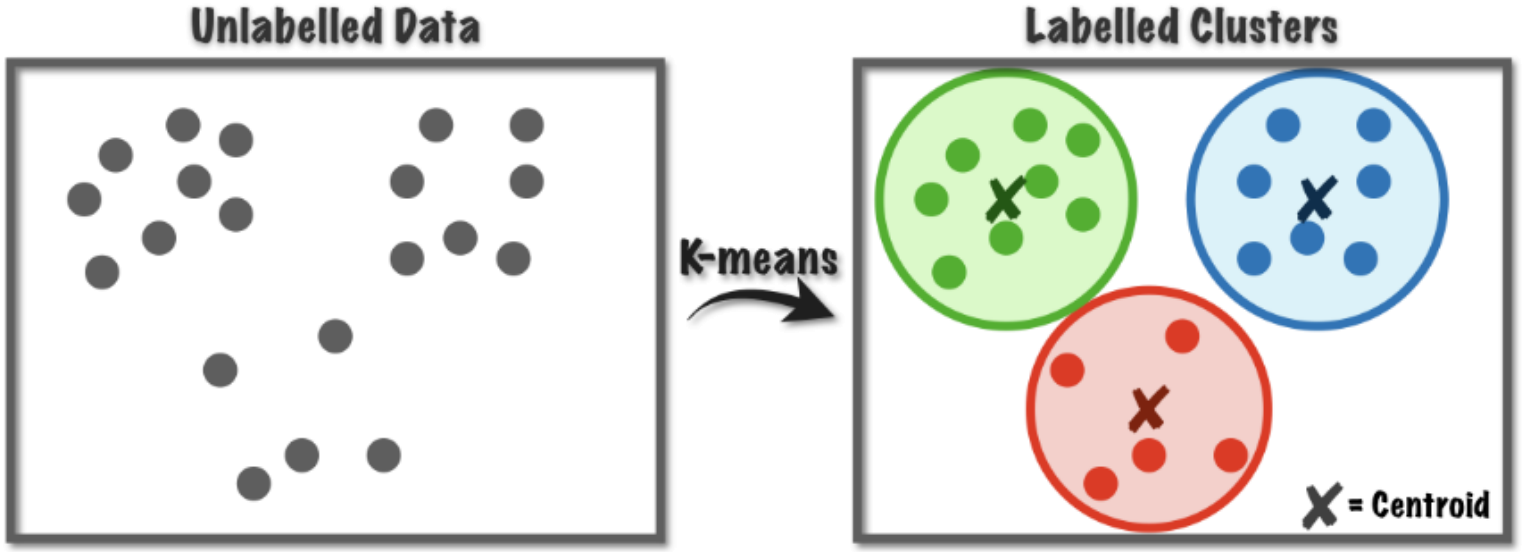
More columns of data: difficult to 'see' – this is where clustering models come in handy and why Data Science Developers created algorithms to get machines to spot what we can't.

Note: These methods are unsupervised – it detects, but doesn't explain why there are clusters.

KMeans

WHAT IS CLUSTERING?

Clustering is an unsupervised machine learning algorithm that segments groups of abstract objects – rows of data – into classes where the members share similar attributes.



Source

Business example

We have clustered customers based on their shopping habits at a major grocery retailer. To better understand each cluster, we have generated summary statistics for each one. Based on the results, derive a name for each cluster that can be used for communication. Do you think this clustering solution is interpretable? Actionable?

Type	Metric	Cluster 1	Cluster 2	Cluster 3
Clustering variables	% spend on fruit and veg	15%	15%	23%
	% spend on frozen food	22%	12%	5%
	% spend on anything else	63%	73%	72%
Profiling variables	Average age	44.3	38.8	36.7
	Average household size	3.2	2.1	1.8
	% shopping trips which are at the weekend	45%	23%	19%
	Average spend per transaction	£80	£43	£72

USE CASES FOR CLUSTERING

Examples:

1. Retail marketing
2. Streaming services
3. Sports science
4. Email marketing
5. Healthcare

What other examples can you think of?

THE K-MEANS ALGORITHM

In simple terms, the K-Means Algorithm measures the distance between every point and the current guess of the centre of a group.

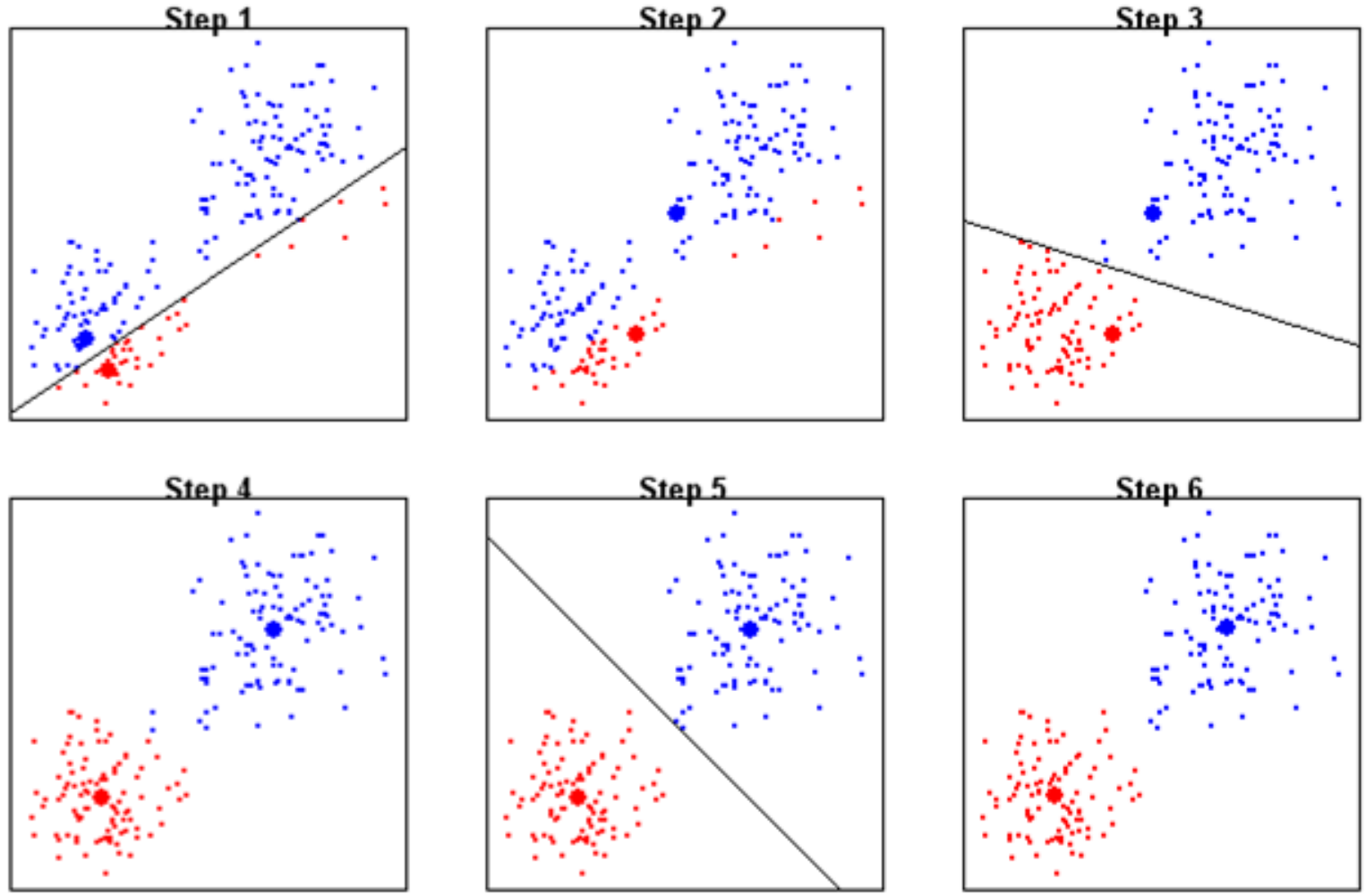
It allocates all points closest to that group to it, recalculates the centre, and then repeats. It iterates through this process until either a pre-defined limit has been reached, or there are no more changes.

This will be much better illustrated with an example!

It is important to note here that a crucial part of the algorithm is to know how many groups there are.

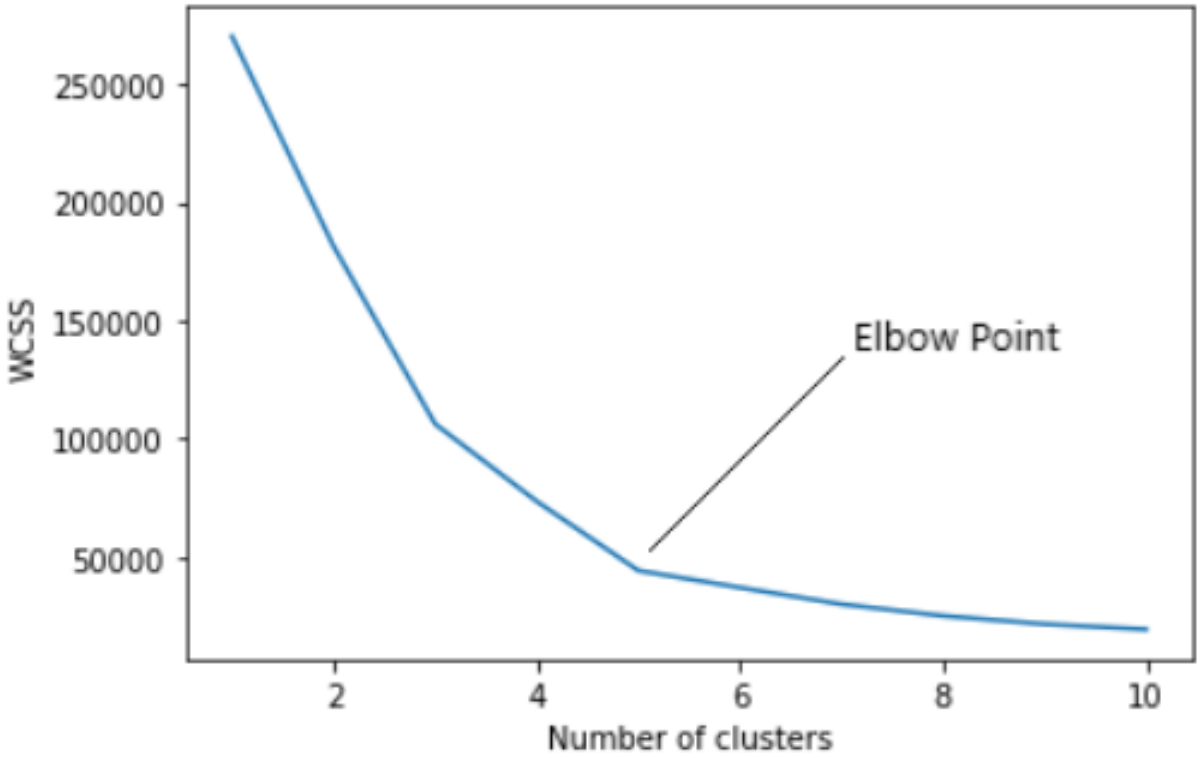
This is a Hyperparameter for the model, meaning it is something that we set.

K-MEANS ALGORITHM



THE K-MEANS ALGORITHM

HOW DO WE CHOOSE HOW MANY CLUSTERS?



WCSS is Within Cluster Sum of Squares, a common measure used to be used in K-means.

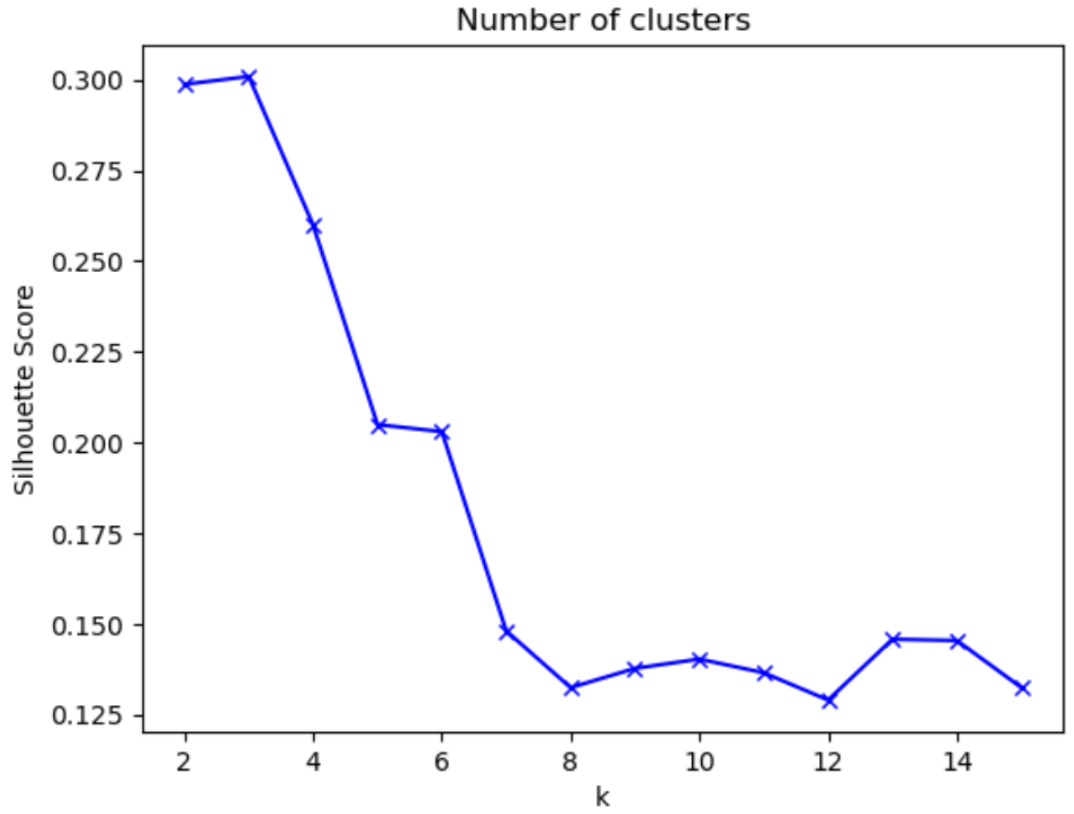
It is simply the Sum of Squares distance of each point to the centroid.

Source

SILHOUETTE SCORE

Describes the degree to which points look like they belong to their assigned cluster

- +1 means all points appear perfectly clustered
- 0 means 50% of points appear to belong to other clusters
- -1 means all points appear poorly clustered



K-MEANS ADVANTAGES AND DISADVANTAGES

Advantages:

- High performance
- Easy to use
- Unlabelled data
- Result interpretation

Disadvantages

- Spherical clustering only
- Only really works with continuous data
- Very sensitive to outliers

Other clustering approaches

Other clustering models

DBScan

- Can handle complex cluster shapes.
- Can handle uneven cluster sizes.
- Does outlier removal.

