SCC361: Artificial Intelligence

Week 1: Introduction to Artificial Intelligence and Machine Learning

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

SCC361: Artificial Intelligence

Introduction to the Module

Lectures, Materials and Expectations

Overview of Artificial Intelligence

Overview of Machine Learning

Expected Learning Outcomes

We will to be able to:

- understand AI concepts, applications and trends
- understand machine learning terms
- train machine learning models for specific tasks
- learn to implement simple AI-based systems
- learn how to evaluate the performance of AI systems

Teaching Staff

Lecturers

Dr Bryan M. Williams

Weeks 1-5

Office Hours:

Tuesday 16:00
Wednesday 9:00
Wednesday 13:00

Book via Teams message

Dr Hossein Rahmani

Weeks 6-10

Teaching Staff

TAs for Lab Sessions

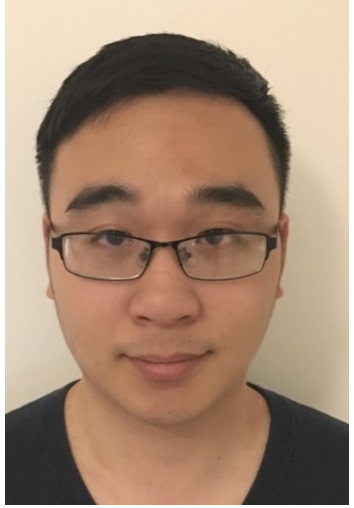
Name	Group 1	Group 2	Group 3	Group 4	Group 5
Ahlam Althobaiti	■			■	
Ovini Gunasekera	■				■
Haoran Peng	■	■			
Somayeh Bazin		■	■		
Xavier Hickman			■		■
Zhaonian Zhang				■	
TBC		■	■	■	■

Teaching Staff

Guest Lecturers

"Assisting the Criminal Justice System" – Week 8

Professor Dame Sue Black, Baroness of Strome, DBE OBE PhD DSc FRSE, FBA FRAI FRSB FRCP
President, St John's College, Oxford



"Vein Pattern Mapping and Matching" – Week 9

Dr Zheheng Jiang, BSc PhD AFHEA
Senior Research Associate, Lancaster University

Lecture Plan

Weeks 1-5

1. Introduction to Artificial Intelligence and Machine Learning
2. Features in Machine Learning and Feature Extraction
3. Computer Vision and Natural Language Processing
4. Clustering and Classification
5. Artificial Neural Networks

Weeks 6-10:

6. Genetic Algorithms
7. Naïve Bayesian Classifier
8. Decision Tree Classifier
9. Introduction to Deep Neural Networks
10. Introduction to Convolutional Neural Networks

Teaching Structure

Lectures

- Monday
  12:00 – 12:50
  Faraday Lecture Theatre

- Monday
  18:00 – 18:50
  George Fox LT1

Labs

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076

Assessment

2 Courseworks: 40%

- CW1 (20marks):
  - Submission: On Moodle
  - Deadline: **4pm Friday 18th November, 2022**
- CW2 (20marks):
  - Details: to be confirmed

Exam: 60%

- Next semester in 2023
- Date: To be confirmed

Questions?

SCC361: Artificial Intelligence

Introduction to the Module

Lectures, Materials and Expectations

Overview of Artificial Intelligence

Overview of Machine Learning

Learning Support

- All materials will be available on Moodle, including
  - Lecture slides and recordings
  - Lab details, assignments and solutions
- Lab sessions
  - Discuss assignments and questions with lecturer and TAs
  - Try to maximise usage of labs
- Arrange extra support if you are struggling and let us know on time
  - Office hours
  - Send me a message on Teams
- Check other available (online) resources

Accessibility

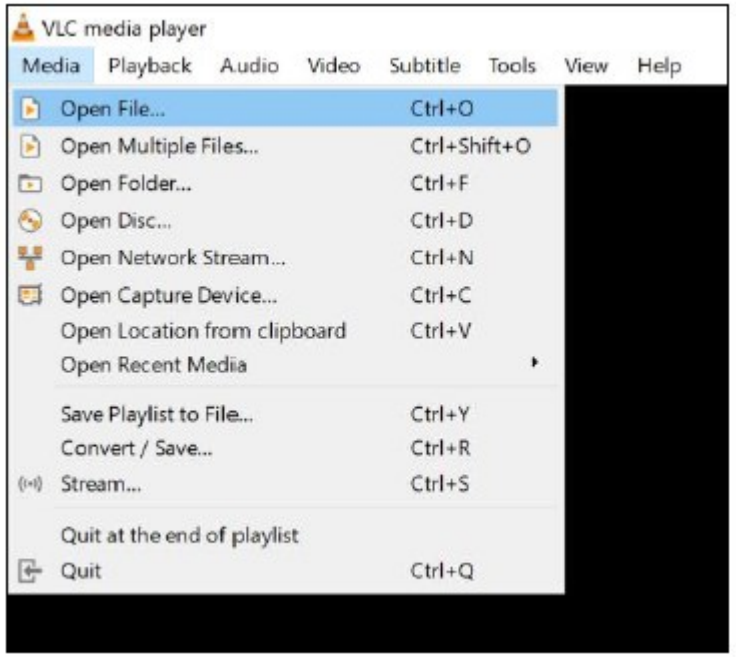
- All our content is expected to meet the UK accessibility requirements
- We have done our best to ensure that that is the case with these course materials
- However, if any course material or part of its content is inaccessible in anyway to any individual or group, kindly let us know.

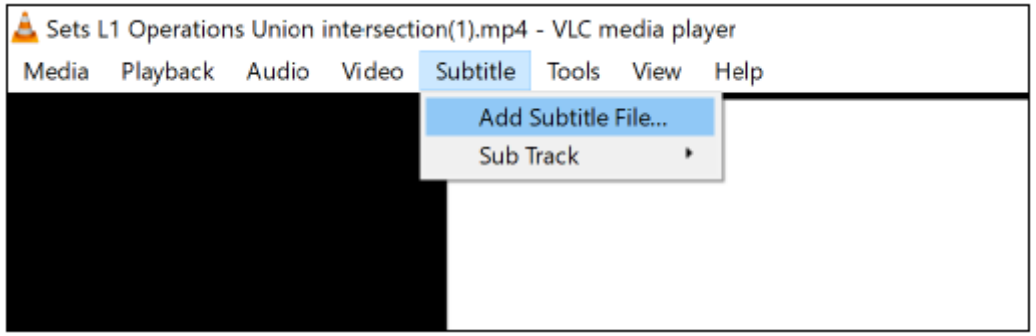
Using Materials Offline

The lectures can be watched on the Moodle space.

If you are struggling to watch the videos on Moodle:

- Download the video and caption file (*.vtt) from Moodle
- Download the free, open source VLC Media player: https://www.videolan.org/vlc/index.en-GB.html
- Open video file in VLC and add caption file





Note:

All learning materials: slides, videos and caption files are @Lancaster University copyright and are **not to be shared or distributed**.

Plagiarism

Passing off someone else's work as your own, including:

- Colluding with a classmate or someone else to do your work
  - Submitting code written by someone else
  - Paying for someone else to do your work
  - Adapting code by someone else with only a minor modification
- Course work is submitted on Moodle and will be checked automatically for plagiarism!
- Academic malpractice and plagiarism applies online and offline
- Direct sharing of code, sharing solutions and/or partial solutions with other students, either privately or in an open chat, is **not acceptable**

What do we expect from you?

Attendance:

- Lectures, labs etc., be punctual

Active learning:

- Read around (explore) the subject
- Use recommended books and available online resources
- Ask questions, try things yourself, keep notes
- Have a study plan, get a study partner

Integrity:

- Honesty, no plagiarism/ result manipulation

Attendance Check-in

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the “You are checked in” confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Questions?

SCC361: Artificial Intelligence

Introduction to the Module

Lectures, Materials and Expectations

Overview of Artificial Intelligence

Overview of Machine Learning

Welcome to SCC361

Part 1 (Weeks 1 – 5):

- Wk1: Intro to Artificial Intelligence & Machine Learning
- Wk2: What are features and how to extract features from texts, images, etc.
- Wk3: Computer Vision and Natural Language Processing
- Wk4: Clustering & Classification
- Wk5: Intro to Artificial Neural Networks & Review of Previous Lectures

Part 2 (Weeks 6 – 10):

- Dr Hossein Rahmani (GAs, NBCs, DTCs, DNNs, CNNs)

SCC361: Week 1

Lancaster University

Artificial Intelligence

AI Overview

Definition of AI

Goals of AI

AI and Society

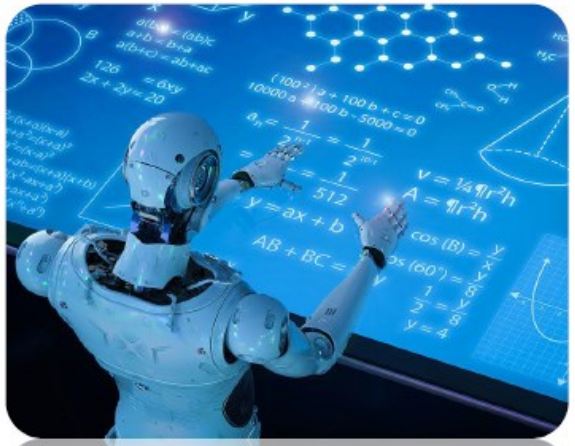
Introduction to Machine Learning

Machine Learning Overview

Types of Machine Learning

Supervised Learning

Unsupervised Learning

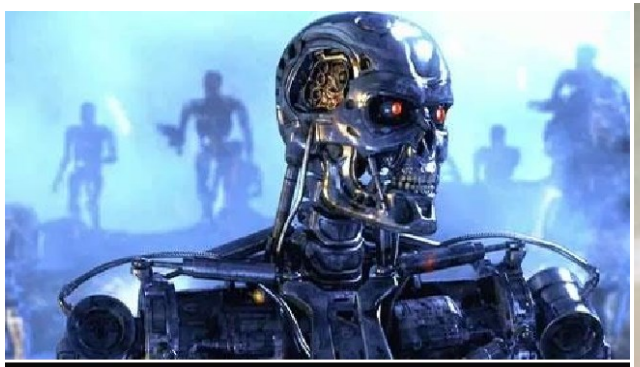


Recommended Reading

- Artificial Intelligence: *Foundations of Computational Agents* 2ed. Poole & Mackworth 2017
- *Artificial Intelligence A Modern Approach*, Russell & Norvig, 2016 (chapters 1,2,5,6)
- *Deep Learning*, Goodfellow et al. , 2016
- *Deep Learning with PyTorch*, Stevens et al, 2020.
- *Machine Learning*, T. M. Mitchell, 1997
- Deisenroth MP, Faisal AA, Ong CS. *Mathematics for machine learning*. Cambridge University Press.
- Artificial Intelligence on Wikipedia
- Many online resources

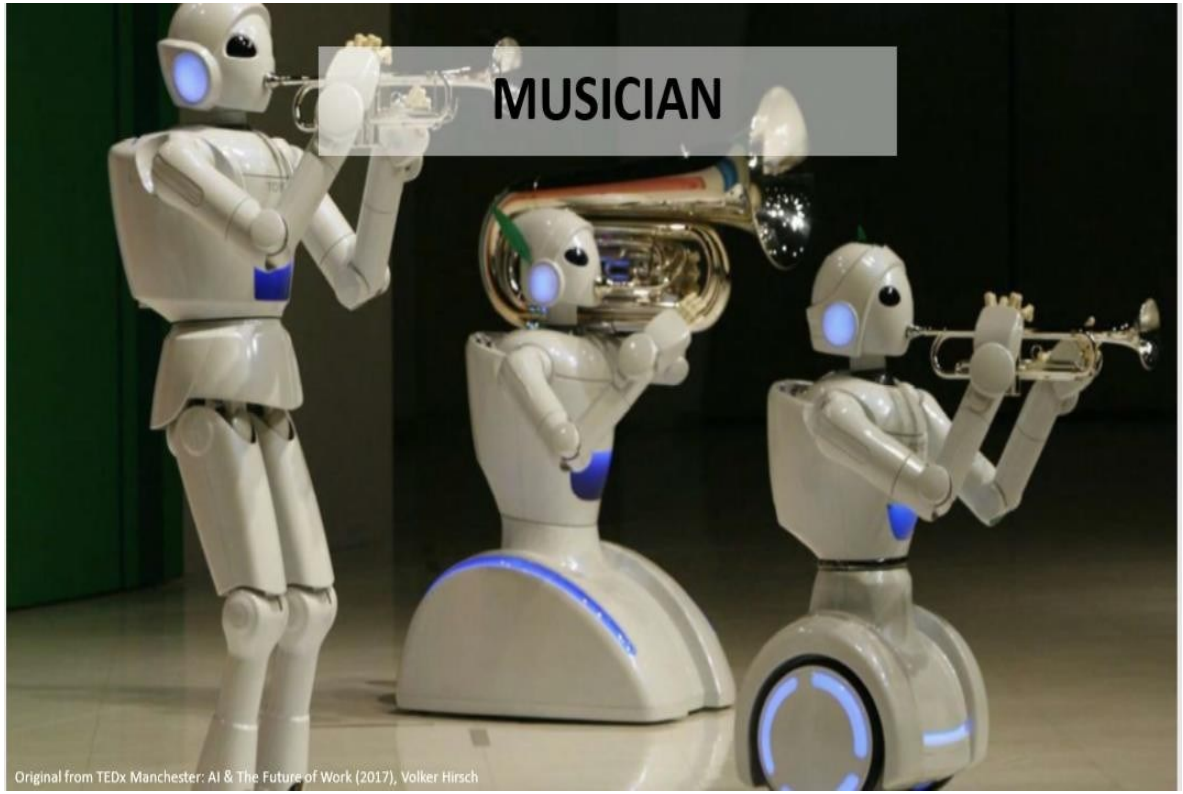
AI in Real Life

AI in Science Fiction Movies



AI in Science Fiction Movies

AI in Music


MUSICIAN
Original from TEDx Manchester: AI & The Future of Work (2017), Volker Hirsch