Hierarchical Agglomerative Clustering (using WARD)

- Many clusters.
- Connectivity constraints.

GMM (Gaussian Mixture Modelling)

- Can't handle complex cluster shapes.
- Good for density estimation.
- Expensive.

Dimensionality reduction

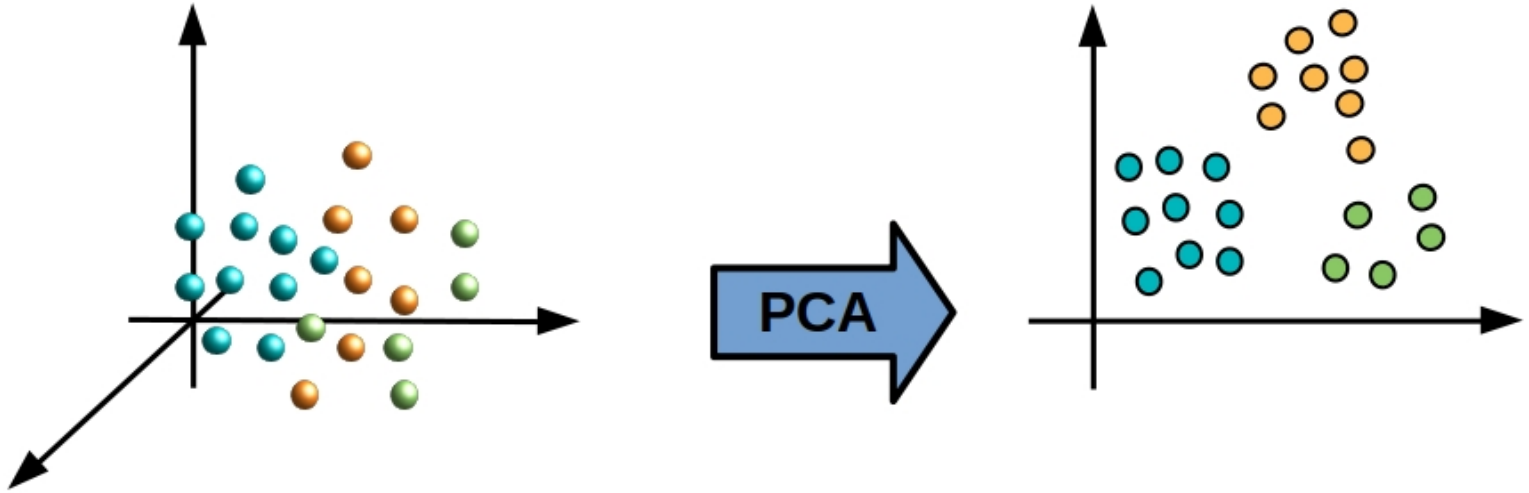
FEATURE SELECTION

The simplest way to reduce features could be to iterate through different feature sub sets and see if this either:

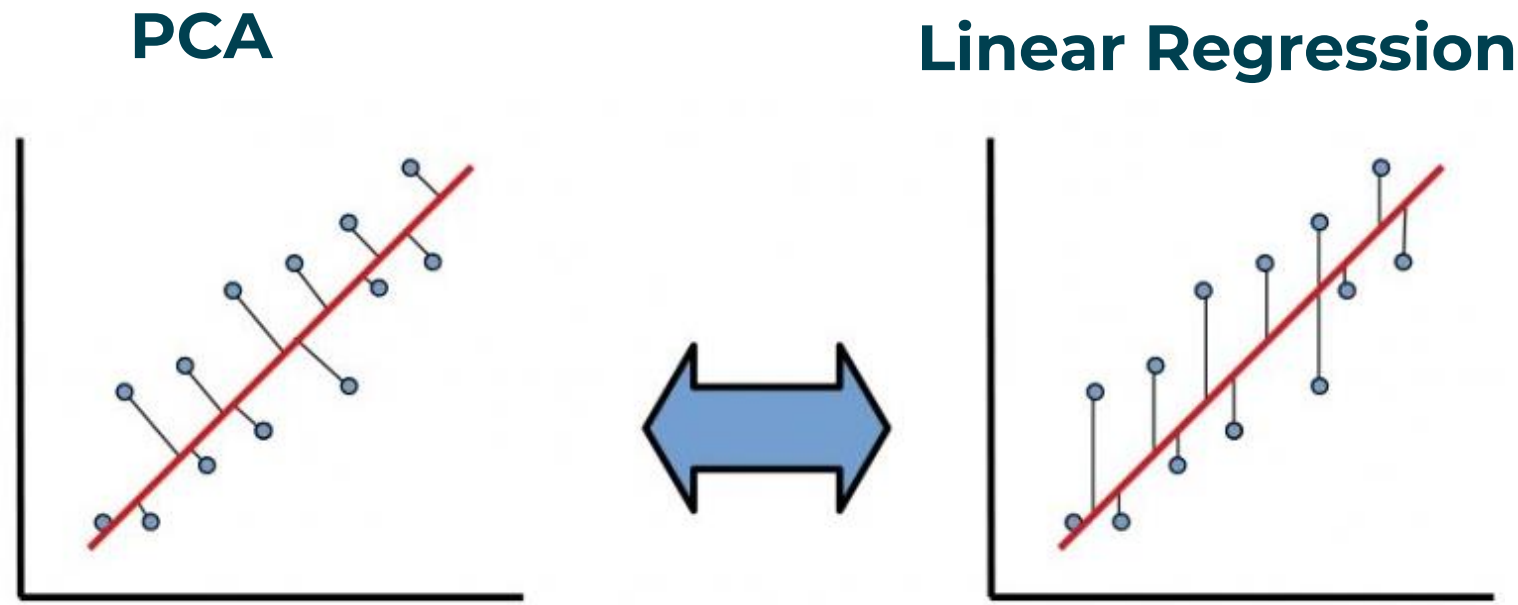
1) Increases the accuracy of your model, or
2) Does not affect the accuracy too badly and therefore may be preferable to build a simpler model.

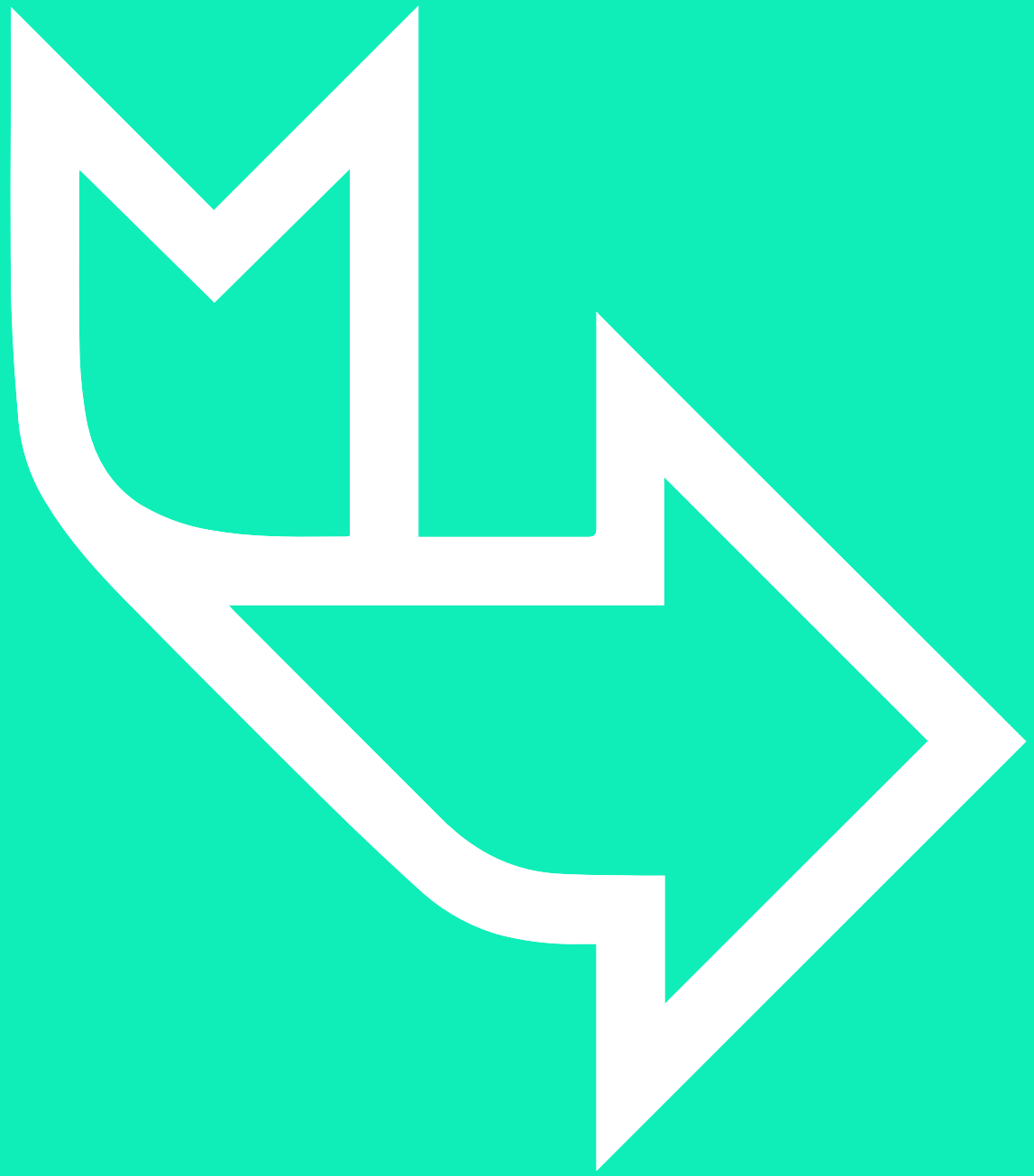
PCA IF FEATURES ARE CORRELATED

PCA from 3 dimensions to 2



PCA VS. LINEAR REGRESSION





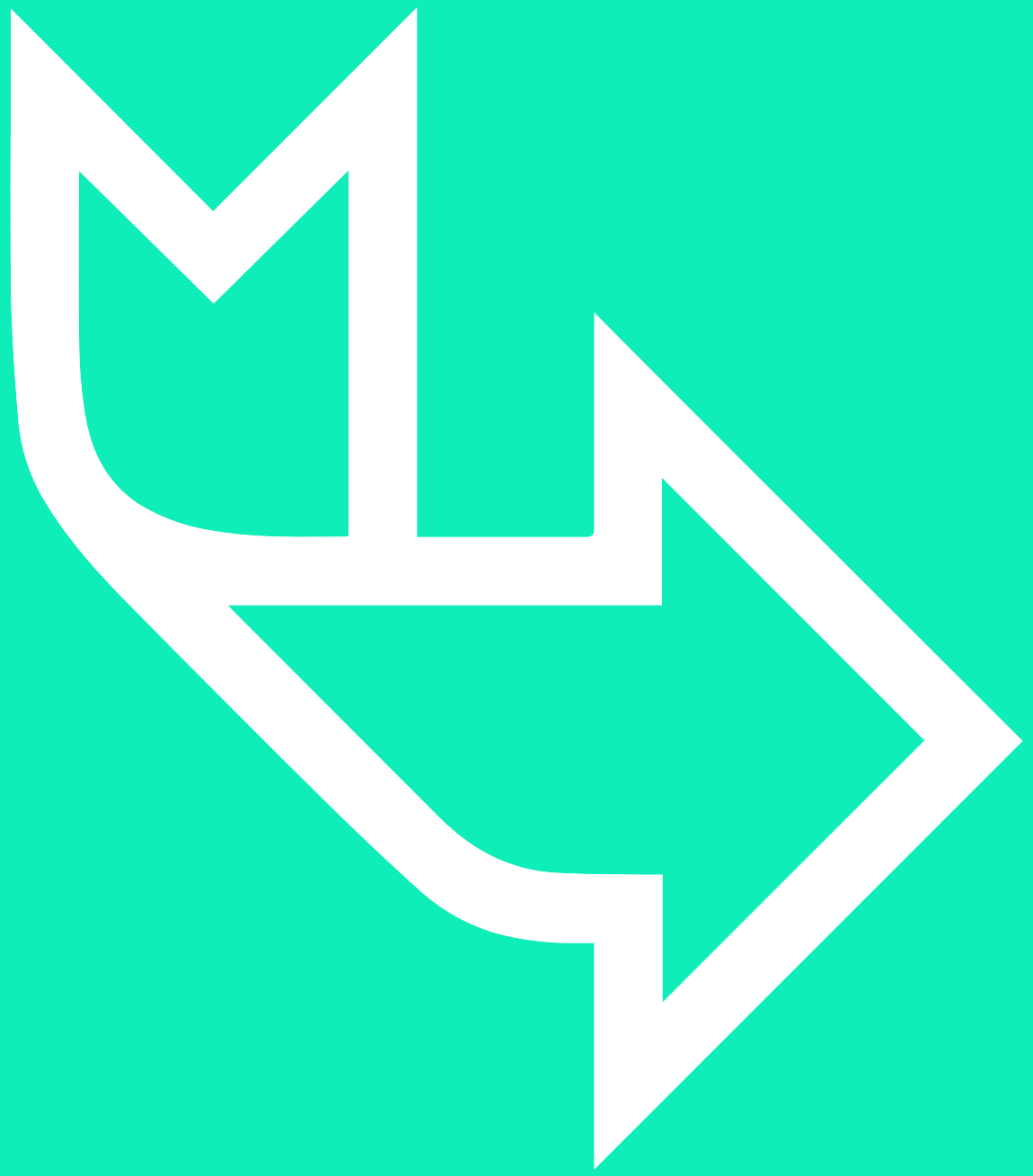
EXERCISE

Work though Module 8 Exercises: Unsupervised Learning

LEARNING CHECK

Think about your answers to these questions:

- What are clustering and dimensionality reduction in the context of machine learning?
- How does KMeans cluster data?
- How can we perform dimensionality reduction?

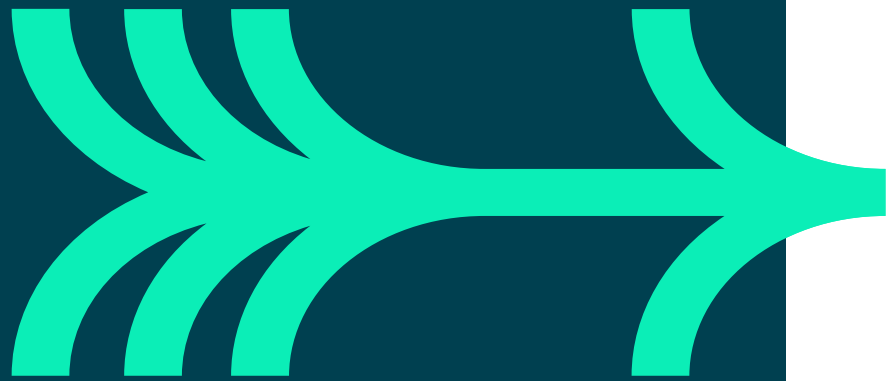


HOW DID YOU GET ON?