AI in Agriculture

AI in Delivery Services

AI in Self-Driving Vehicles

AI in Natural Language Processing

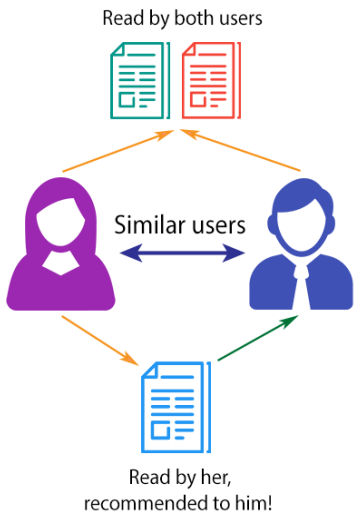
- Web search engines
- Text classification: sentiment, topic
- Spam filtering etc
- Machine translation
- Question answering
- Recommender Systems

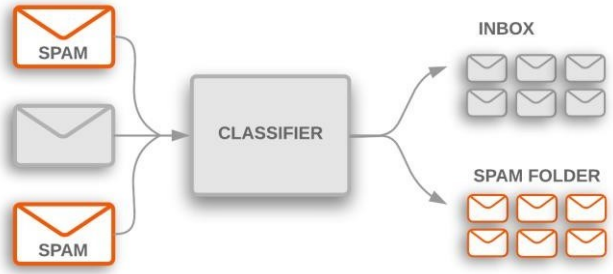
"Il est impossible aux journalistes de rentrer dans les régions tibétaines"

Bruno Philip, correspondant du "Monde" en Chine, estime que les journalistes de l'AFP qui ont été expulsés de la province tibétaine du Qinghai "n'étaient pas dans l'illégalité".

Les faits Le dalaï-lama dénonce l'"enfer" imposé au Tibet depuis sa fuite, en 1959

Vidéo Anniversaire de la rébellion





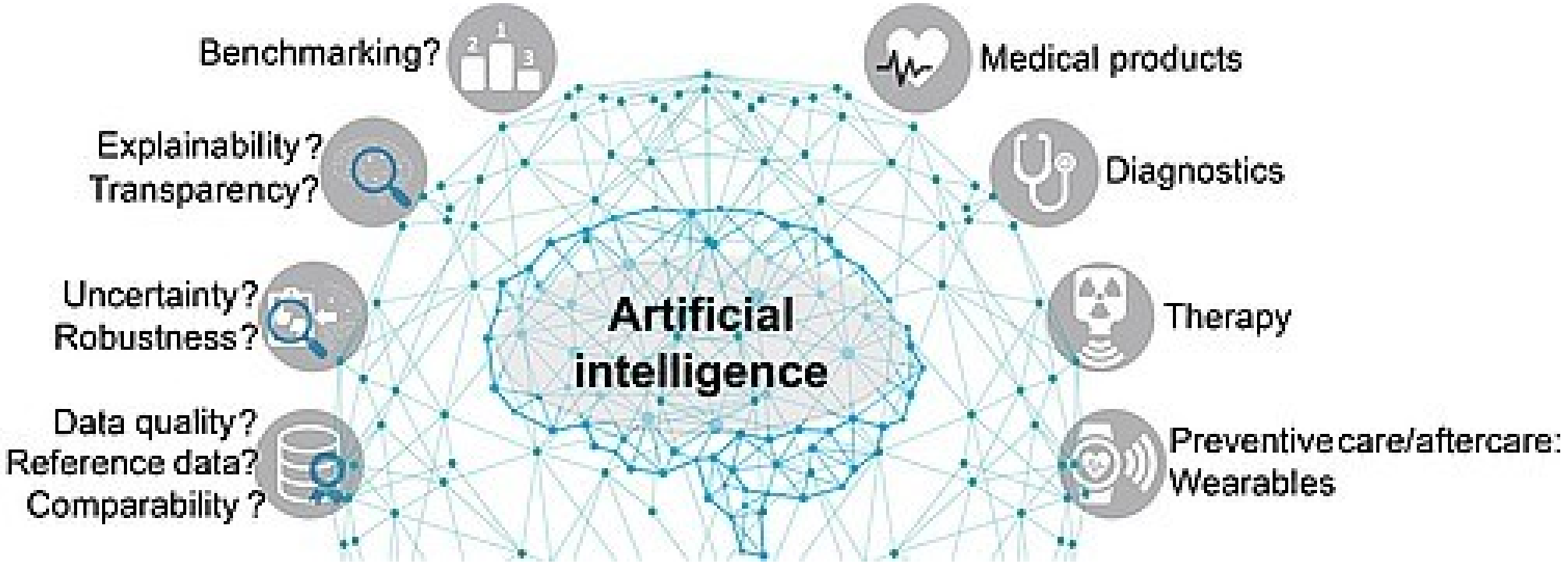
AI in Natural Language Processing

Speech Technologies

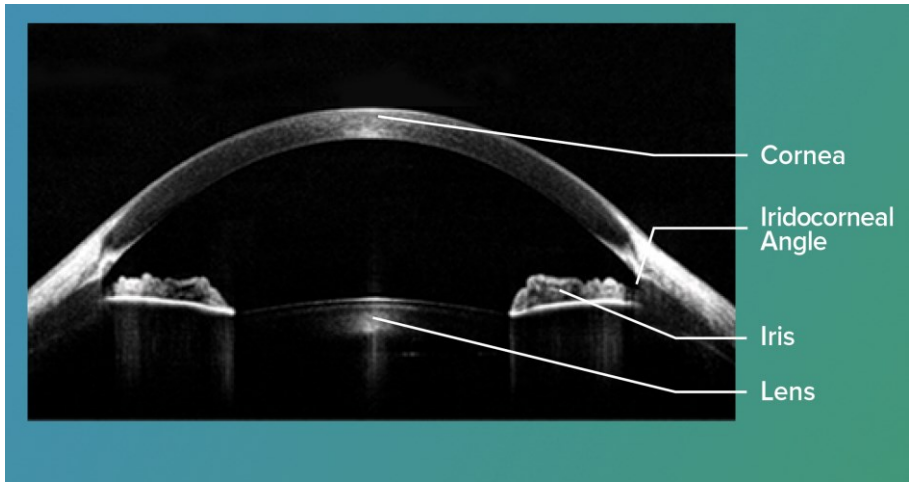
- Siri, Alexa, Cortana, Google Assistant
- Automatic Speech Recognition
- Dialogue systems

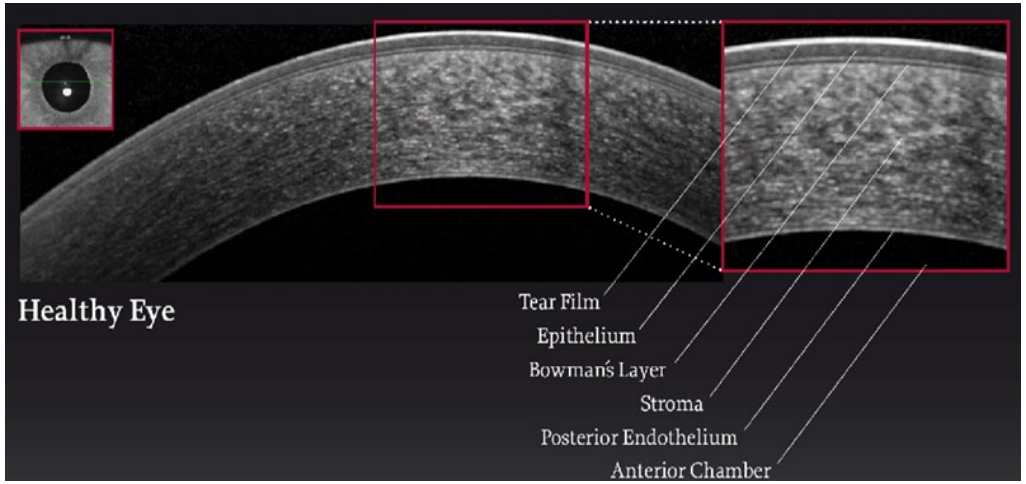
AI in Medicine

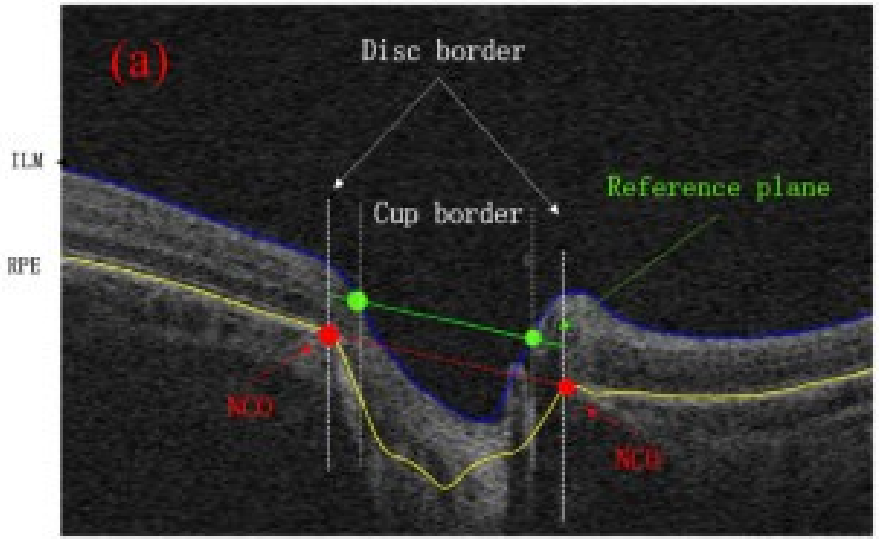
AI in Medicine

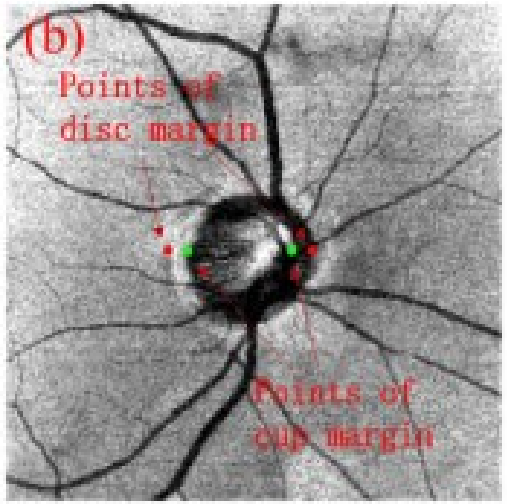


AI in Medicine



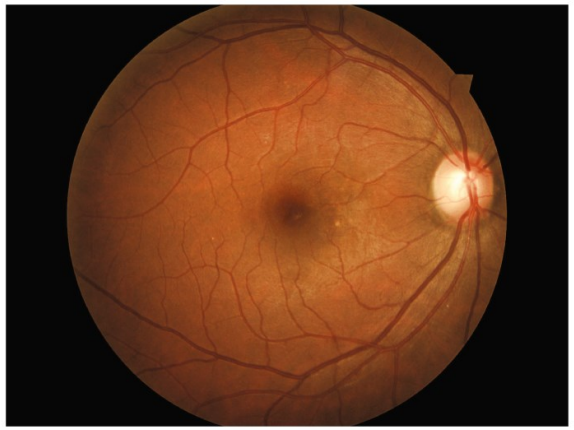




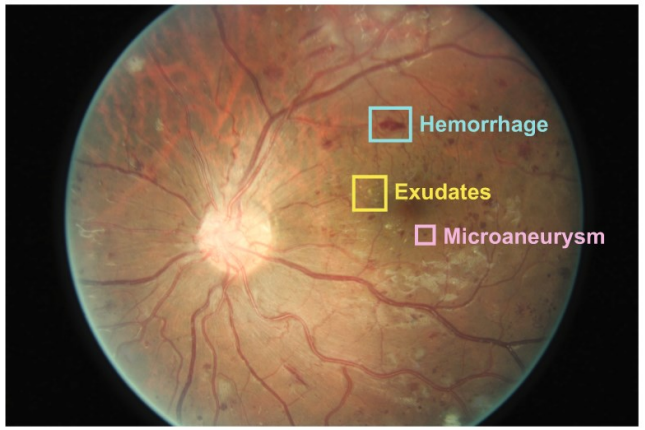


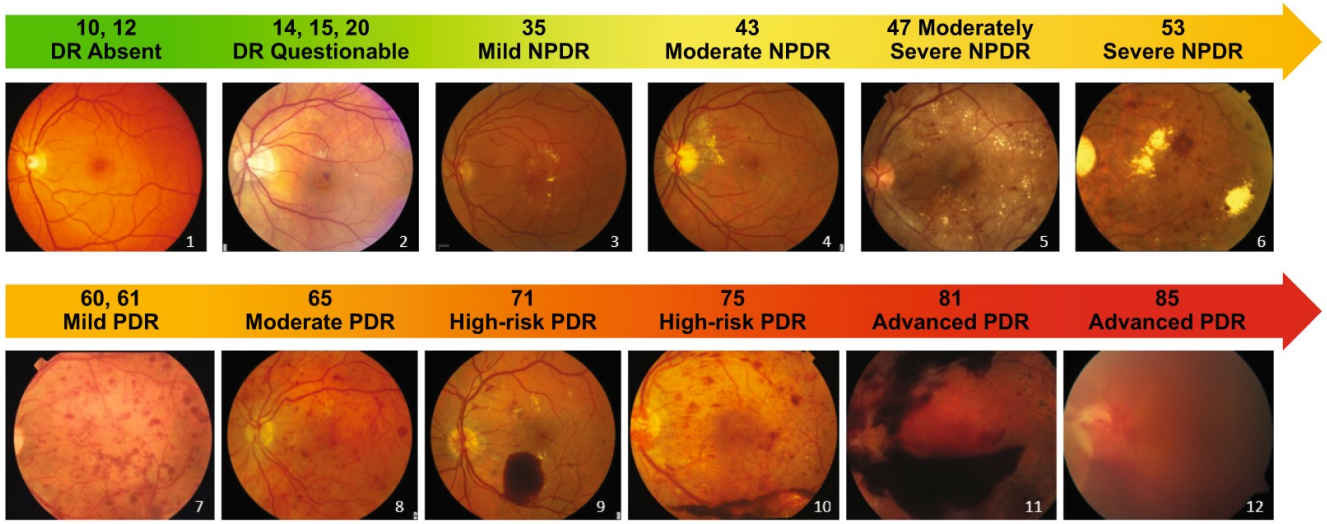
AI in Medicine

Patient without DR

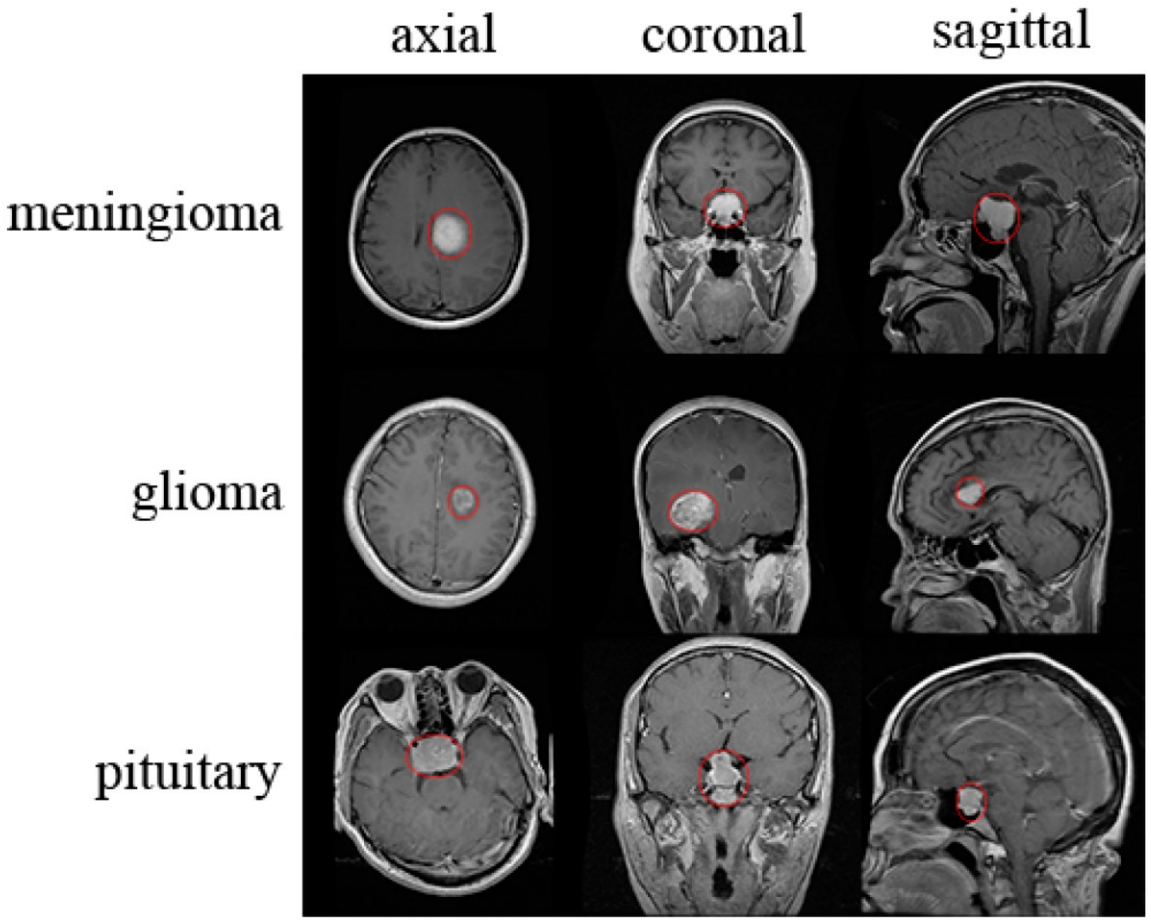


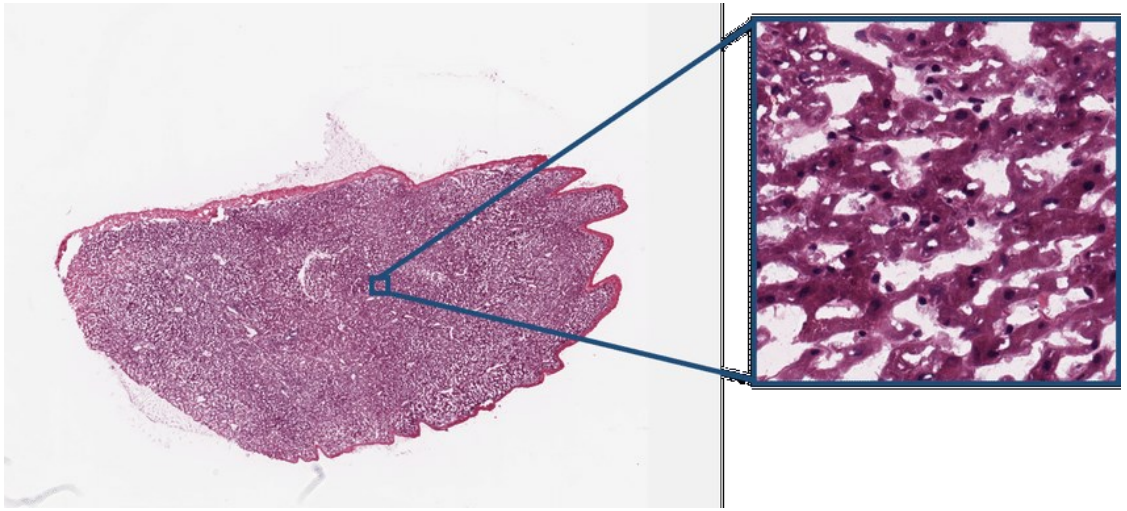
Patient with DR

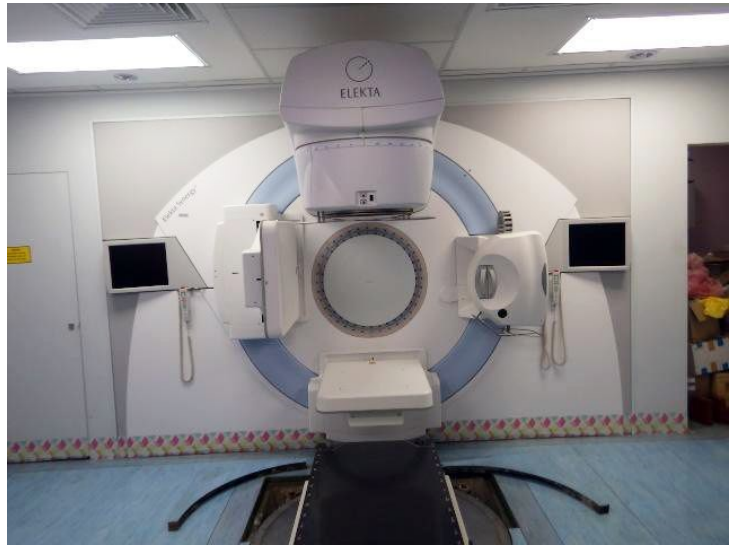




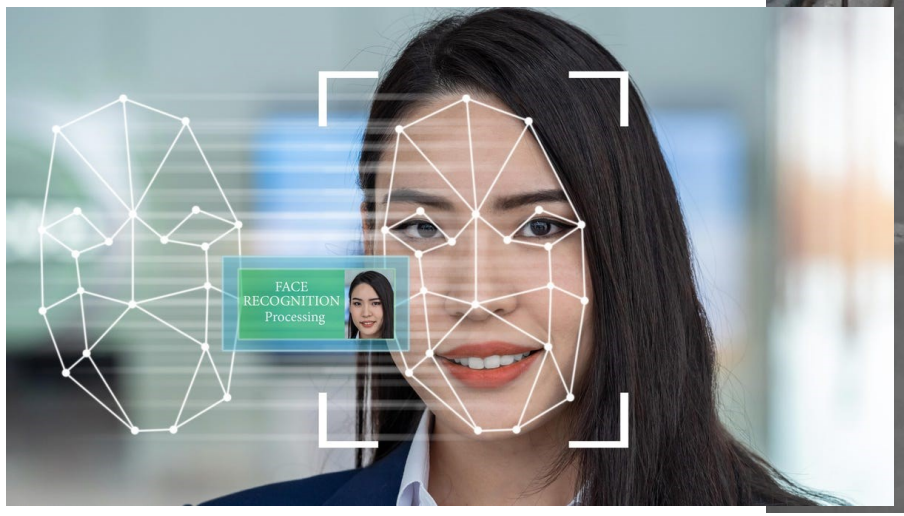
AI in Medicine

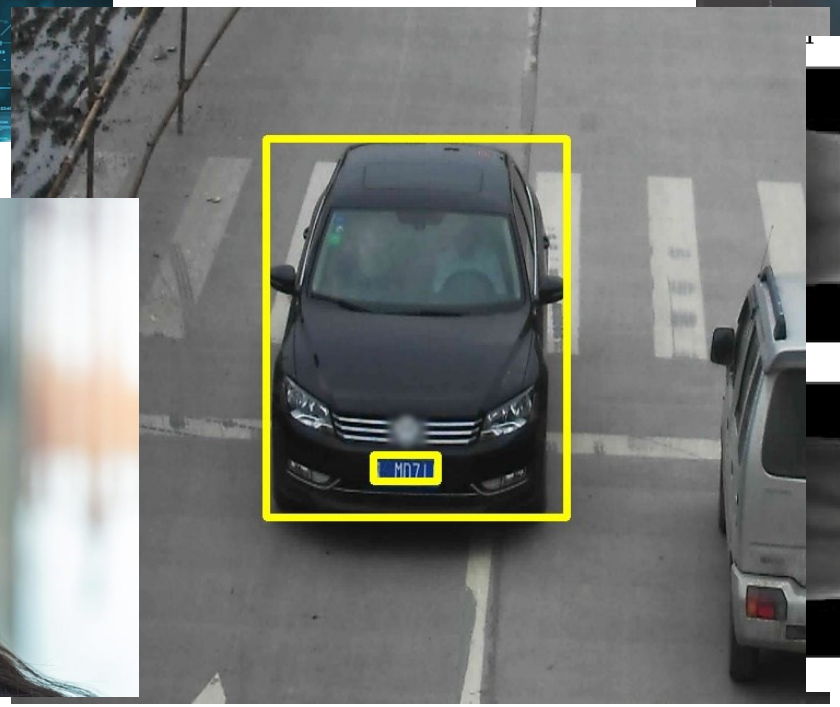






AI in Security





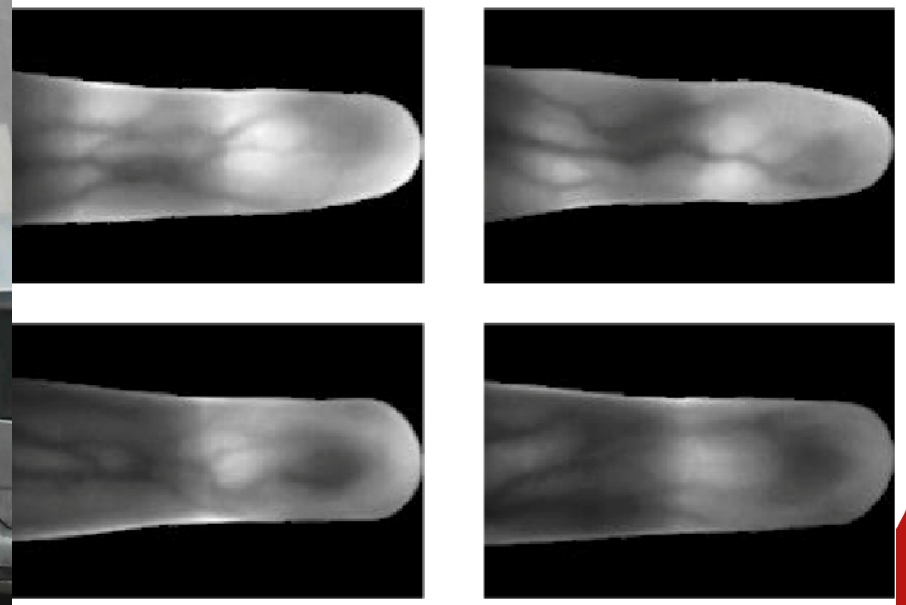
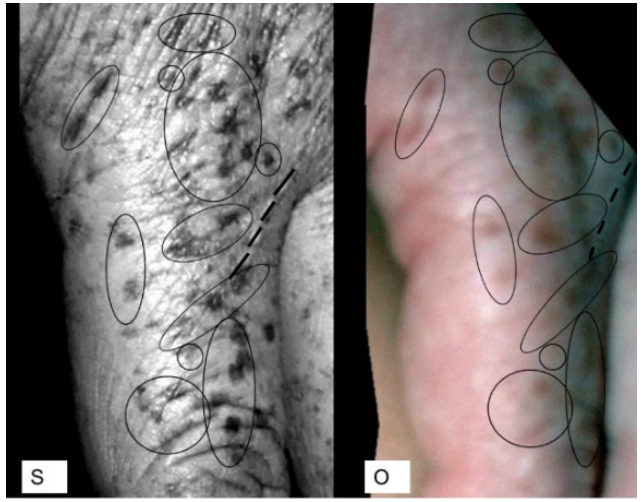


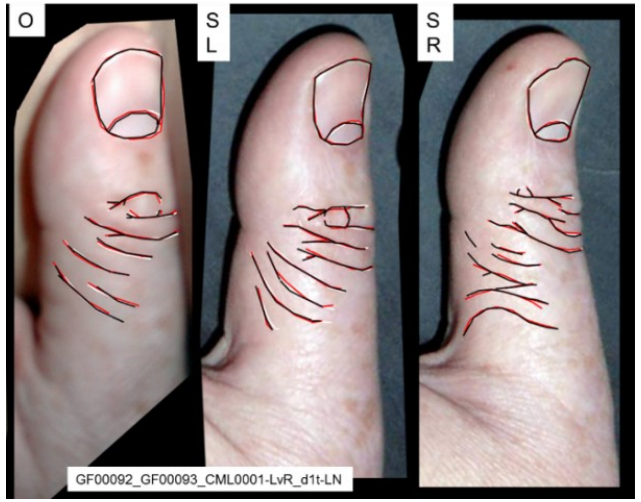
Figure 7. The captured infrared finger vein images

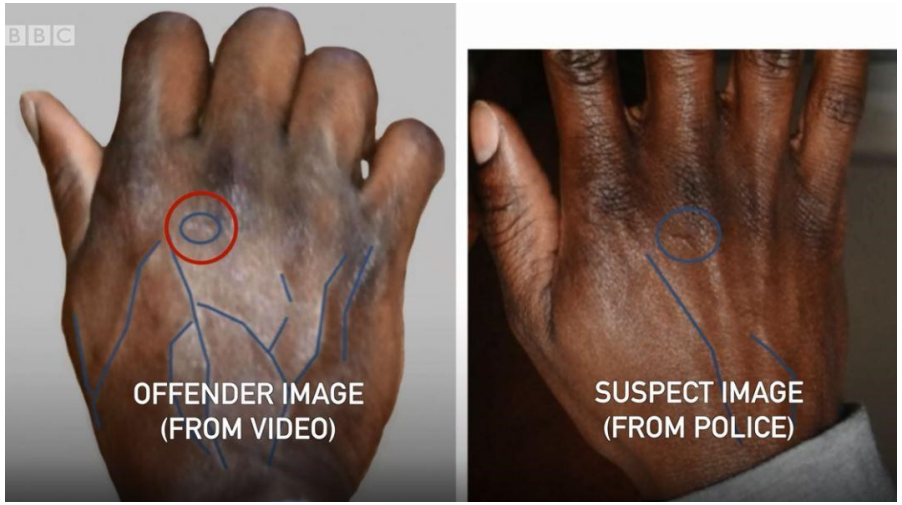
AI in Forensic Investigation

AI in Forensic Identification

AI in Forensic Identification







H-Unique

https://h-unique.lancaster.ac.uk/

SCC361: Artificial Intelligence

Week 1: Introduction to Artificial Intelligence and Machine Learning