Learning objectives

- Describe clustering and dimensionality reduction in the context of machine learning.
- Apply and evaluate KMeans clustering.
- Apply and evaluate dimensionality reduction techniques.

ETHICS FOR DATA SCIENTISTS



Learning objectives

- Be aware of the legislation and standards Data Scientists must adhere to.
- Discuss the importance of legal, ethical, and moral considerations in Data Analytics projects and identify applicable UK legislation for which employees should receive training.
- Discuss ethical considerations for data handling.
- Recognise ethical considerations in examples of machine learning, deep learning, and AI.

Expected prior knowledge

- Nothing is assumed about your background.

UK legislation and regulatory standards for Data Analytics

Personal Data	• EU General Data Protection Regulation (GDPR) • Data Protection Act 2018 (DPA 2018)
Equality and Discrimination	• Equality Act 2010
Sharing and Re-use of data	• ICO Code of Practice for Data Sharing and guide to GDPR
Copyright and Intellectual property	• Copyright and Rights in Databases Regulations 1997 • Intellectual Property Act 2014
Freedom of Information	• Freedom of Information Act 2000 • Protection of Freedoms Act 2012
Sector Specific	• For example, UK Policy Framework for Health and Social Care Research • For example, in law enforcement and intelligence agencies: Regulatory and Investigatory Powers Act 2000 (RIPA) • For example, Financial Conduct Authority regulations
Statistics	• National Statisticians Data Ethics Advisory Committee • UK Office for Statistics Regulation (UK Gov) – Code of Practice for Statistics
Security and Information Governance	• Organisational Security Policies • Computer Misuse Act 1990
AI and ML Governance	• Regulatory development: AI Standards Hub and UK National AI Strategy (and alignment to EU AI Act)
Privacy	• Privacy and Electronic Communications Regulations 2003
Professional Standards	• Recognised and Chartered Professional accreditations usually require registrants to agree to ethical guidelines and codes of conduct • For example, Memberships and Professional recognitions from BCS, RSS, ORS, IMA.

DISCUSSION


What has gone wrong in the past?
Are there any foreseeable risks that can be mitigated or avoided?
The importance of legal and ethical considerations for Data Analytics
What are the consequences for individuals?
What are the consequences for organisations?

COMPLIANCE REQUIREMENTS

Compliance is about ensuring we follow the rules and regulations – which are often legally binding or may affect organisational funding.

Requirements vary by location and depend sector your organisation is part of.

Examples:

- Privacy laws
- Data Protection laws
- Accounting Standards
- Banking Supervision

ACTIVITY: COMPLYING WITH REGULATIONS

Select a regulation or law which your organisation must comply with and answer the below questions.

- **Why is it relevant to your organisation?**
- **What policies or procedures support in complying with the regulations?**
- **How and when is your organisation audited for compliance?**
- **What evidence is used to show auditors?**
- **What happens if you haven't complied?**
- **Is there a process to manage and fix non-compliance?**

AI, DEEP LEARNING, AND MACHINE LEARNING REGULATION

Under development!

UK AI Regulation will use a principles-based framework and will be implemented by sector-based regulatory bodies.

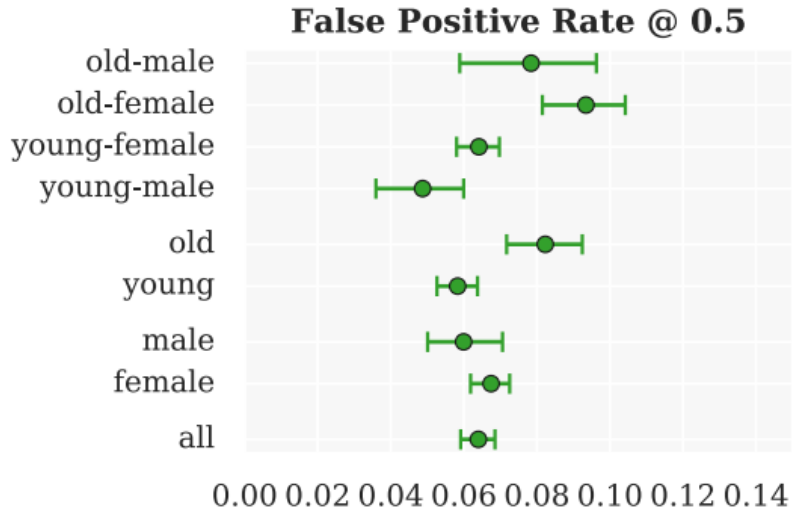
- **Safety, Security, and Robustness**
- **Appropriate Transparency and Explainability**
- **Fairness**
- **Governance and Accountability**
- **Contestability and Redress**

PRACTICAL WAYS TO MONITOR DATA ANALYTICS

When developing an Analytical or Machine Learning model, it is vital that there is a diverse range of human oversight.

One practical way to support this is using Model Report Cards, which Data Scientists can submit to a Data Governance panel.

For automated models, the use of reporting and dashboarding tools may be useful.



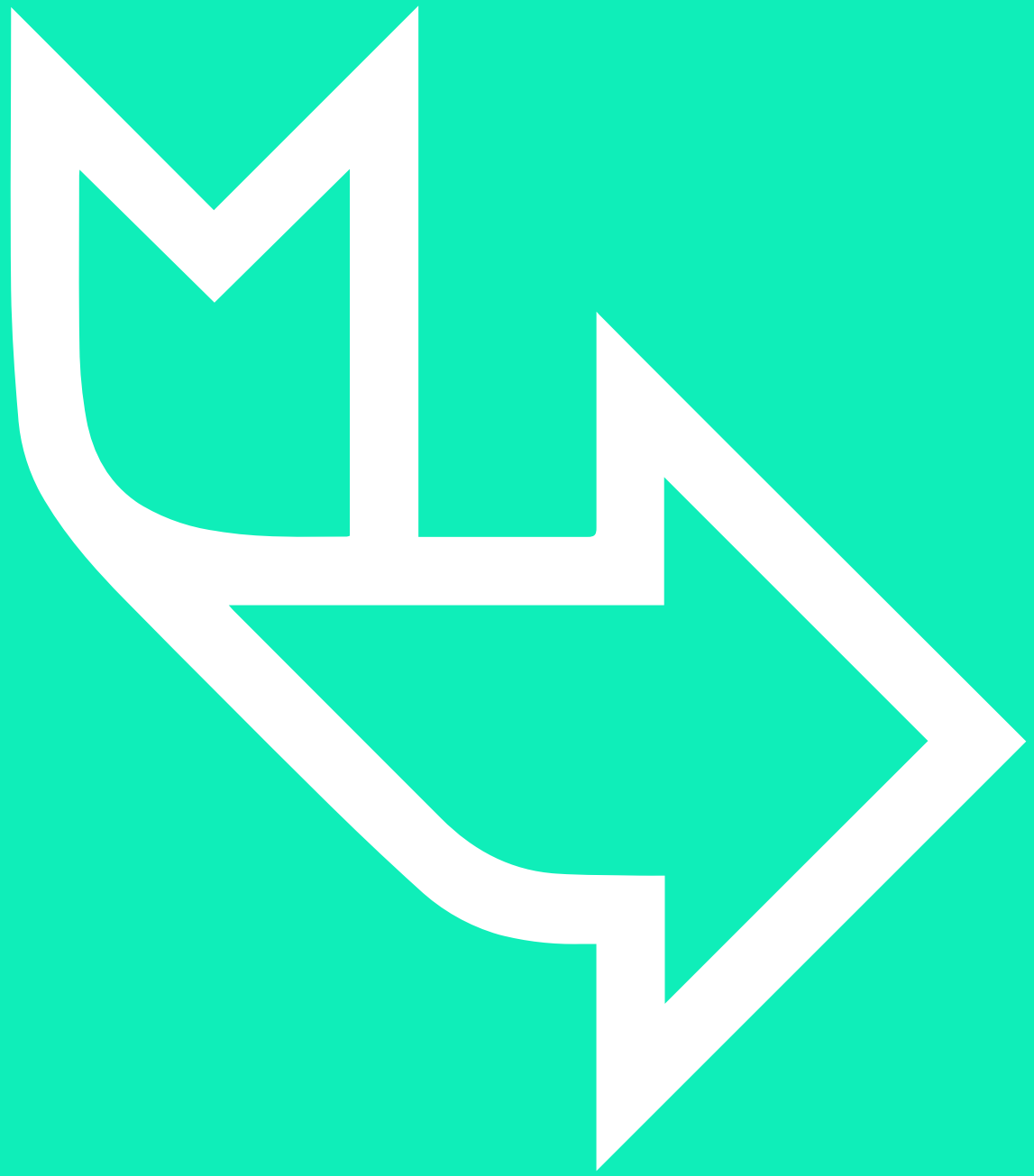
LEARNING CHECK

Think about your answers to these questions:

- Which legislation and regulatory standards apply to your organisation when doing Data Analytics?

- What ethical risks might your organisation need to avoid or mitigate?

- Who in your organisation could be involved in monitoring the use of analytical and machine learning models?



HOW DID YOU GET ON?

Learning objectives

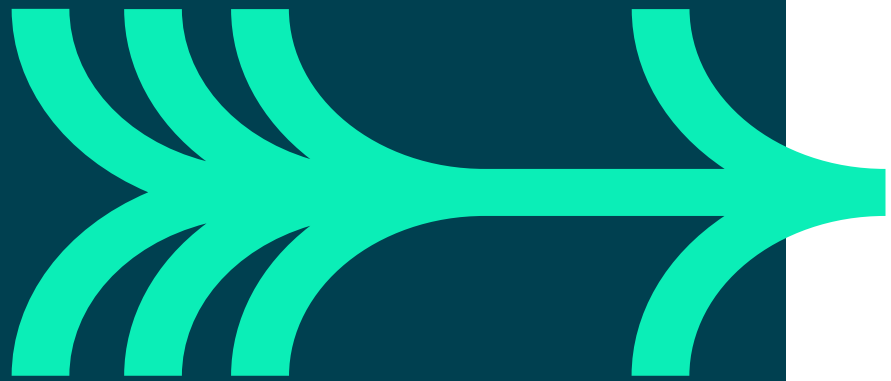
•Be aware of the legislation and standards Data Scientists must adhere to.

•Discuss the importance of legal, ethical, and moral considerations in Data Analytics projects and identify applicable UK legislation for which employees should receive training.

•Discuss ethical considerations for data handling.

•Recognise ethical considerations in examples of machine learning, deep learning, and AI.

DEPLOYING MODELS AND INSIGHTS



Learning objectives

- Understand how analytical models can be deployed.
- Evaluate how best to deploy a given model.
- Define checks which can be used to prevent model failures.
- Use Python and associated libraries to deploy a machine learning model.
- Describe which metrics can be used to monitor deployed machine learning models.

Expected prior knowledge

- Nothing is assumed about your background.

Approaches to deployment

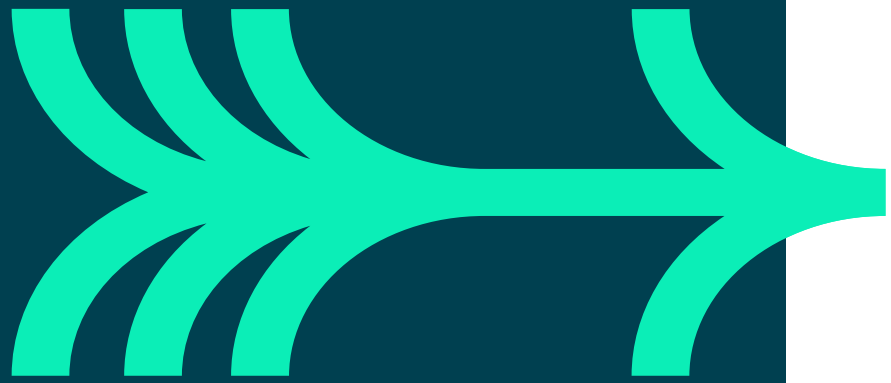
SERVER-SIDE

In server-side deployment, we create a web server to handle requests from clients.

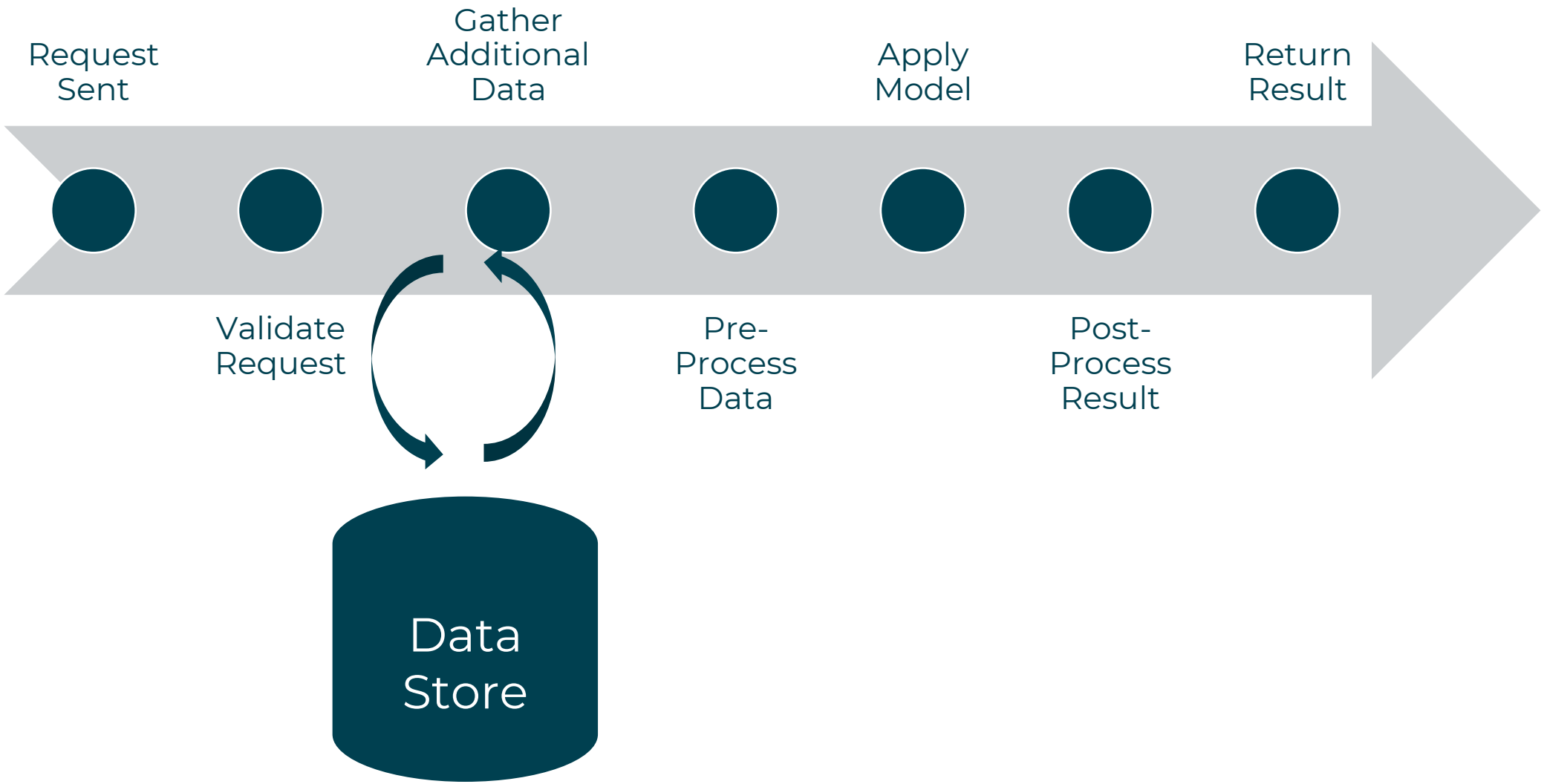
Request data is passed to an inference pipeline, processed, and results sent back to the client.