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

SCC361: Artificial Intelligence

Introduction to the Module

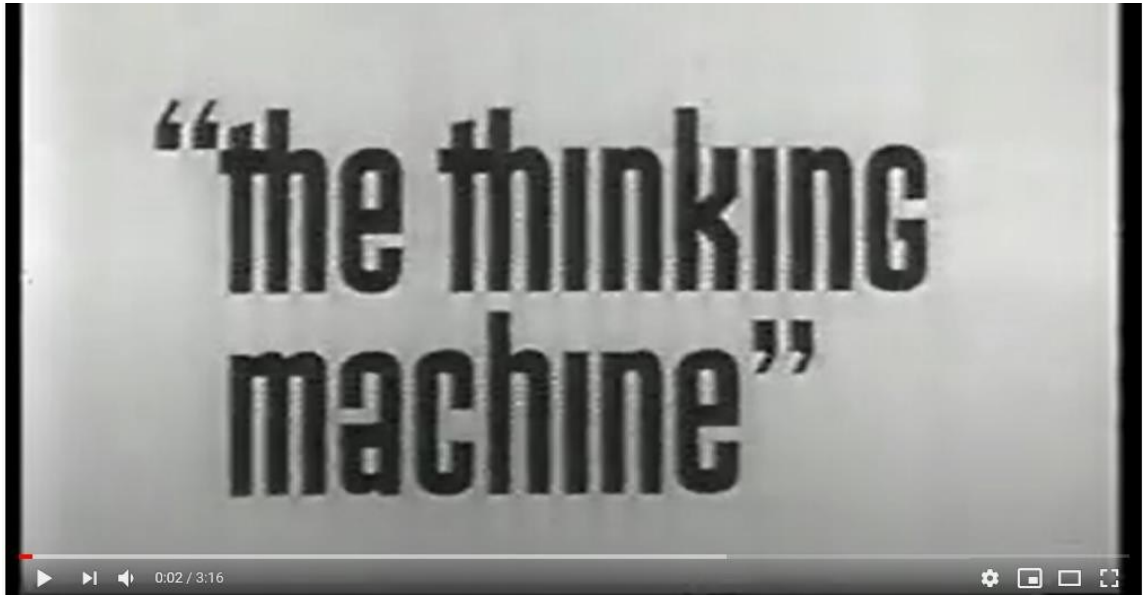
Lectures, Materials and Expectations

Overview of Artificial Intelligence

Overview of Machine Learning

The Thinking Machine

- Can machines really think?
- Interviews by some of the AI pioneers in the 1960s:
  - Jerome Wiesner,
  - Oliver Selfridge,
  - Claude Shannon
- Can a robot marry my daughter?
- Can AI translate write poetry?



https://www.youtube.com/watch?v=aygSMgK3BEM

Human Intelligence

Learning

Reasoning

Perceiving

Understanding of Language

Feeling

What is Artificial Intelligence?

Approach 1

- "The exciting new effort to make computers think ...machines with minds, in the full and literal sense." (Haugeland, 1985)
- "[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning ..." (Bellman, 1978)

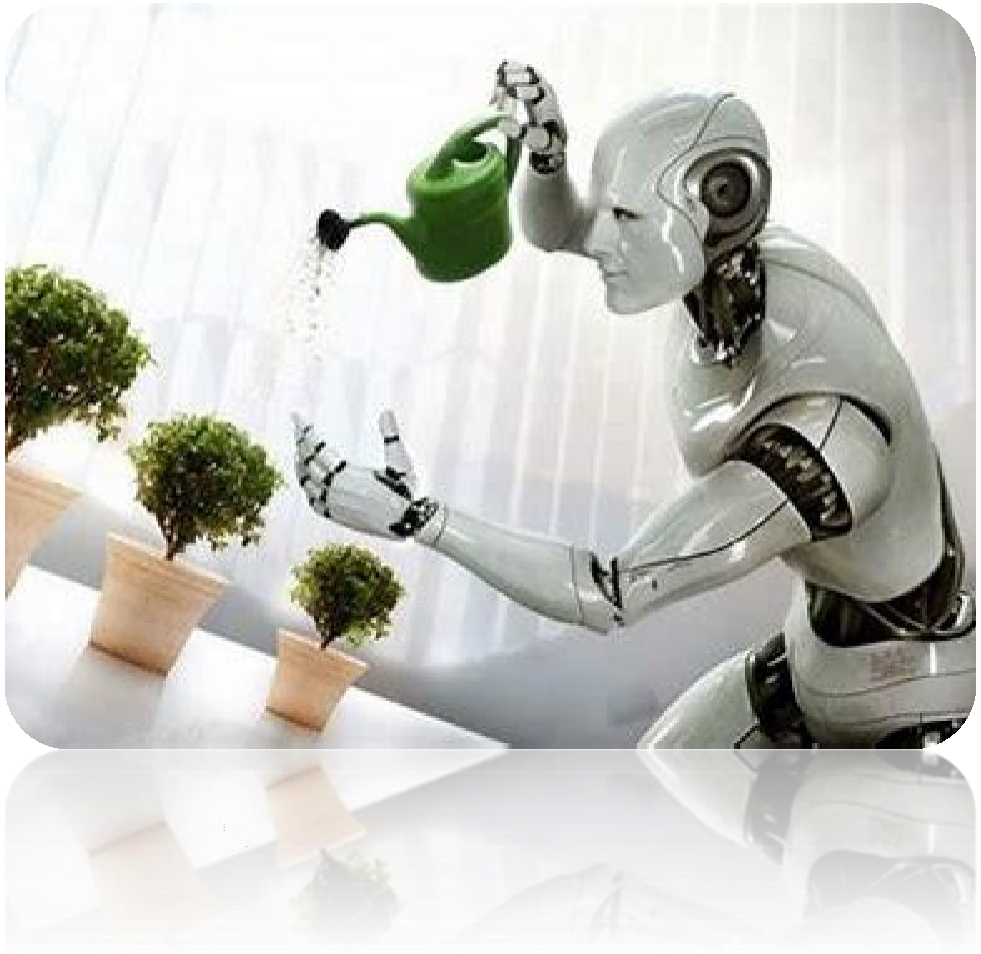
Thinking Humanly

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

What is Artificial Intelligence?

Approach 2

- "The art of creating machines that perform functions that require intelligence when performed by people." (Kurzweil, 1990)
- "The study of how to make computers do things at which, at the moment, people are better." (Rich and Knight, 1991)



Acting Humanly

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

What is Artificial Intelligence?

Approach 3

- "The study of mental faculties through the use of computational models." (Charniak and McDermott, 1985)
- "The study of the computations that make it possible to perceive, reason, and act." (Winston, 1992)

Thinking Rationally

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

What is Artificial Intelligence?

Approach 4: Acting Rationally

- "Computational Intelligence is the study of the design of intelligent agents." (Poole et al., 1998)
- "AI ... is concerned with intelligent behavior in artifacts." (Nilsson, 1998)

Acting Rationally

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Approaches to defining AI

	Human	Rational
Thinking	**Systems that think like humans**	**Systems that think rationally**
Acting	**Systems that act like humans**	**Systems that act rationally**

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Approaches to defining AI

	Human	Rational
Thinking	**Systems that think like humans** • Cognitive modelling approach • Introspection, psychological experiments, brain imaging • Cognitive Science	Systems that think rationally
Acting	Systems that act like humans	Systems that act rationally

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Approaches to defining AI

	Human	Rational
Thinking	**Systems that think like humans** • Cognitive modelling approach • Introspection, psychological experiments, brain imaging • Cognitive Science	**Systems that think rationally** • Laws of thought approach • "Logicist" tradition • Mostly rule-based • Logic
Acting	Systems that act like humans	Systems that act rationally

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Approaches to defining AI

	Human	Rational
Thinking	**Systems that think like humans** • Cognitive modelling approach • Introspection, psychological experiments, brain imaging • Cognitive Science	**Systems that think rationally** • Laws of thought approach • "Logicist" tradition • Mostly rule-based • Logic
Acting	**Systems that act like humans** • The (total) Turing Test • Requires the 6 disciplines • NLP, KR, Reasoning, ML, Computer vision, Robotics	Systems that act rationally

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Approaches to defining AI

	Human	Rational
Thinking	**Systems that think like humans** • Cognitive modelling approach • Introspection, psychological experiments, brain imaging • Cognitive Science	**Systems that think rationally** • Laws of thought approach • “Logicist” tradition • Mostly rule-based • Logic
Acting	**Systems that act like humans** • The (total) Turing Test • Requires the 6 disciplines • NLP, KR, Reasoning, ML, Computer vision, Robotics	**Systems that act rationally** • The rational agent approach • Autonomous, perceptive, persistent, adapts to change • Creates and pursues goals

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

What is an Agent?

An **agent** ‘acts’ (does something) within an **environment**

- e.g. worms, dogs, thermostats, airplanes, robots, humans, companies, and countries.

An agent acts **intelligently** if:

- action is appropriate for circumstances and goals
- flexible to changes in environment and goals
- learns from experience
- makes appropriate choices given perceptual and computational limitations

Computational Agent

A **computational agent** is:

- An agent whose decisions and actions can be explained in terms of computation.
- Decision can be broken down into primitive operations that can be implemented in a physical device.
- Computations can take many forms
  - The human brain (“wetware”)
  - Computers (“hardware”)
- Non computational agents:
  - wind, rain, etc.

Rational Agent

- **Rational agent** acts to ‘achieve the best outcome or, when there is uncertainty, the best expected outcome.’
- AI focuses on building the *general principles* of **rational agent** and *components* for constructing them
- Two key advantages of the **rational-agent** over others:
  - Amenable to scientific development than approaches on human thoughts and behaviour
  - It is more general than the “laws of thought” approach
- Also deals with **limited rationality** – acting appropriately with limited computations

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig

Intelligence

- AI is the field that studies the *synthesis* and *analysis* of **computational agents** that **act intelligently** - *Poole & Markworth*

- An agent acts **intelligently** if:
  - action is appropriate for circumstances and goals
  - flexible to changes in environment and goals
  - learns from experience
  - makes appropriate choices given perceptual and computational limitations

Artificial Intelligence: Foundations of Computational Agents, 2017, Poole & Markworth

Intelligence

- AI is the field that studies the *synthesis* and *analysis* of **computational agents** that **act intelligently** - *Poole & Markworth*

- An agent acts **intelligently** if:
  - action is appropriate for circumstances and goals
  - flexible to changes in environment and goals
  - learns from experience
  - makes appropriate choices given perceptual and computational limitations

it is 'rational'

Artificial Intelligence: Foundations of Computational Agents, 2017, Poole & Markworth

Definition of AI

Artificial intelligence, or **AI** is the field that studies the *synthesis and analysis of* ***computational agents*** *that* ***act rationally***

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig
Artificial Intelligence: Foundations of Computational Agents, 2017, Poole & Markworth

Goals of AI

Two types of goals: **Scientific** and Engineering

- **Scientific goal** – understand the principles of intelligent behaviour:
  - Analysis of natural and artificial agents
  - Formulating and testing hypothesis
  - Designing, building and experimenting with computational agents
- Uses a general scientific approach
- Focuses on building **empirical systems**
- And **not** on the final **applications** that could be deployed to use

Goals of AI

Two types of goals: Scientific and **Engineering**

- **Engineering goal** – concerned with constructing intelligent agents
- Focuses on the **design** and **synthesis** of useful, intelligent artefacts.
- Builds agents that act intelligently
- Agents that are useful in many real-world applications

Business Benefits of AI

- Workflow/Process automation
  - Use of bots for routine, repetitive tasks
- Enhance creative tasks
  - More time and tools to explore creative functions
- Increased accuracy
  - Human errors can be reduced
- Better predictions & improved decision making
  - Predictions of risks, performance targets, tailored product offerings etc

Social benefits of AI

- Healthcare
  - There is a huge effort in mobilizing AI for health.
- Smart cities, transportation, security
  - Maps, navigation systems, unmanned vehicles, route planning, security
- Forecasts and predictions
  - Weather, natural disasters, earthquakes, hurricanes, stock prices, economic
- Agriculture
  - Real-time data analytics help farmers to maximise their crop yields and profits
- Overall lifestyle

Risks and Challenges of AI

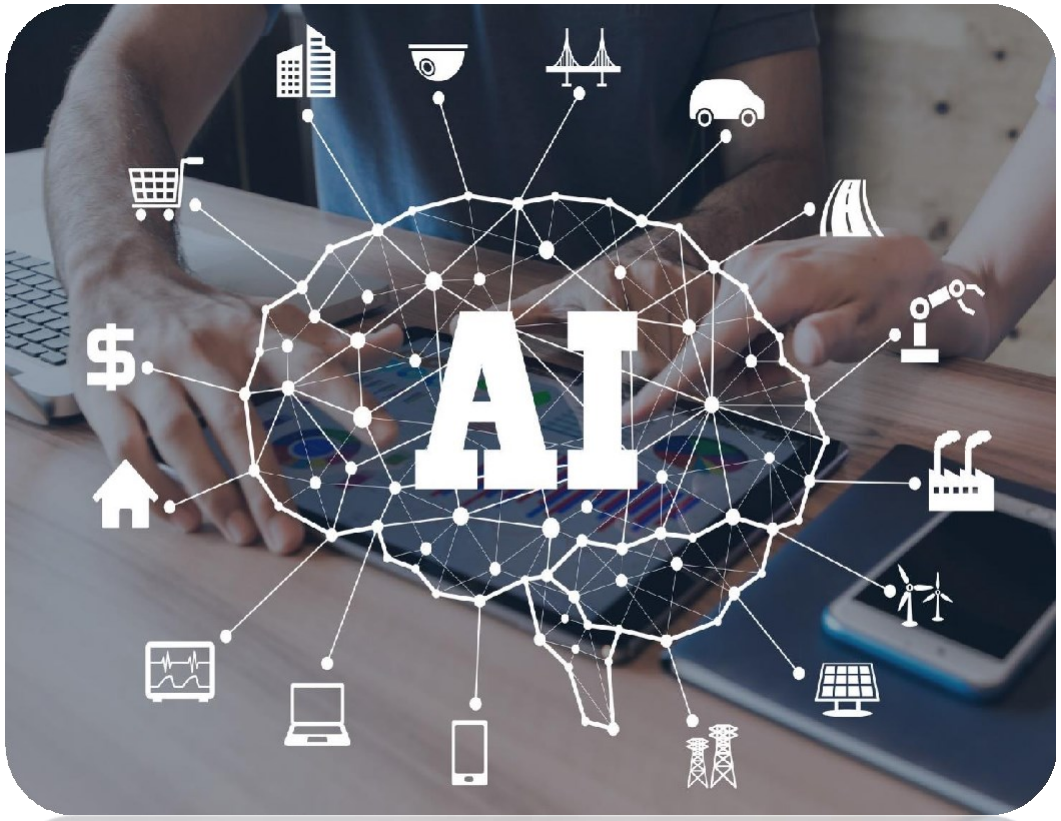
- Safety and security
  - Driverless cars can be hacked
  - Failed Facebook AI chatbot experiment
  - Racist hijack of Microsoft AI Tweeter feed
- Trust and social manipulation
  - Facebook-Cambridge Analytica Scandal
- Explainable (or Interpretable) AI (XAI)
  - Deep neural models are naturally opaque
- Possible job losses
  - “AI will replace more than 75 million jobs by 2022” – World Economic Forum

Ethical Concerns of AI

- Accountability
  - If AI violates ethical rules, who will be responsible?
- Accuracy, bias, privacy and inequality
- AI learns from data provided by humans which may encode human biases and prejudices
  - Facial recognition to ‘predict criminals’ sparks row over AI bias – BBC
  - IBM abandons “biased” facial recognition tech – BBC
- Technological social responsibility (TSR)
  - a conscious alignment between short- and medium-term business goals and longer-term societal ones – McKinsey Quarterly, August, 2019

AI Summary

- Artificial Intelligence: An overview
  - Application, history, foundations of AI
- Definition of AI
- Goals of AI
- AI and the Society
  - Benefits
  - Risk and Challenges
  - Ethical Issues



Questions?