There are two main ways we can do this:

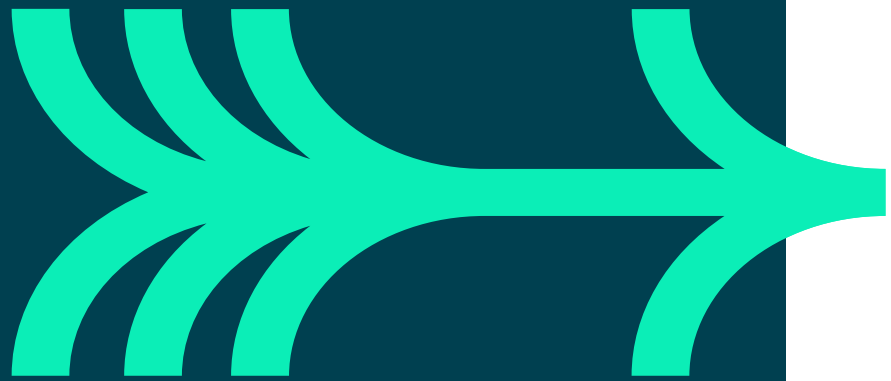
- Stream processing (typically as an API)
- Batch processing



Streaming or API



STREAMING OR API CONSIDERATIONS



Scaling applications

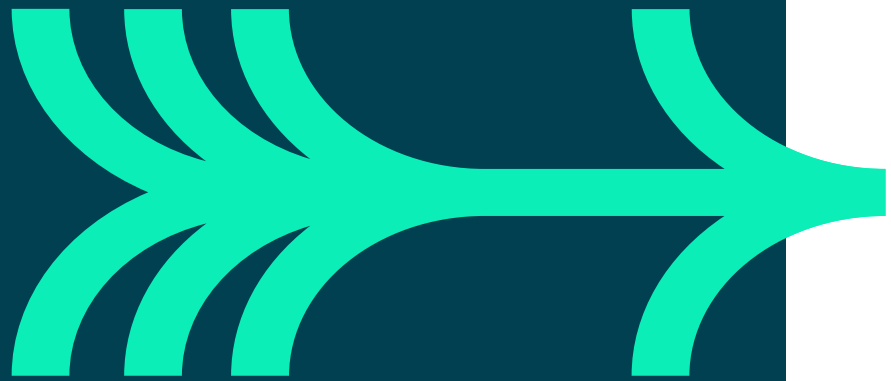
Fluctuation in demand

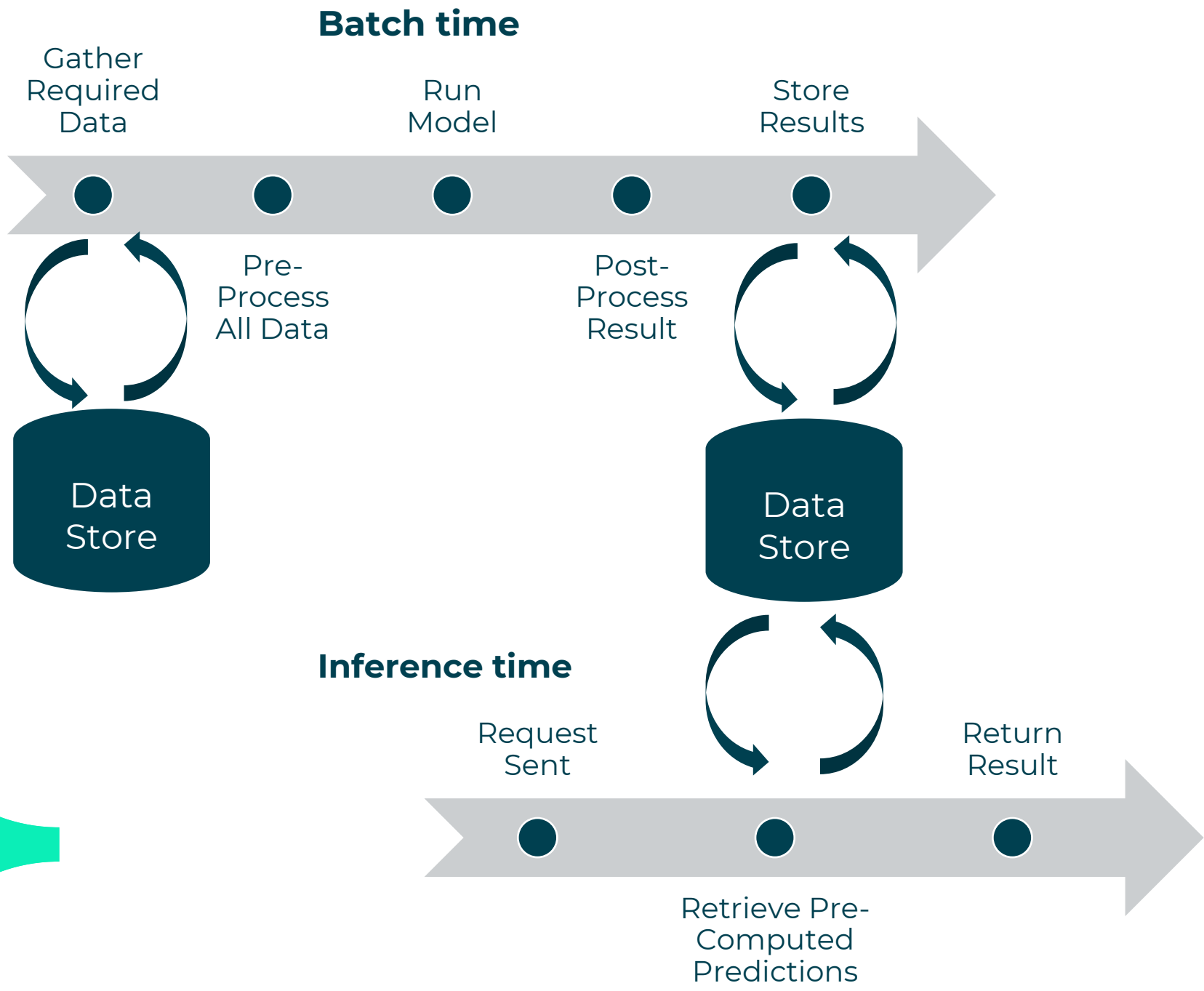
Latency

Streaming or API – example deployment

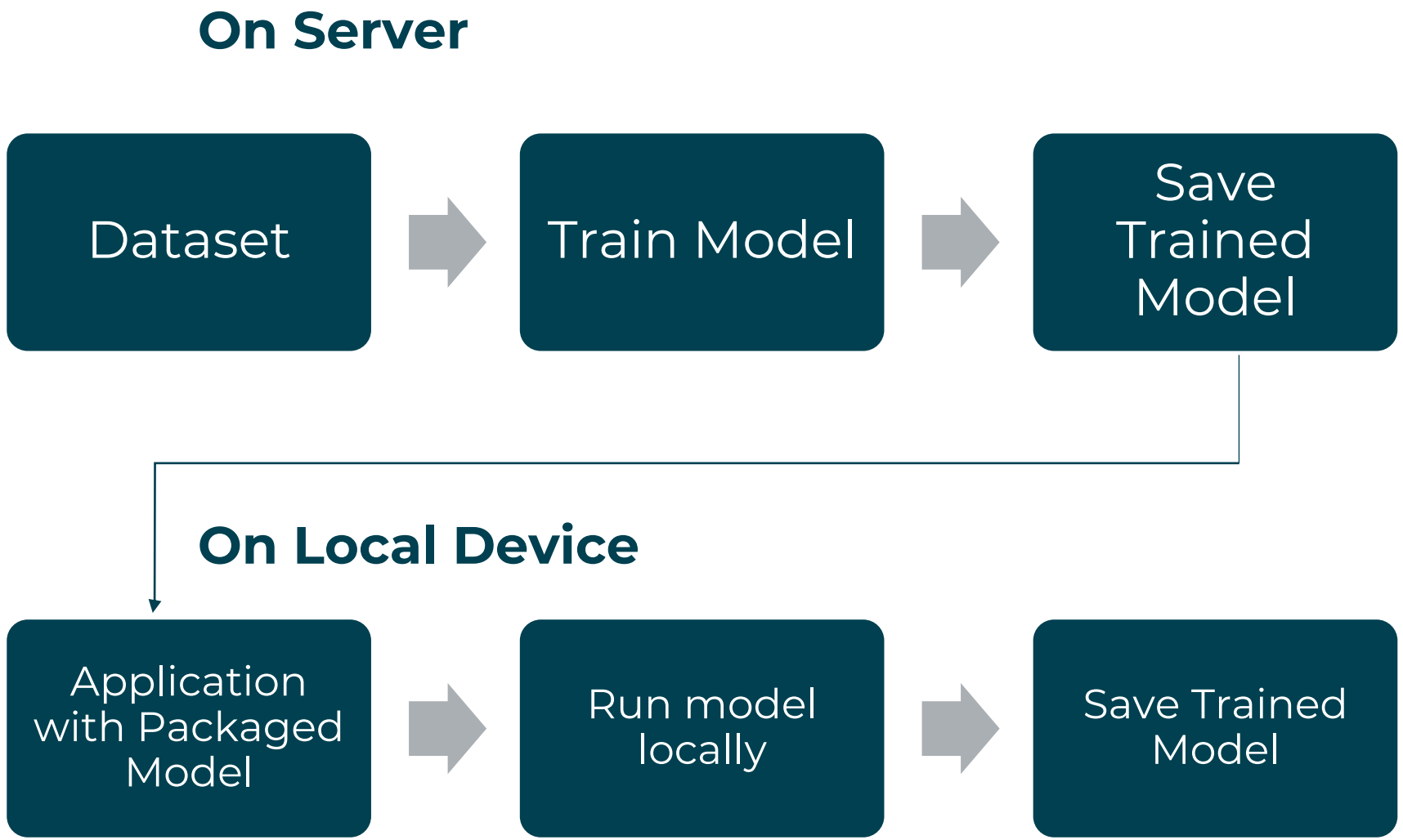
```python
from typing import Union
from utilities import make_prediction
from fastapi import FastAPI

app = FastAPI()

@app.get("/")
def read_root():
    return {"Hello": "World"}


@app.get("/items/{item_id}")
def read_item(item_id: int, q: Union[str, None] = None):
    return {"item_id": item_id, "q": q}

@app.get("/predict/{item_id}")
def get_prediction(item_id: int, ds: Union[str], fr: Union[str], vc: Union[str], dur: Union[int, float]):
    data = {"destination": [ds], "fare": [fr], "vehicle_class": [vc], "duration": [dur]}
    return {"item_id": item_id, "model_in":data, "model_out": round(make_prediction(data)[0], 2)}

```

BATCH




Batch time
Gather Required Data
Run Model
Store Results
Pre-Process All Data
Post-Process Result
Data Store
Data Store
Inference time
Request Sent
Return Result
Retrieve Pre-Computed Predictions

Client Side


On Server
Dataset
Train Model
Save Trained Model
On Local Device
Application with Packaged Model
Run model locally
Save Trained Model

ON DEVICE

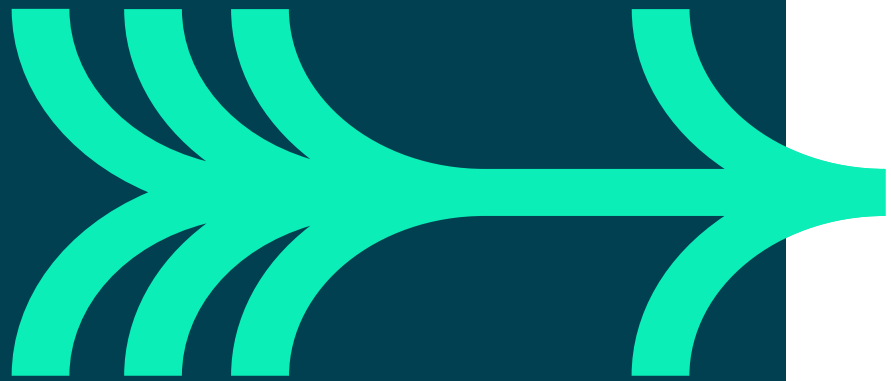
Modern devices can run models locally, known as **native deployment:**

- These should be as small as possible.
- Better suited to less complex models, though changing.
- Some models can tolerate degradation in performance.
- Trade-off is between time saved sending data over network.
- Can aid information security.