SCC361: Artificial Intelligence

Introduction to the Module

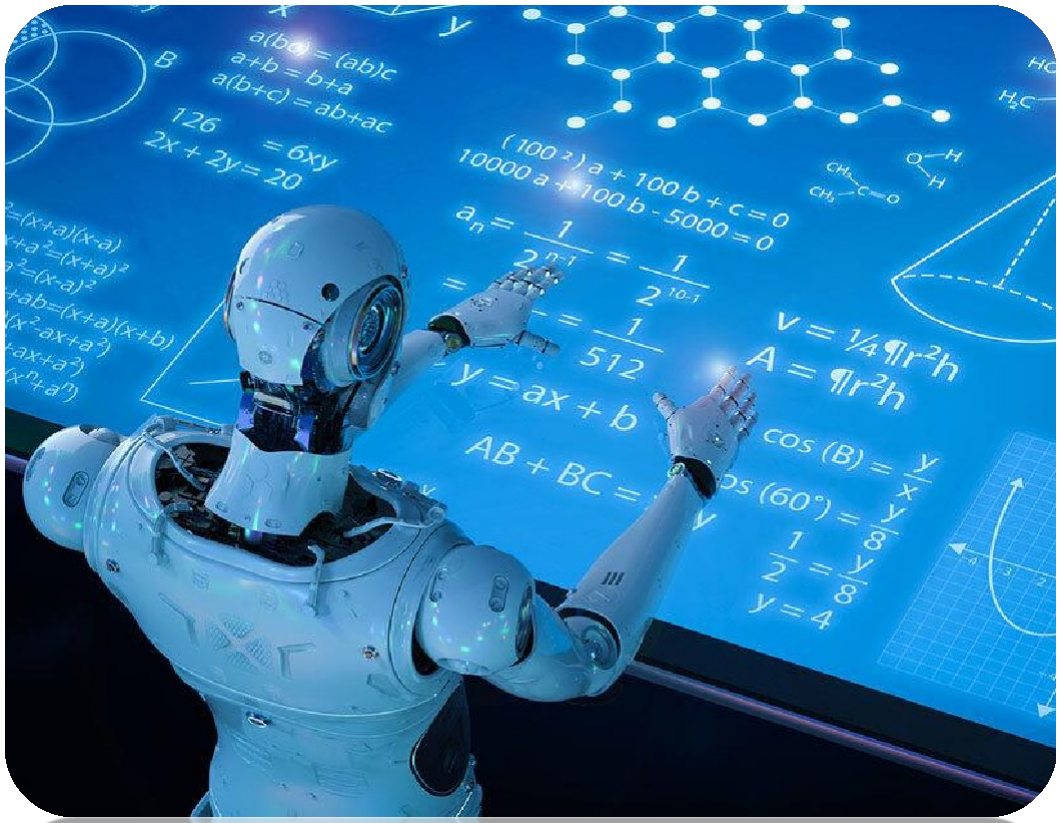
Lectures, Materials and Expectations

Overview of Artificial Intelligence

Overview of Machine Learning

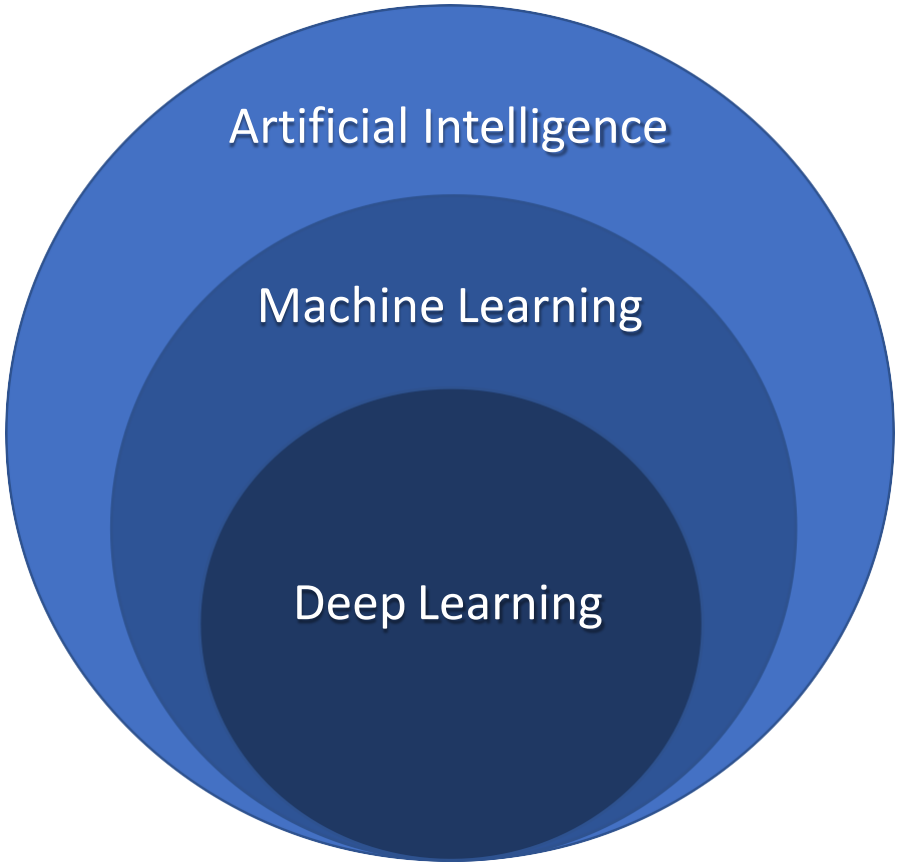
Introduction to Machine Learning

- Overview of Machine Learning
  - AI and ML, Definitions of ML, How to learn
- Types of Machine Learning
  - Supervised, unsupervised, semi supervised
- Supervised Learning
  - Classification and regression
- Unsupervised Learning
  - Clustering and association



AI and Machine Learning

- AI systems were mostly **rule-based**
  - i.e. depended on hand-crafted rules
- Machine learning drives AI
  - Learning algorithms create a logical mapping from data to output
- Deep learning:
  - a subset of ML with additional layers to learn deeper representations data



What is Machine Learning?

Early definition of machine learning

"Field of study that gives computers the ability to learn without being explicitly programmed"

– Arthur Samuel (1959)

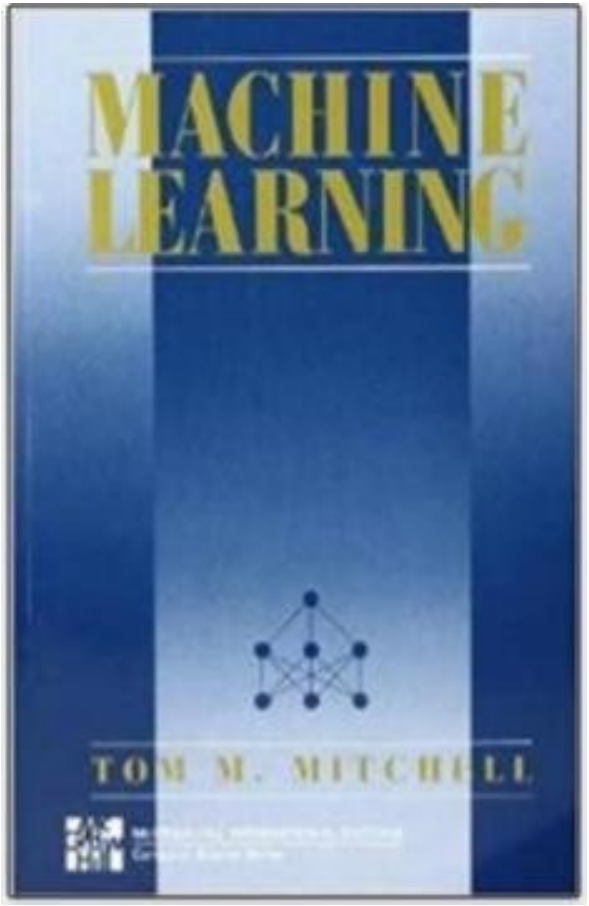
- ML pioneer that built first "self-learning" program that played checkers by learning from experience
- Inverted alpha-beta pruning widely used in decision tree searching

What is Machine Learning?

Another popular definition:

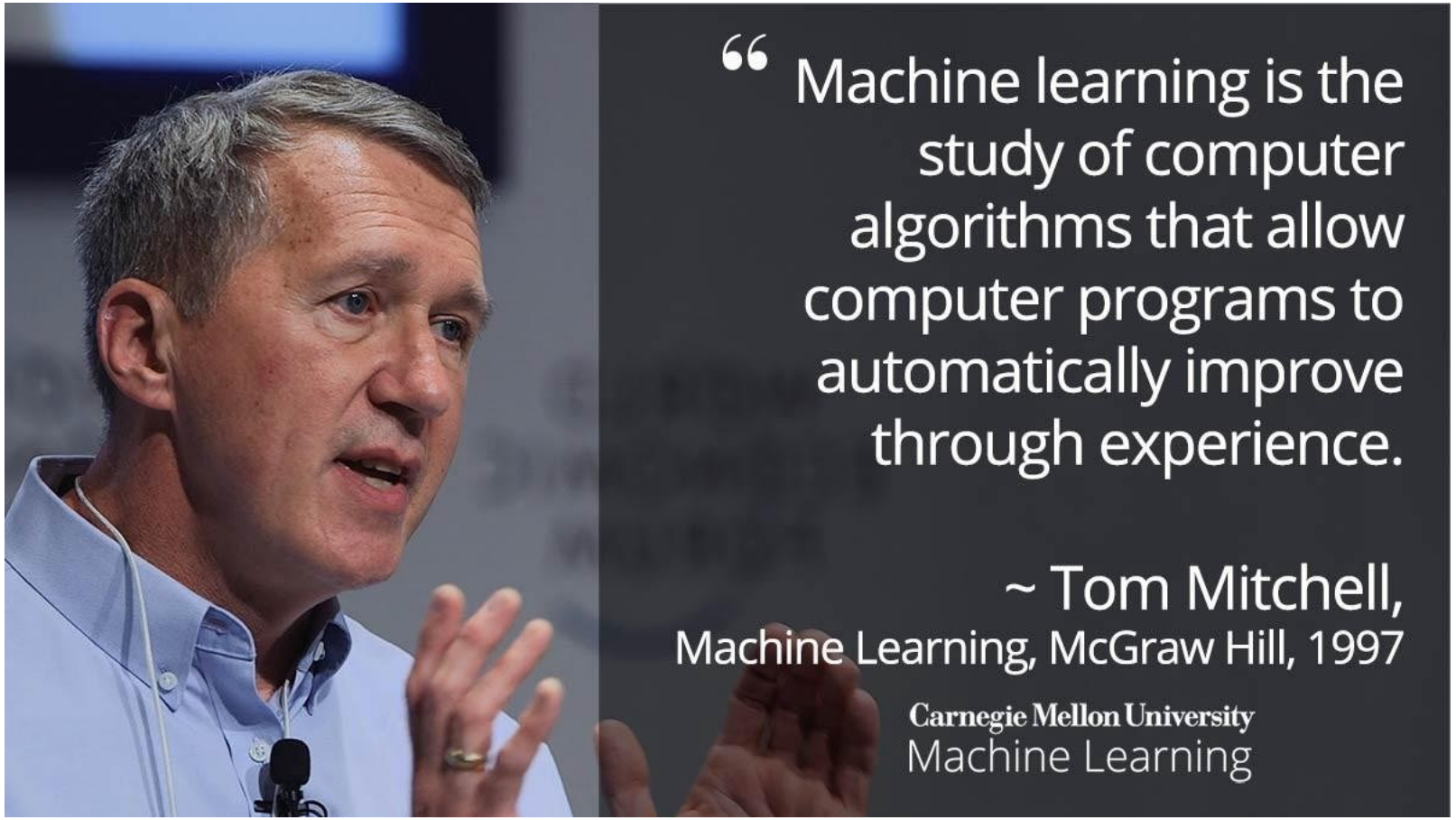
*"A computer is said to **learn** from experience **E** with respect to task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improved with experience **E**"*

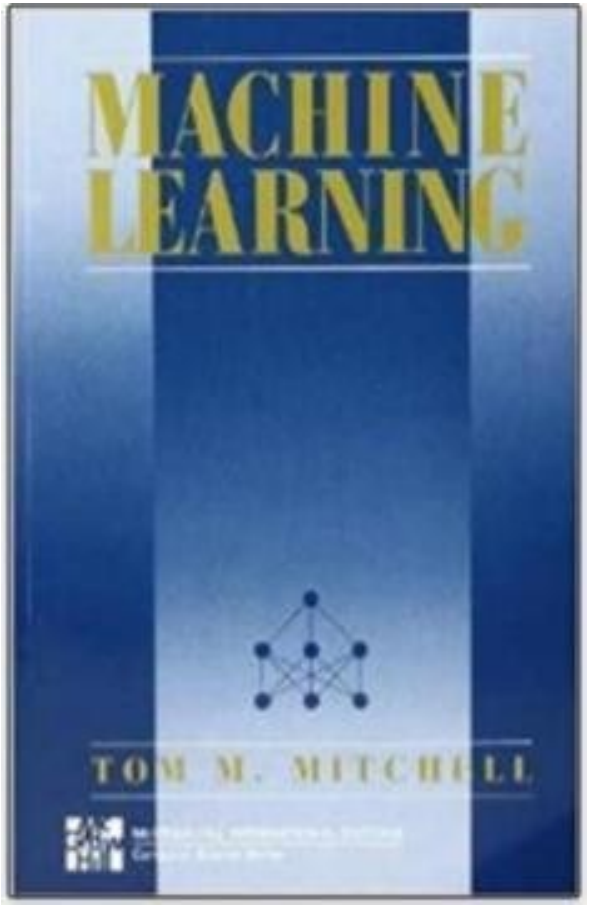
– Tom Mitchell (1997)



- Again, the key is learning from experience
- Not explicitly programmed

What is Machine Learning?





Spam or not SPAM

Given this definition:

*"A computer is said to **learn** from experience **E** with respect to task **T** and some performance measure **P**, if its performance on **T**, as measured by **P**, improved with experience **E**"*

My email program watches me mark some emails as spam, and improves on filtering spams. What is the T, E and P in the setting?

a. Watching me label emails as spam

b. Classifying emails as spam or not spam

c. The fraction of emails correctly classified as spam or not

d. None of the above – this is not a machine learning problem

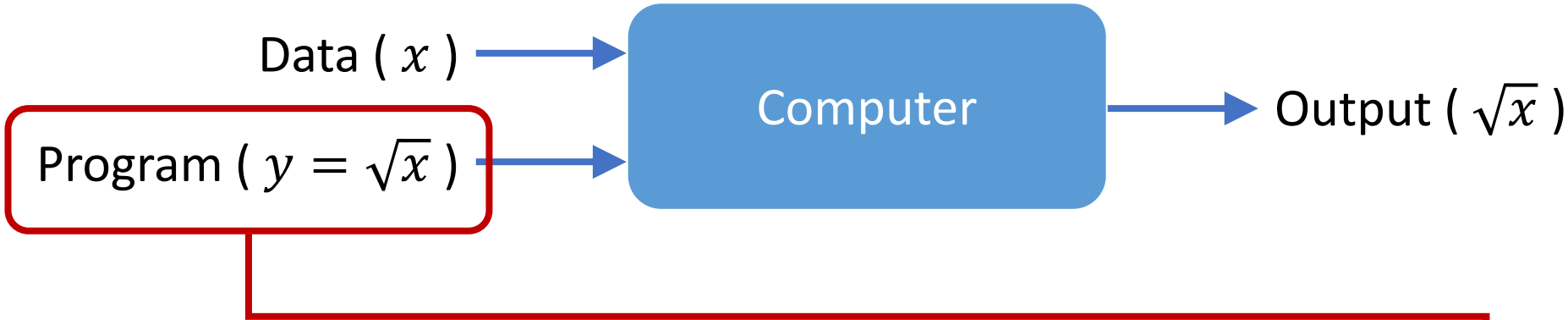
What is Machine Learning?