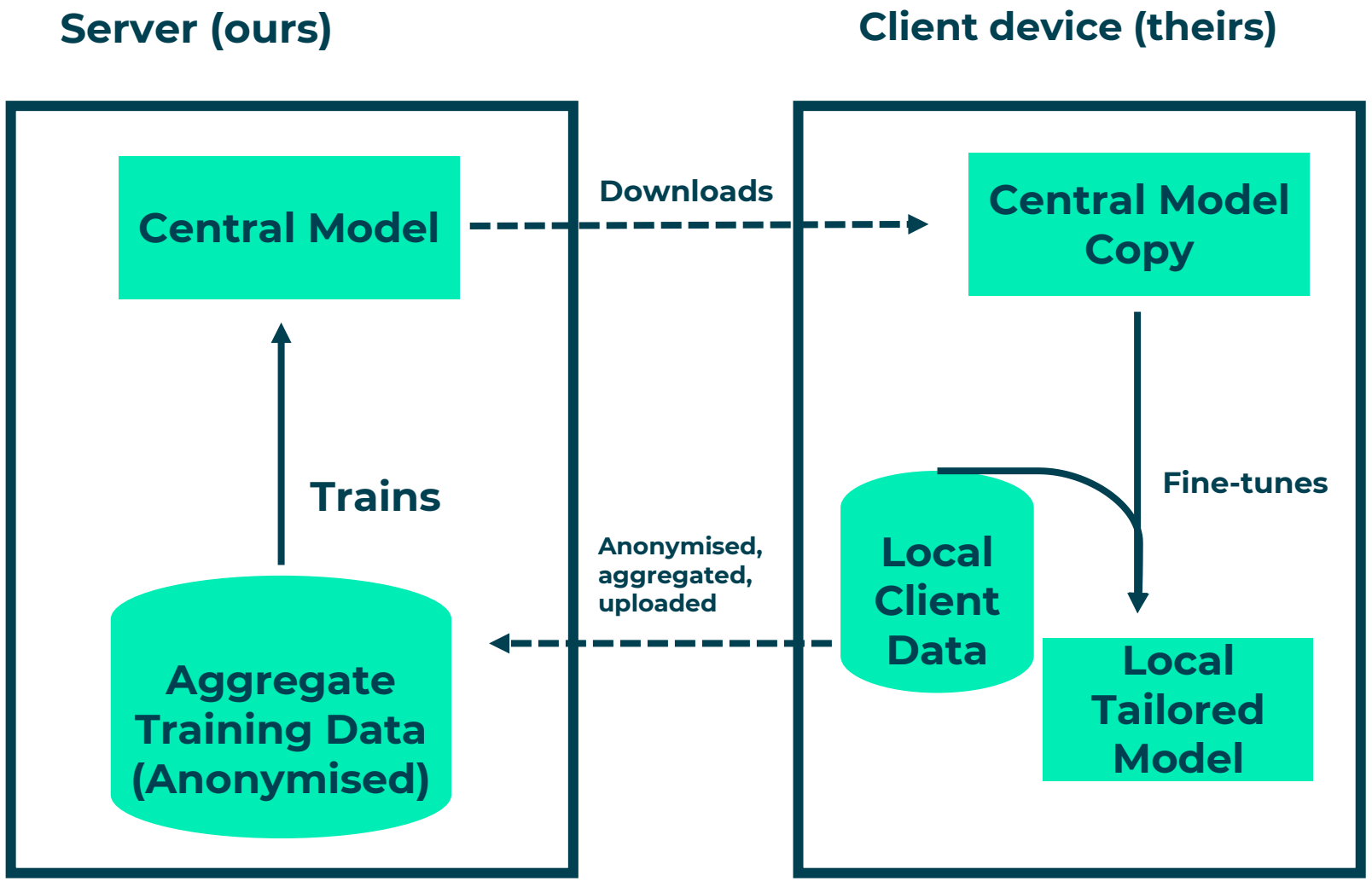
BROWSER SIDE

Instead of **natively** running the model, a device can also run models in the browser:

- Browsers are optimised for graphical calculations.
- Libraries like Tensorflow.js can run complex models in browser.
- Key drawback is having to download model from server.

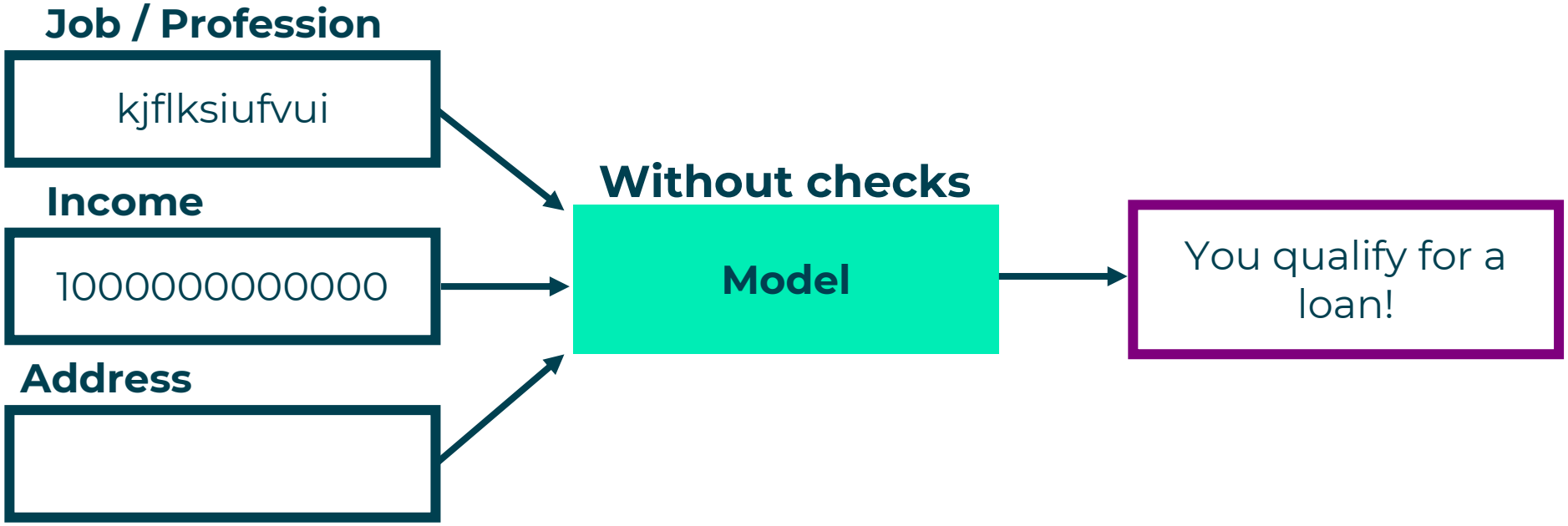


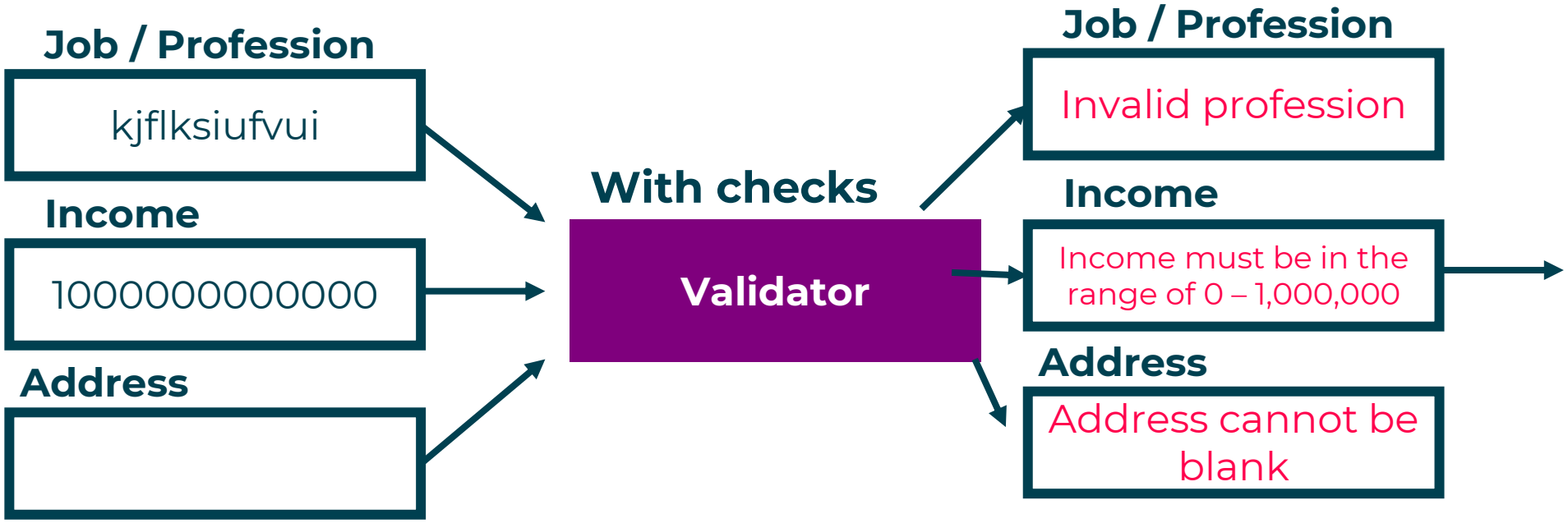
Best of both worlds: Federated Learning



Model Safeguards

Input checks





Output checks

Without checks

Model → Estimated age 150 years

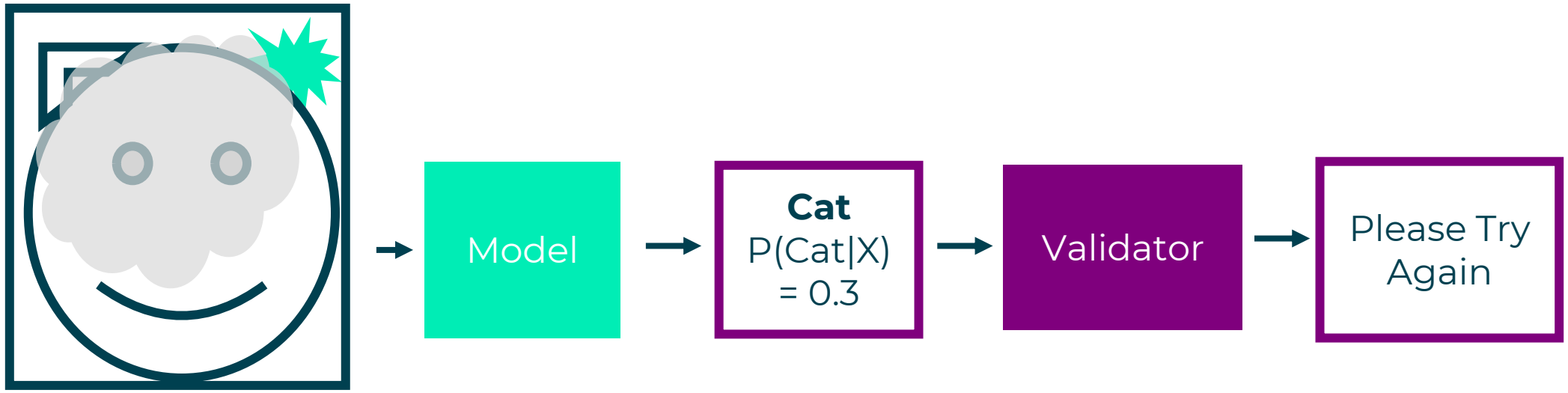
With checks

Model → Validator → Please Try Again

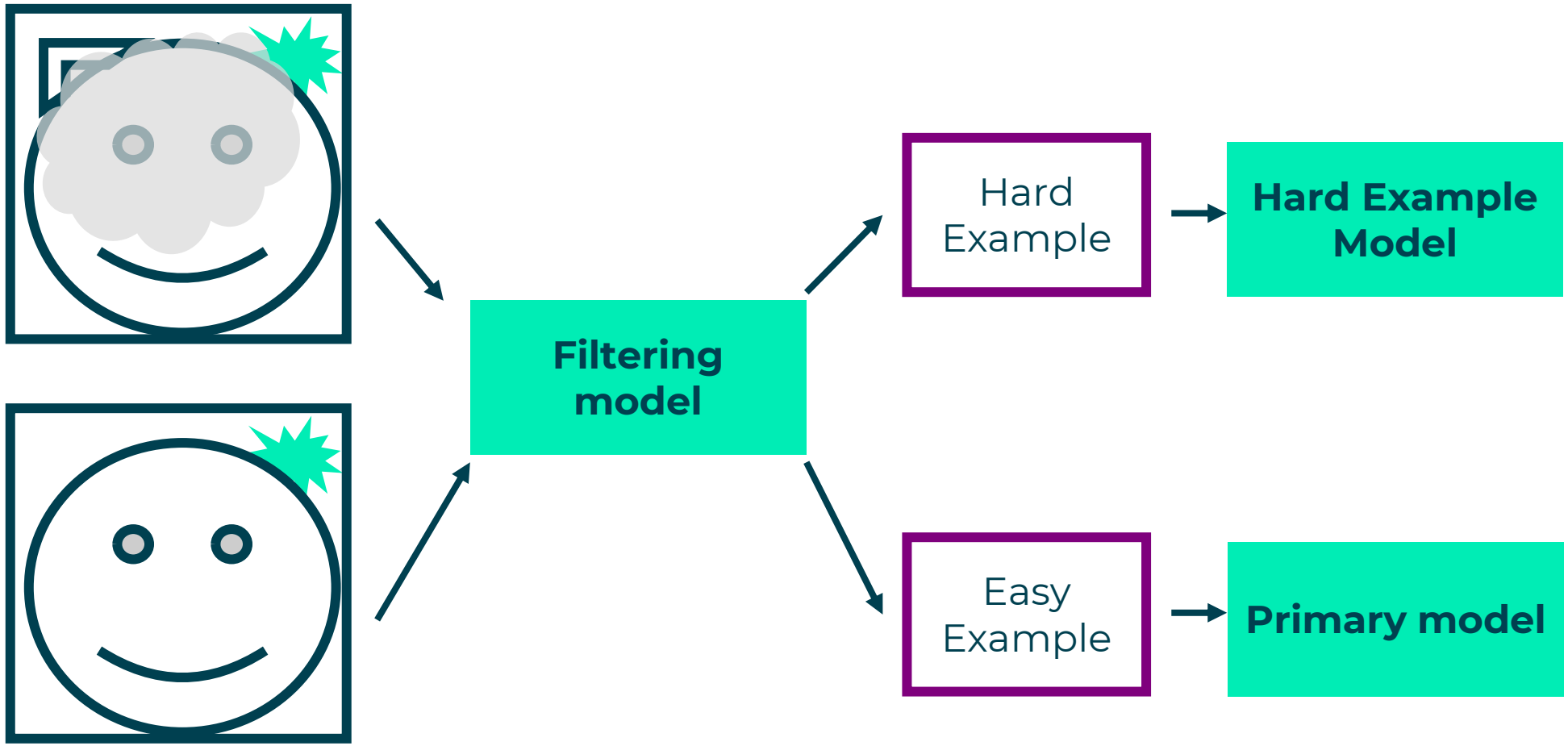
Model failure fallbacks

Even with checks, our model can still be wrong:

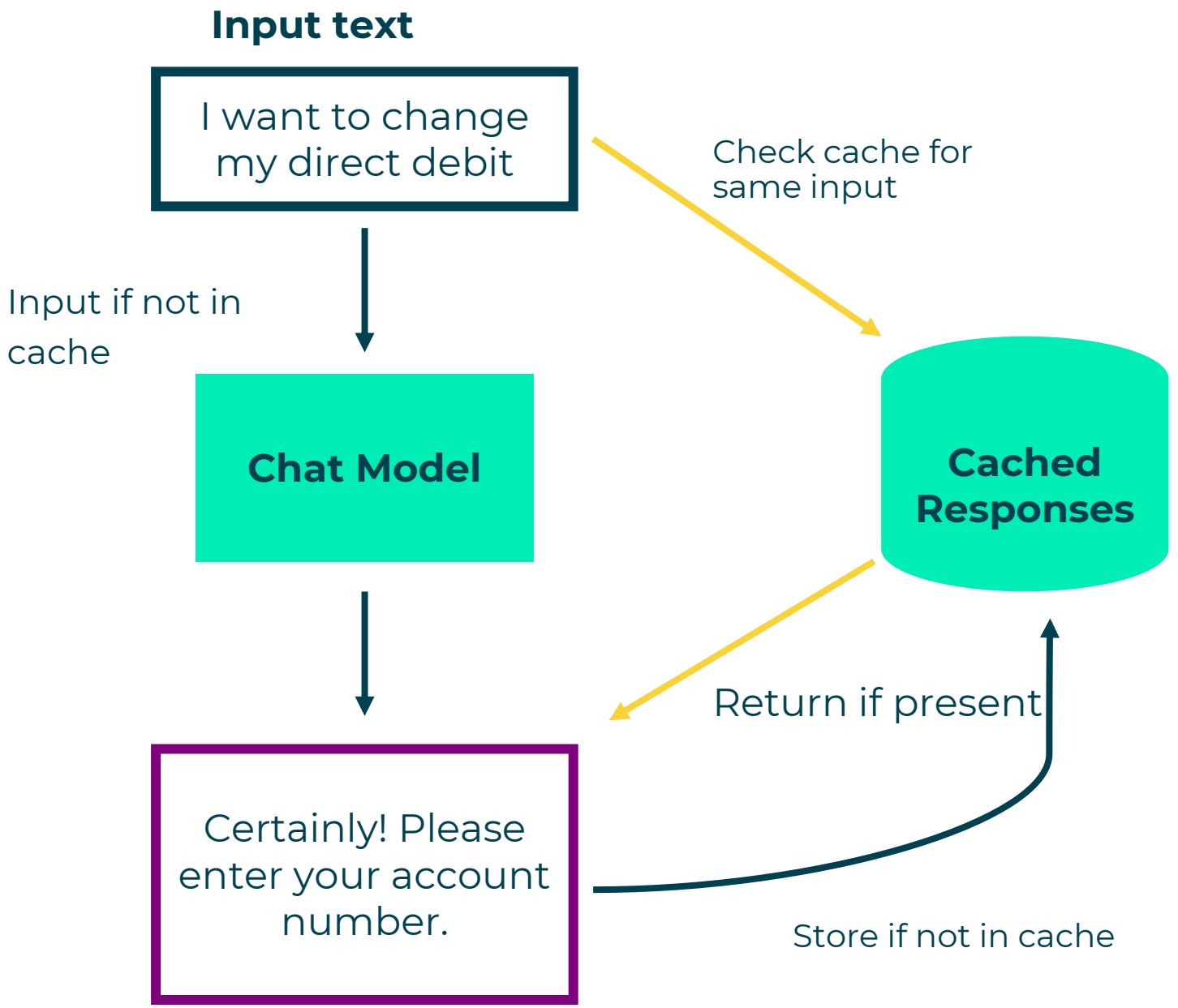
- Well calibrated models make it easier to see when.



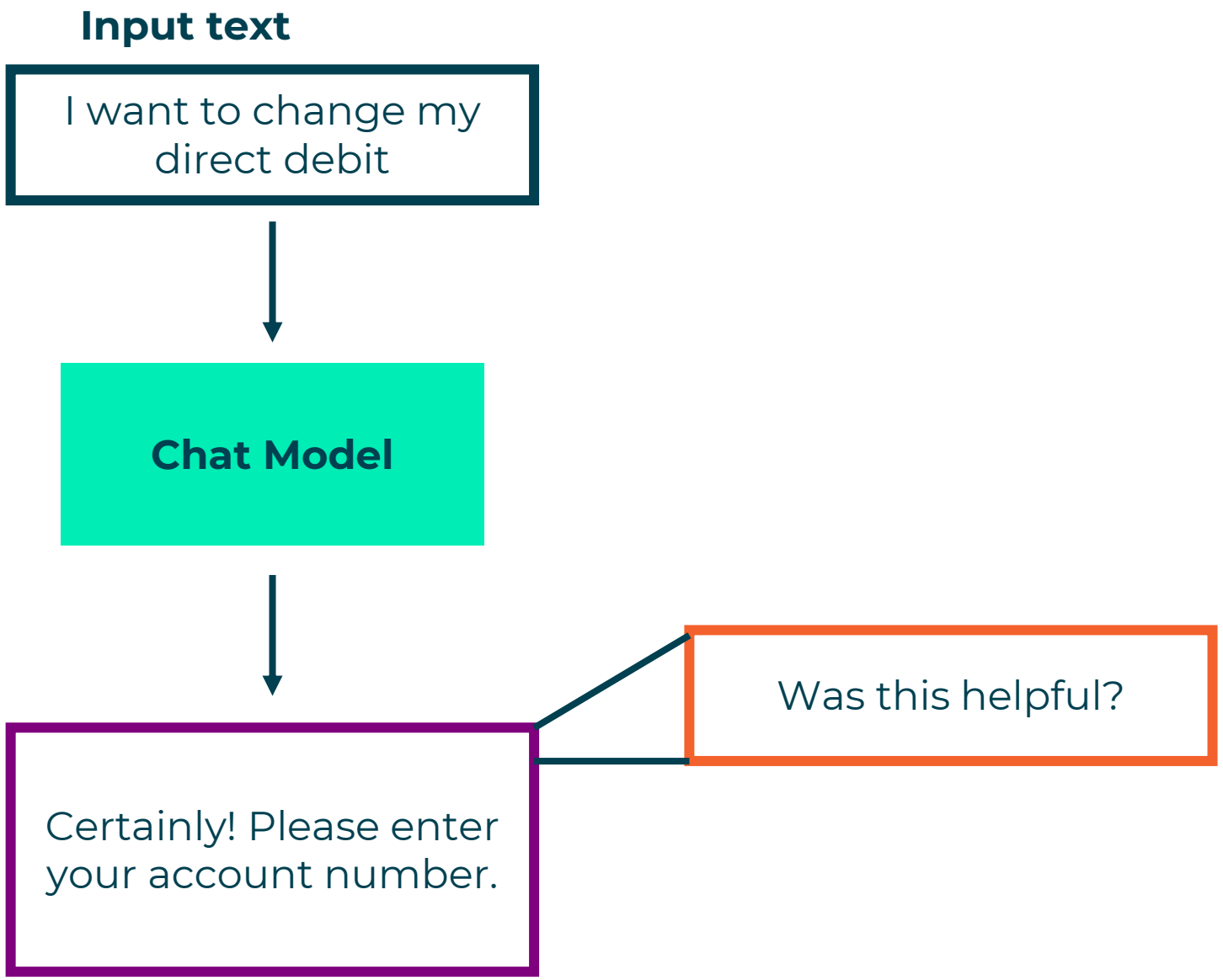
Filtering models



Scaling models

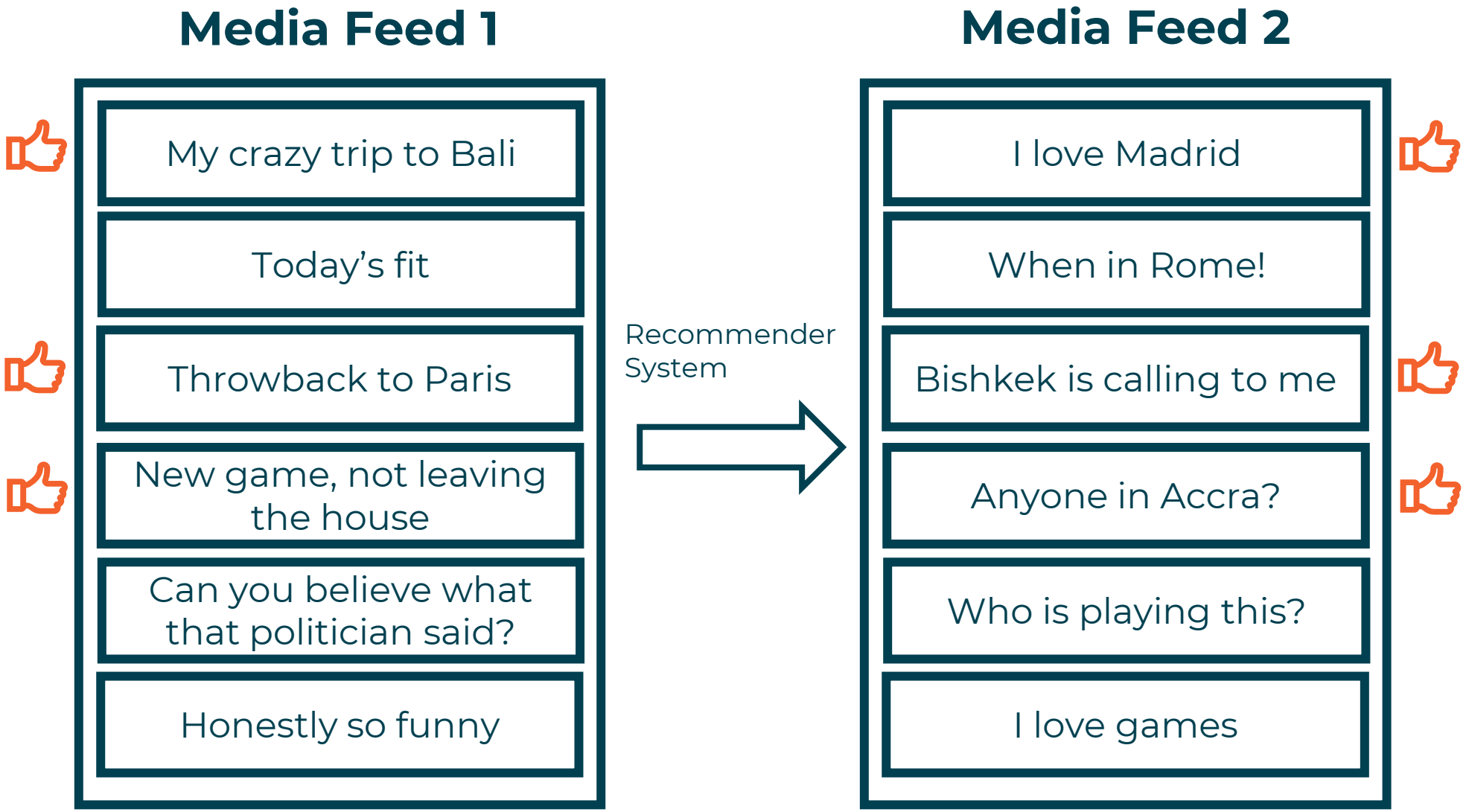


Getting feedback



Monitoring models after deployment

Feedback Loops



TYPES OF DRIFT

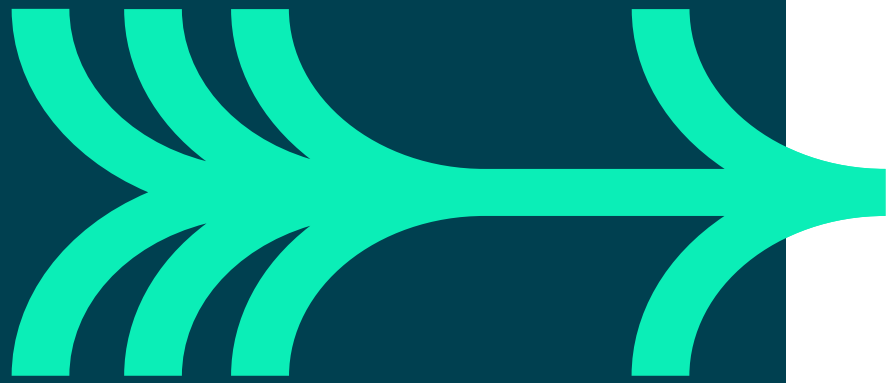
Drift often leads to degradation in model performance.

The key causes of drift are:

- Covariate Shift.
- Label Shift.
- Concept Drift.

These are all detected by monitoring model performance metrics, or the input data distribution.

DEALING WITH DRIFT



Main approach to dealing with drift is retraining the model.

A model can be retrained from scratch

- Include more recent data.
- Typically expensive to create new model.

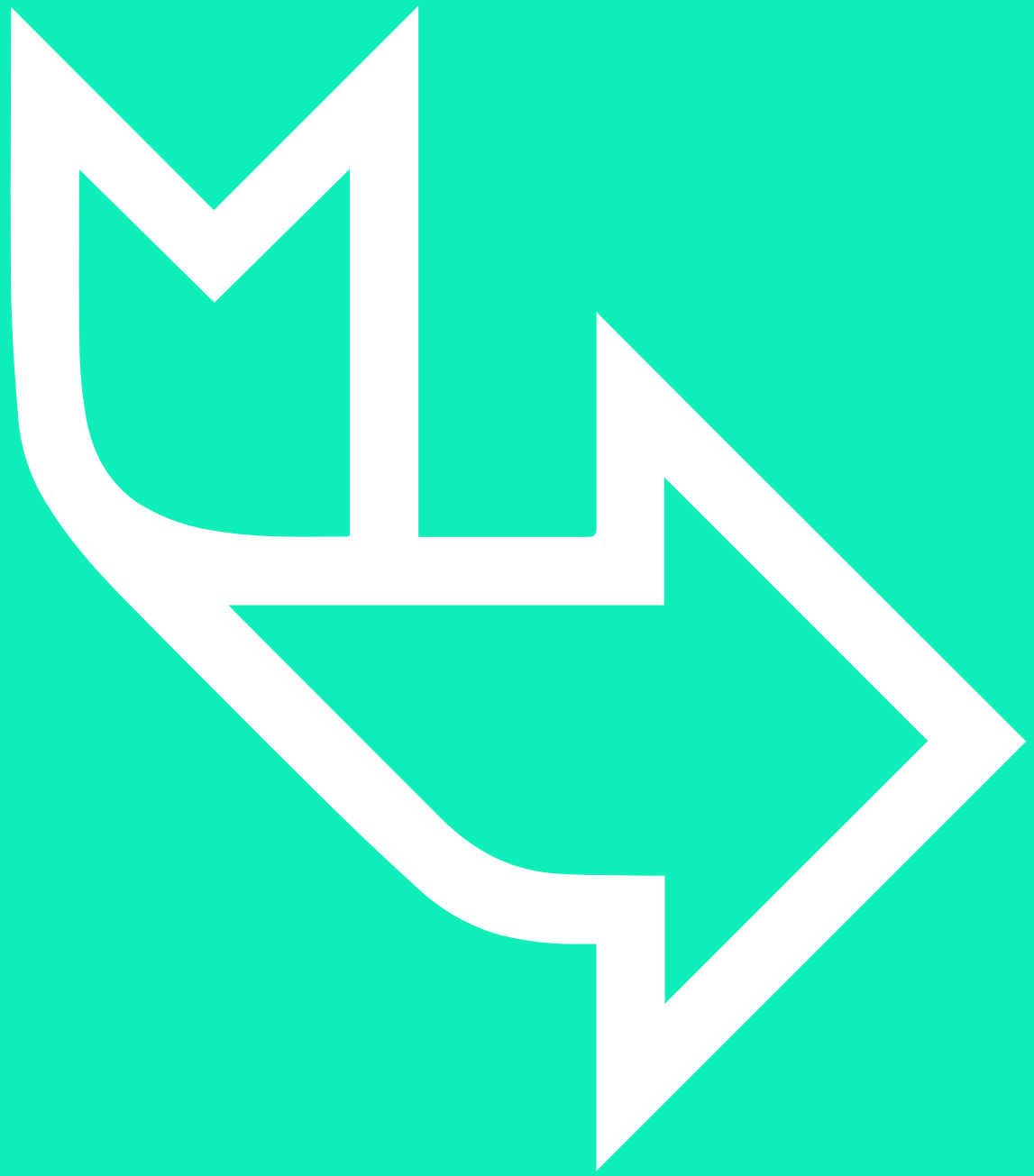
A model could also be updated (statefully retrained)

- Run further training process using new data.
- Less expensive.
- A form of fine-tuning.

Other considerations

AGGREGATES CAN BE MISLEADING

Model	Accuracy Group A	Accuracy Group B	Overall Accuracy
Original	92%	97%	95%
Retrained after drift	85%	98%	96%



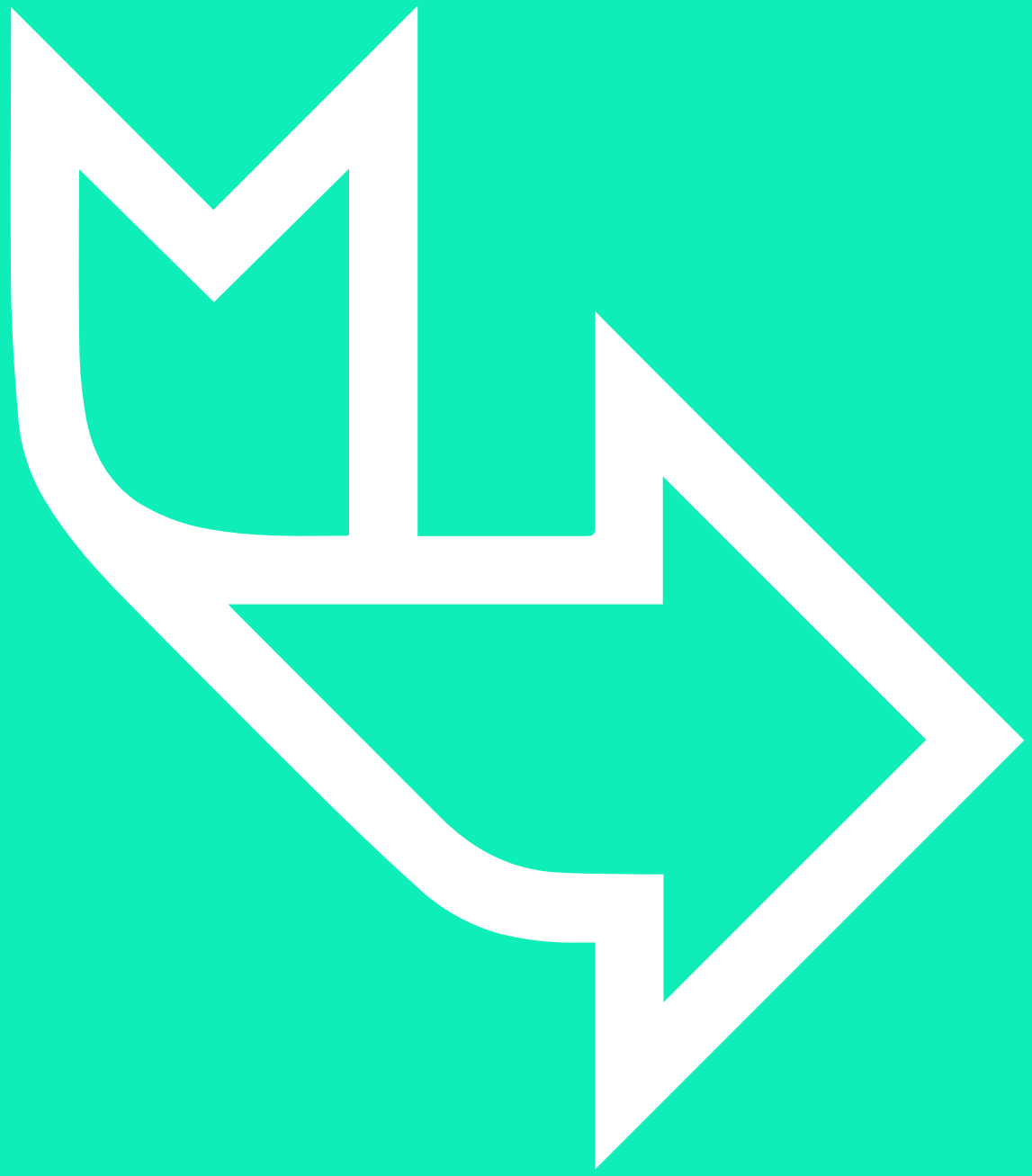
EXERCISE

Work though Module 10 Exercises: Deploying Models and Insights

LEARNING CHECK

Think about your answers to these questions:

- How can analytical models be deployed?
- How do we choose the best way to deploy a given model?
- What can be done to prevent model failures?
- Which metrics can be used to monitor deployed machine learning models?

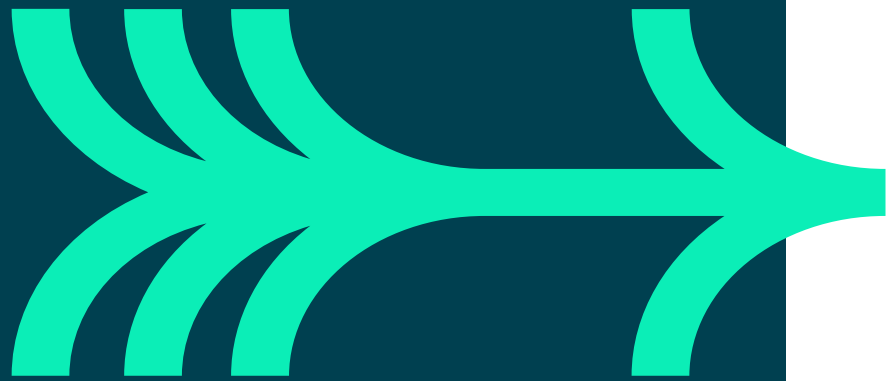


HOW DID YOU GET ON?

Learning objectives

- Understand how analytical models can be deployed.
- Evaluate how best to deploy a given model.
- Define checks which can be used to prevent model failures.
- Use Python and associated libraries to deploy a machine learning model.
- Describe which metrics can be used to monitor deployed machine learning models.

WHERE TO GO NEXT



Learning objectives

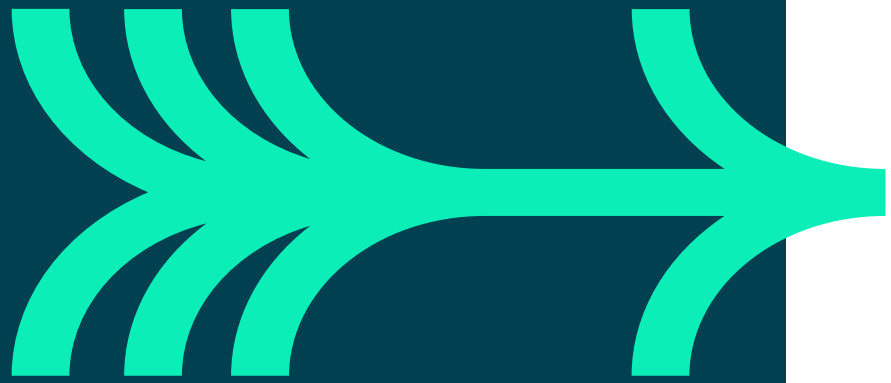
- Understand the role of deep learning in modern Artificial Intelligence.
- Know which qualifications and professional memberships can benefit data scientists.

Expected prior knowledge

- Nothing is assumed about your background.

Study advanced topics

WHAT IS DEEP LEARNING



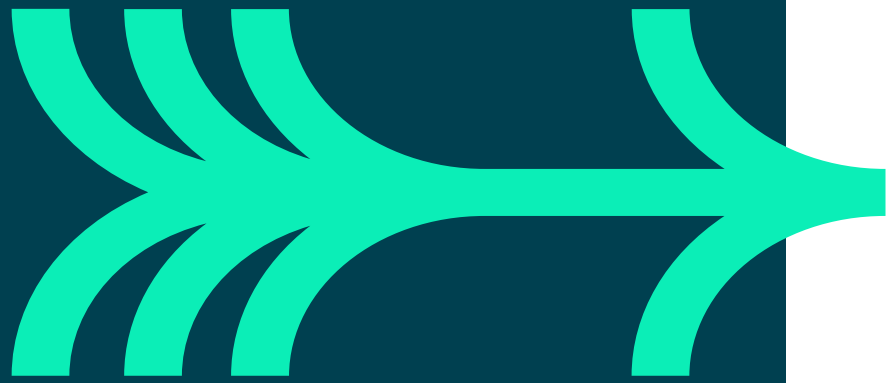
Deep learning uses the neural network algorithm to fit models to data

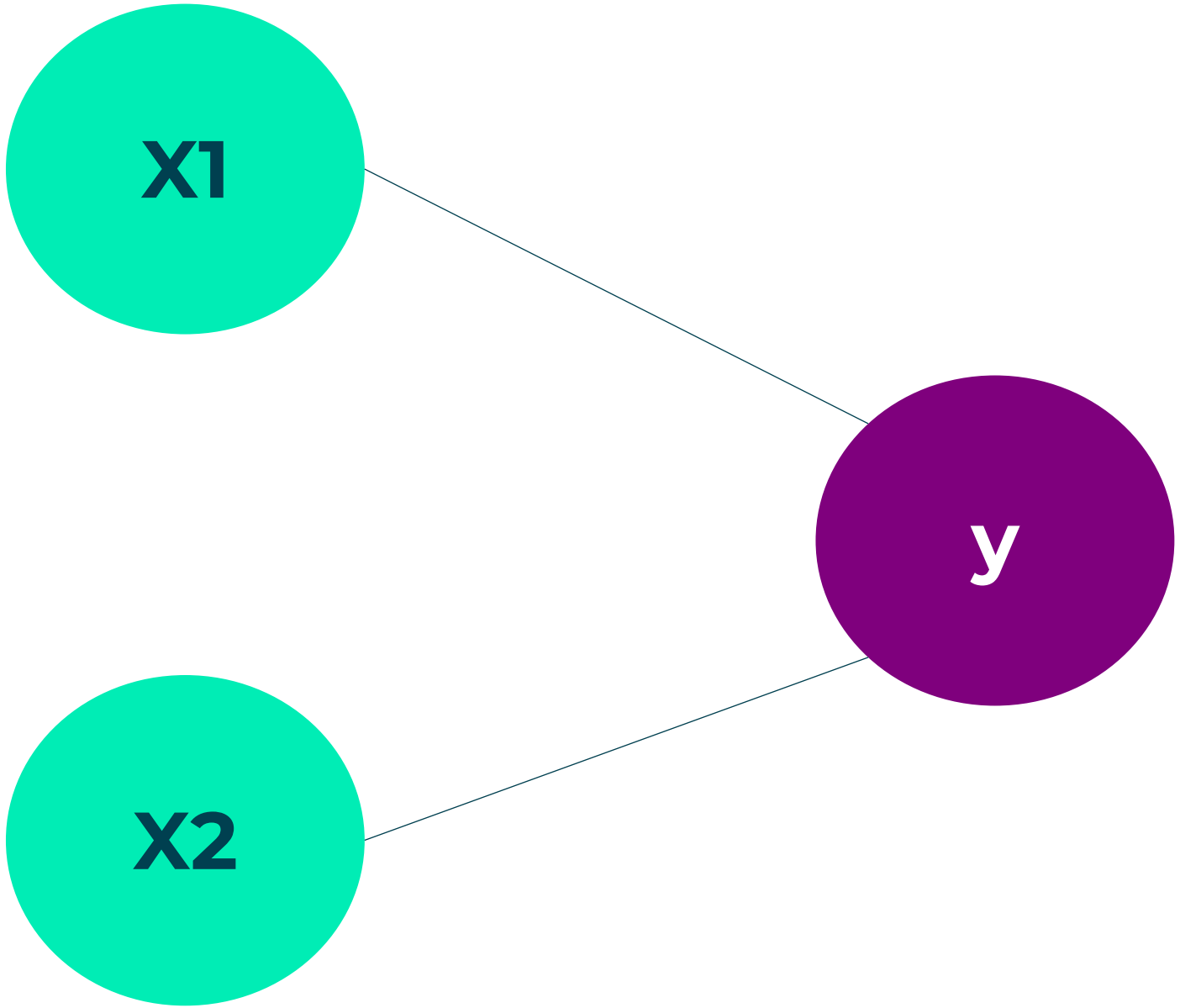
- Highly performant.
- Generic.
- Prone to overfitting.

Neural networks make use of the multilayer perceptron algorithm

- Weighted sums of inputs are passed into a function.
- This either produces the output, or is passed into another neuron.
- The more neurons, the more complex the model can be.

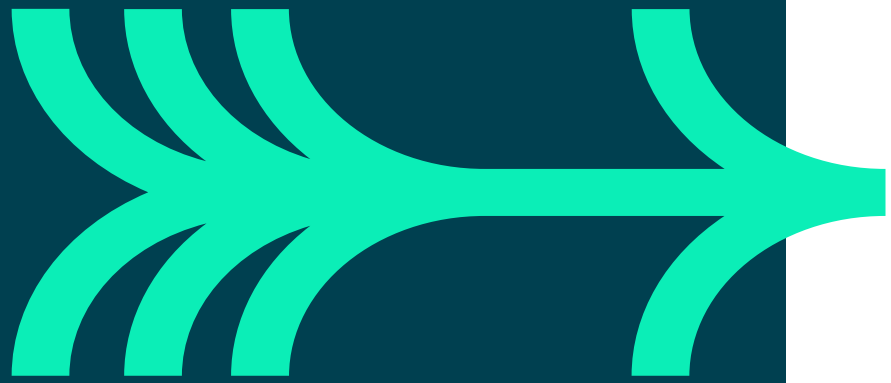
NEURAL NETWORK DIAGRAM





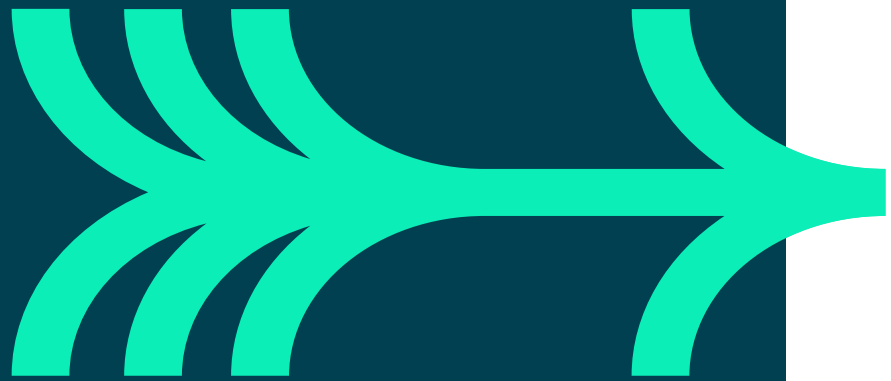
STATE-OF-THE-ART MODELS

- ChatGPT
- BERT
- BART



Attain professional qualifications

WHAT ARE PROFESSIONAL FRAMEWORKS?