- Consider the function $y = f(x)$ (e.g. $f(x) = \sqrt{x}$)
- Traditional Programming (Software 1.0)

- Machine Learning (Software 2.0)

What is Machine Learning?

- Consider the function $y = f(x)$ (e.g. $f(x) = \sqrt{x}$)
- Traditional Programming (Software 1.0)



- Machine Learning (Software 2.0)

How things are learned

- Memorization
  - Accumulation of individual facts
  - Limited by
    - Time to observe facts
    - Memory to store facts

Declarative knowledge

How things are learned

- Memorization
  - Accumulation of individual facts
  - Limited by
    - Time to observe facts
    - Memory to store facts
- Generalization
  - Deduce new facts from old facts
  - Limited by accuracy of deduction process
    - Essentially a predictive activity
    - Assumes that the past predicts the future

Declarative knowledge

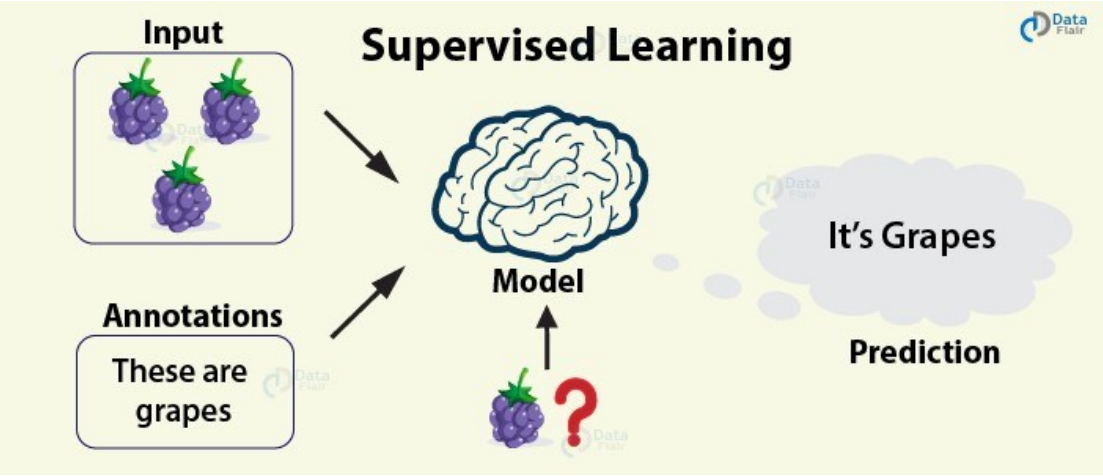
Better

Imperative knowledge

Types of Machine Learning

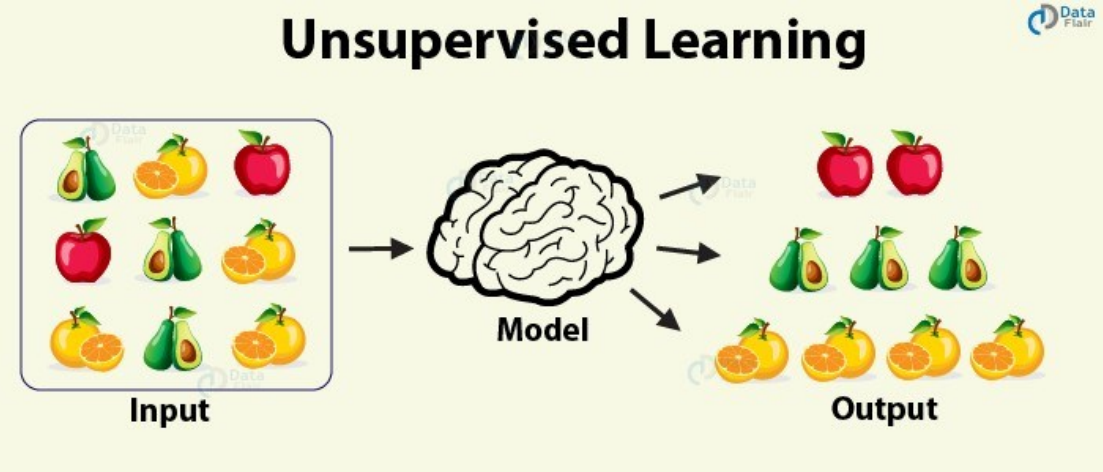
Supervised Learning

- Classification
- Regression



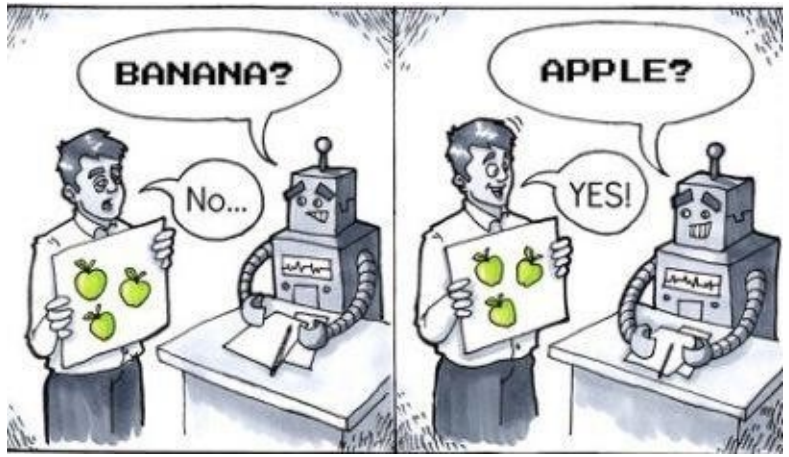
Unsupervised Learning

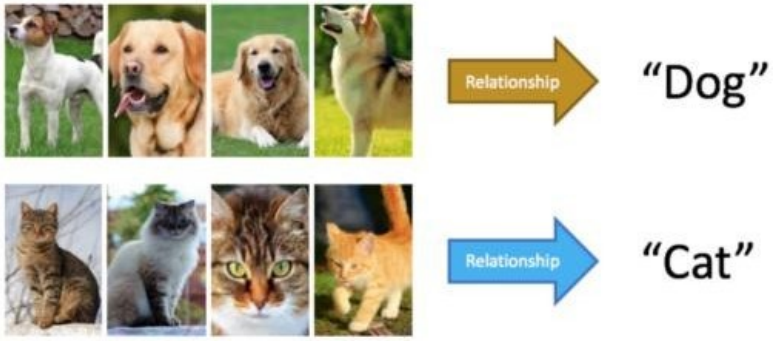
- Clustering
- Association



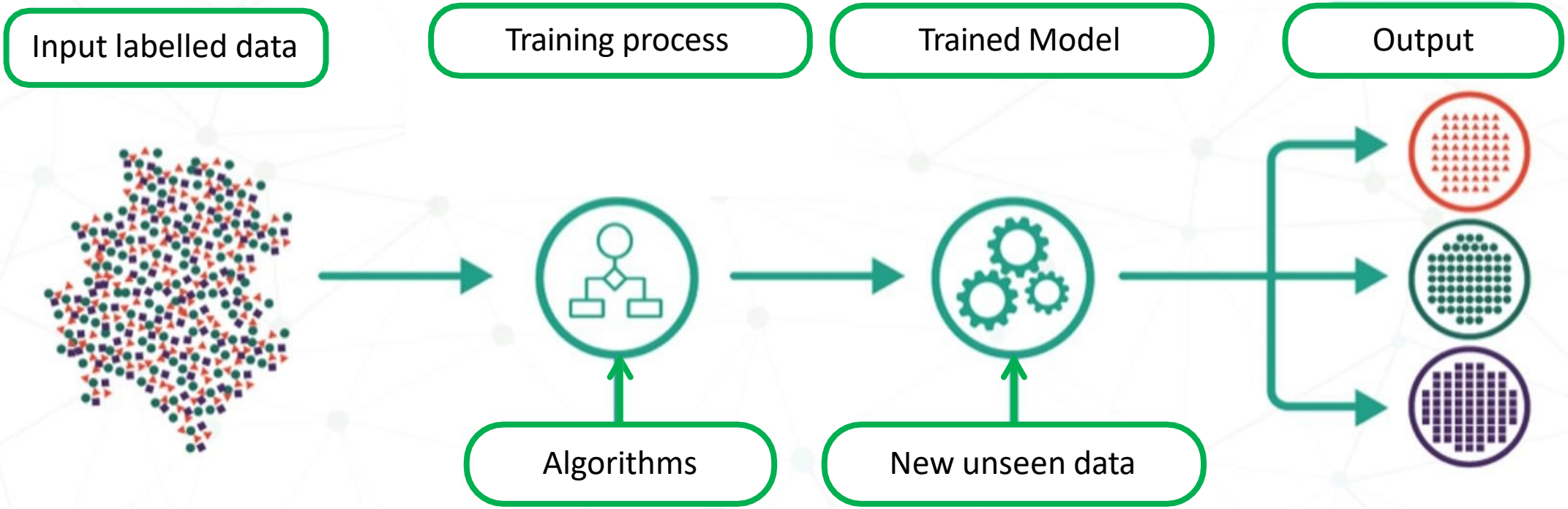
Supervised Learning

- The algorithm learns to map an **input** to a **particular output**.
- Instances of data are presented along with their **correctly labelled** output
- Similar to a **teacher-student** scenario
- The algorithm learns from **experience** to predict new unseen data
- Two broad categories:
  - Regression
  - Classification