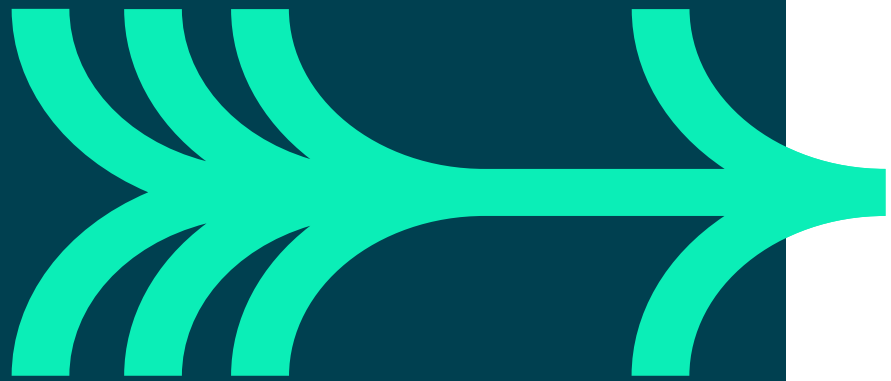


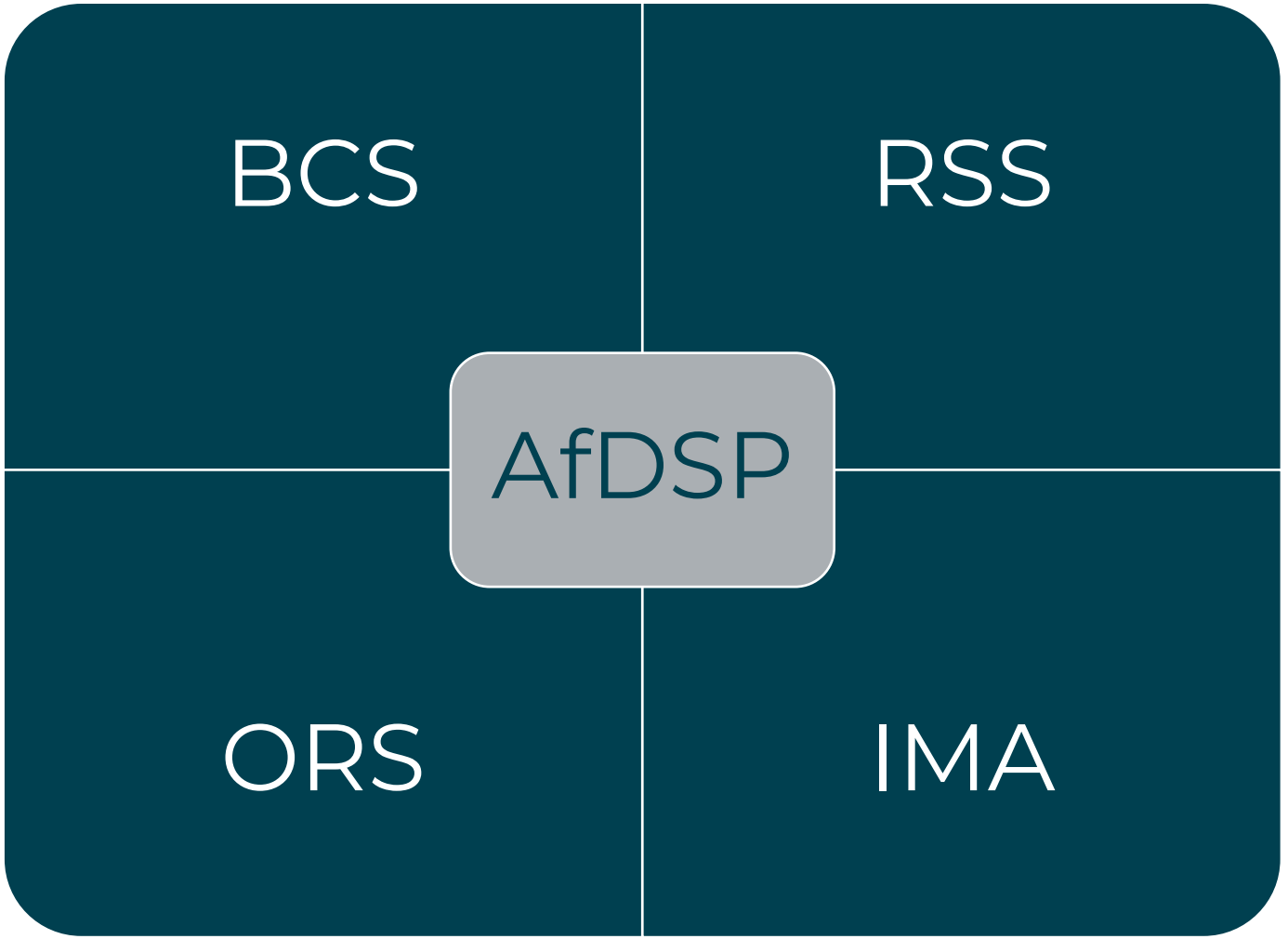
Essentially checklists of standards listing what is required and expected of someone in a professional role, e.g., teaching, accountancy, engineering.

Professional status is used across industry to make it easier to see what training someone needs and to help employers advertise and recruit for the right role.

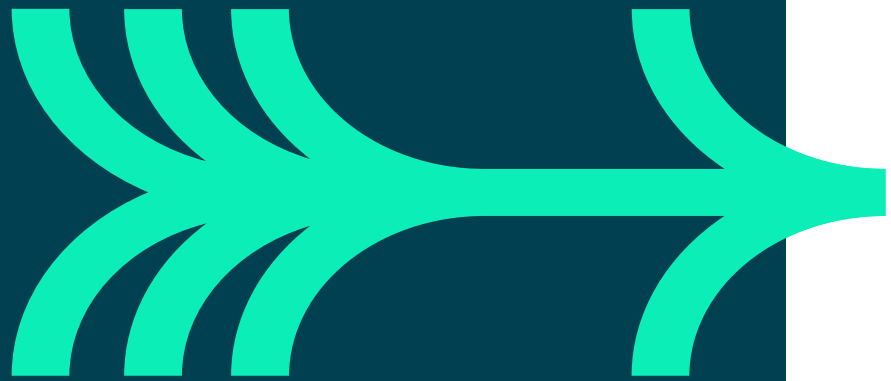
Over time professional frameworks are updated – so in any profession, it is expected we carry out continuous professional development to ensure we stay up to date. As standards are updated over time, the professional bodies overseeing the framework can tackle the latest issues.

DATA SCIENCE PROFESSIONAL BODIES





PROFESSIONAL CERTIFICATIONS



Once you are a member of one of the four key professional bodies you may apply to become recognised as either:

- Data Science Professional.
- Advanced Data Science Professional.

Which level a Data Scientist applies for depends on:

- your level of responsibility (e.g., departmental consultant or strategic consultant for Advanced level).
- how much authority you have to make organisational decisions and the reach of your impact.
- how complex your technical skills are or the organisational problems you tackle.

Your trainer will not be able to definitively answer if your evidence is sufficient for Professional status – this is externally assessed by an AfDSP member body.

MINIMUM REQUIREMENTS TO BEGIN AN APPLICATION

- A UK Level 6 qualification in an appropriate subject
- Some formal training within data science
- Typically, 5 years' relevant work experience
- 2 years' CPD evidence
- Evidence that you meet the competencies and level of responsibilities

These professional certifications are not intended to be achieved by those just starting out, but give you a structured plan to work towards.

Skill areas

A. Data Privacy and Stewardship

- Ensuring the protection of personal and sensitive data
- Managing sensitive data
- Data stewardship and standards (FAIR principles)

B. Definition, acquisition, engineering, architecture, storage and curation

- Data collection and management
- Data engineering
- Deployment

C. Problem definition and communication with stakeholders

- Problem definition
- Relationship management

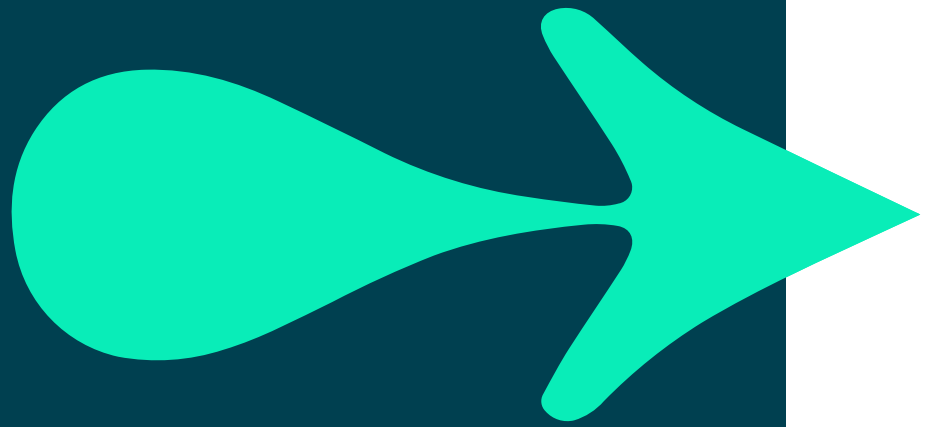
D. Problem solving, analysis, statistical modelling, visualisation

- Identifying and applying technical solutions and project management approaches
- Data preparation and feature modelling
- Data Analysis and model building

E. Evaluation and Reflection (evidenced throughout)

- Project evaluation
- Ethical behaviour
- Sustainability and best Practices
- Reflective practice and ongoing development

ACTIVITY



As we consider each skill area create a skills map for yourself:

- ideas for evidence towards each skill area.
- identifying training needs you have in each skill area.

If you will be working with Data Scientists but aren't intending this for yourself:

- what opportunities could you create or support with to allow your organisation to increase their number of recognised Data Scientists?

Your trainer will not be able to definitively answer if your evidence is sufficient for Professional status – this is externally assessed by an AfDSP member body.

Attend another course!

FURTHER COURSES (ASK FOR SPECIFIC INTERESTS)

- R for Data Handling
- Statistics for Data Analysis
- R for Data Science and Machine Learning
- Practical Big Data Analysis
- Machine Learning, AI and Deep Learning with Python
- Maths and Statistics for Data Science, Big Data, and Operational Analysis
- Machine Learning, AI and Deep Learning with R

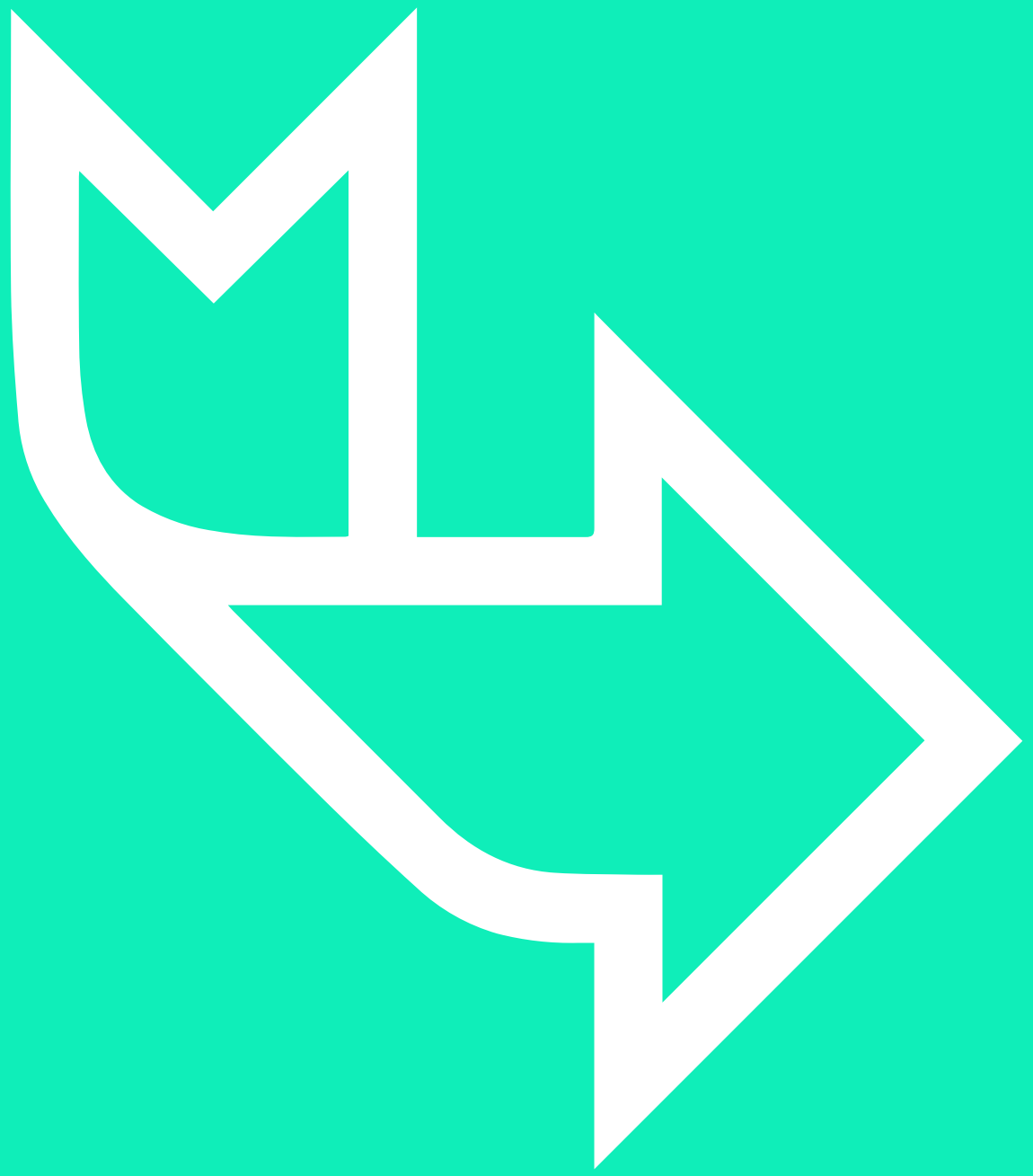
Further reading

LEARNING CHECK

Think about your answers to these questions:

- What is the role of deep learning in modern Artificial Intelligence?

- Which qualifications and professional memberships can benefit data scientists?



HOW DID YOU GET ON?

Learning objectives

- Understand the role of deep learning in modern Artificial Intelligence.
- Know which qualifications and professional memberships can benefit data scientists.