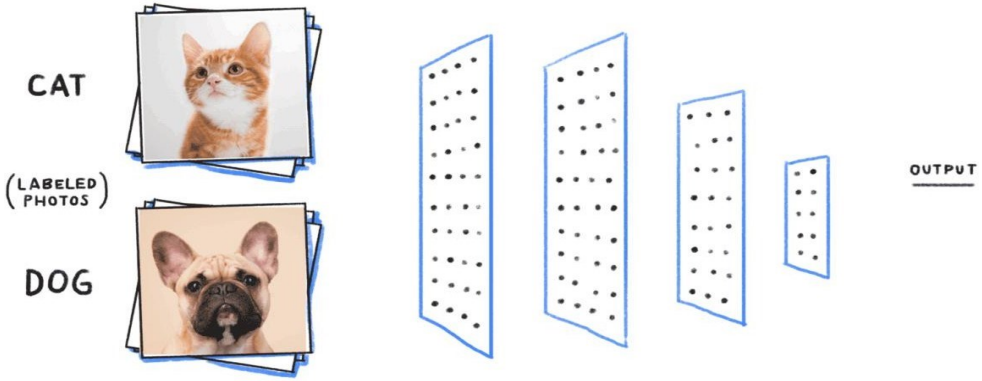


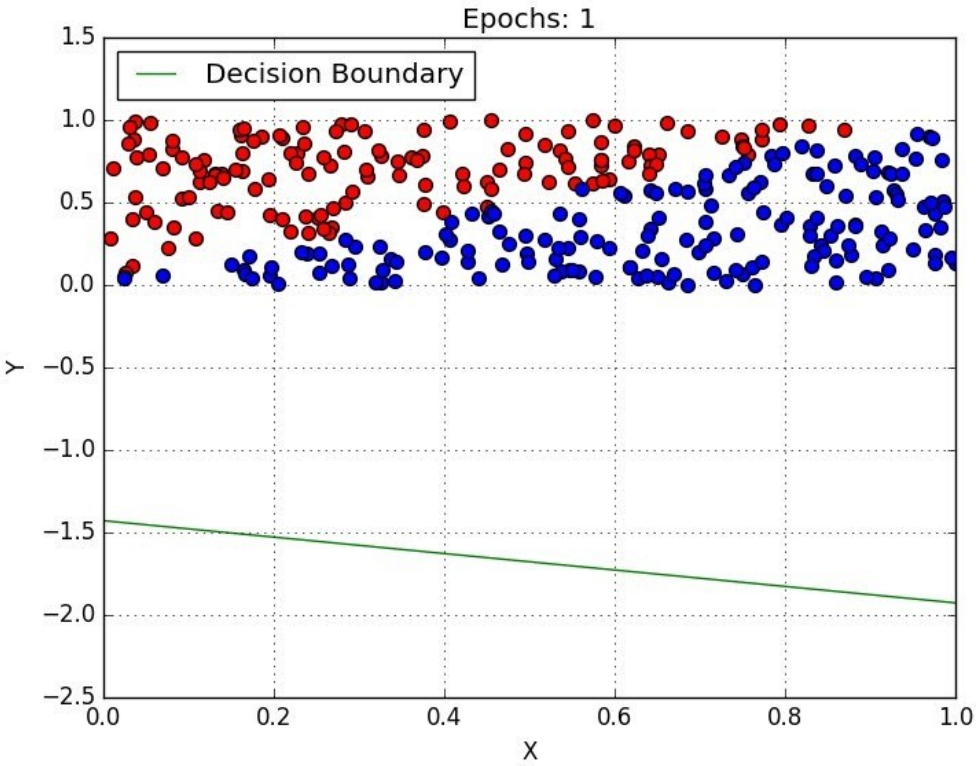
Supervised Learning



Classification

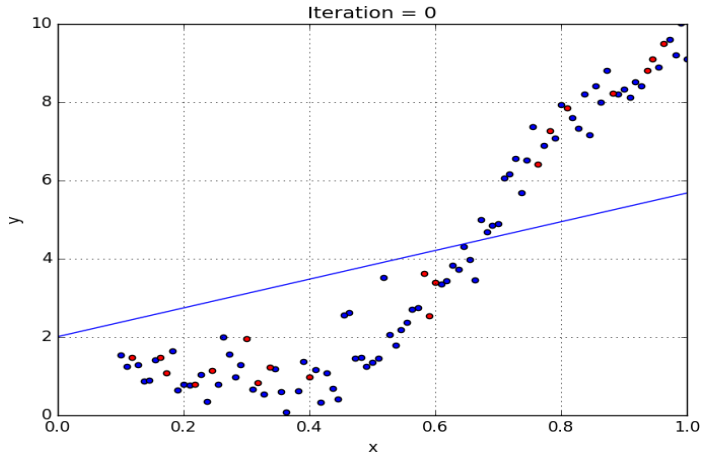
- Learns from labelled data (supervised)
- Predicts a **category** or a **class**
  - Cats|Dogs
  - Spam|Ham
  - Cancer|Not Cancer
- Attempts to separate the data into specific categories (or classes or labels)

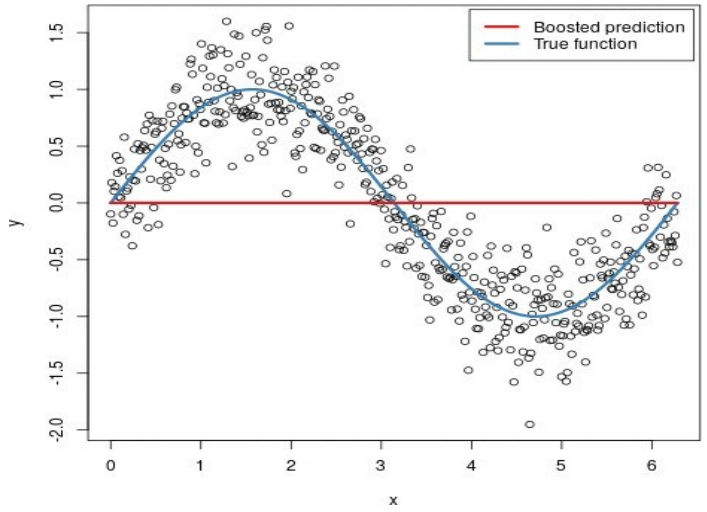




Regression

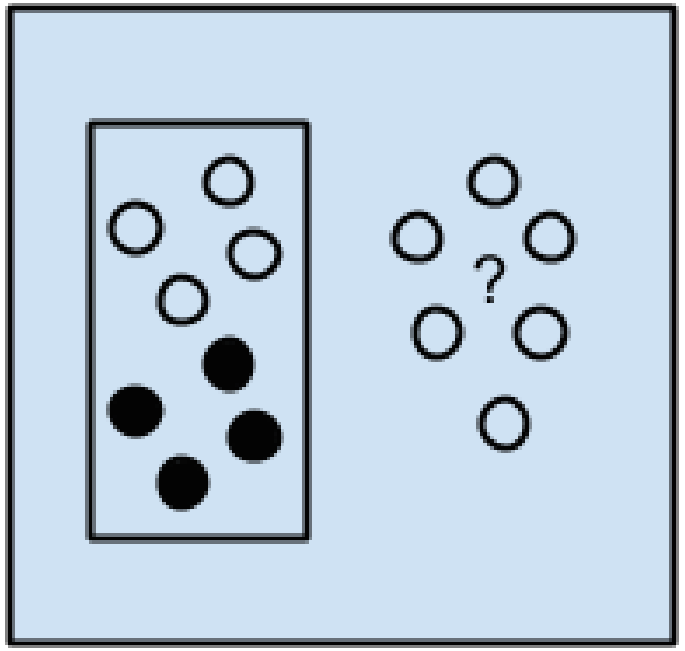
- Learns from labelled data (supervised)
- Predicts a **continuous-valued output**
  - height, price, duration etc.
- Consider a function $y = f(x)$
  - we want our model to predict y_i given x_i
  - x_i not seen during training
- Typically fits some linear or quadratic curve of the data plot
- Linear or logistic regression algorithms are often used





Supervised Learning Algorithms

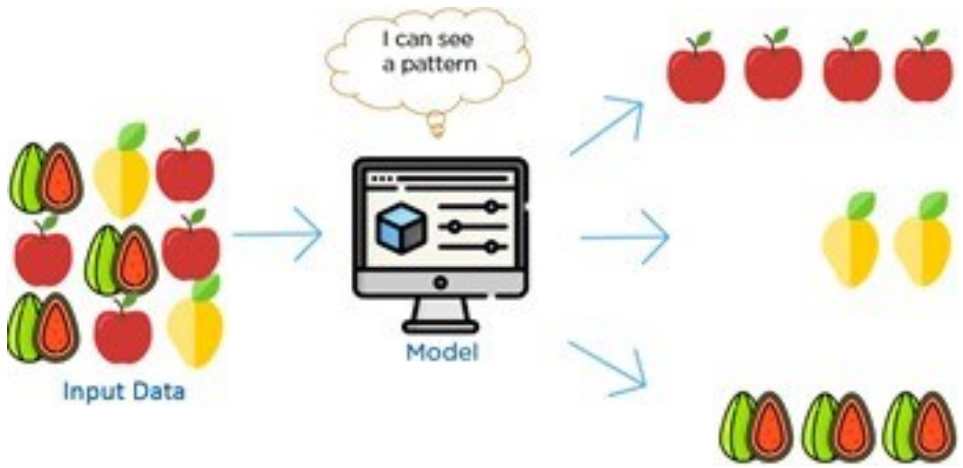
- Input data = training data
  - with labels e.g. spam/ham or stock price at t
- In training
  - the model makes a prediction and is corrected if the prediction is wrong
- Training process continues until a desired accuracy is achieved
- Problem types: Classification and Regression
- **Algorithms**:
  - Logistic Regression
  - Back Propagation Neural Network.

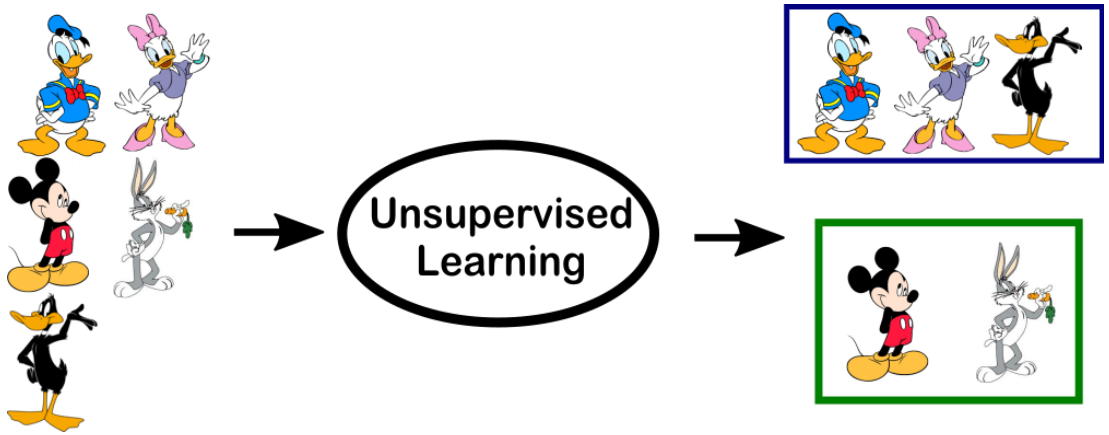


Supervised Learning Algorithms

Unsupervised Learning

- Remember the function $y = f(x)$
- With unsupervised learning, only the input data, x, is available
- There are no corresponding labels (classes or categories) i.e. no output variable, y
- Aims at modelling the underlying structure of the data
- Two main categories
  - Clustering
  - Association



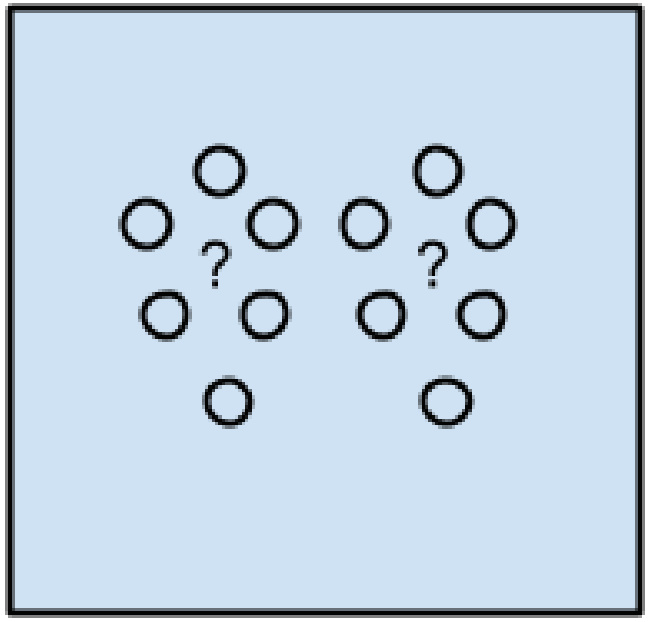


Clustering and Association

- In a clustering problem, we want to discover the inherent groupings in the data:
  - Eg: grouping customers by purchasing behaviour.
- In an association rule learning problem, we want to discover rules that describe large portions of your data
  - E.g. people that buy $\boldsymbol{X}$ also tend to buy $\boldsymbol{Y}$

Unsupervised Learning Algorithms

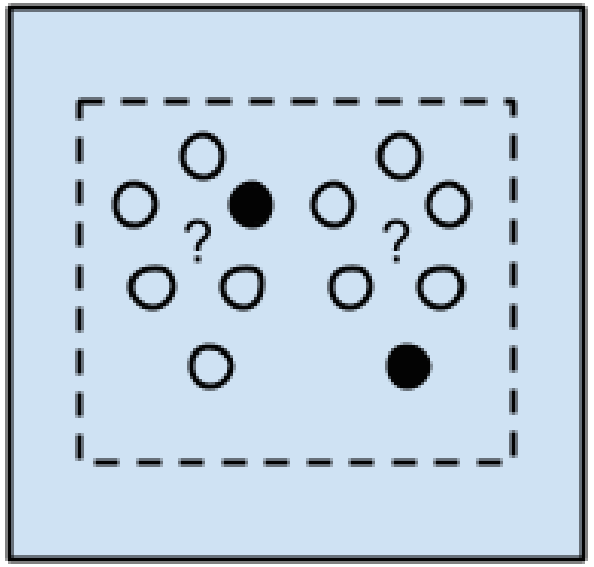
- Input data in not labelled
  - Output not known
- In training
  - Deduces structures present in the input data
  - Extracting general rules, reducing redundancy or organise data by similarity
- Problem types: clustering, dimensionality reduction and association rule learning
- Algorithms:
  - K-Means algorithm
  - Apriori algorithm.



Unsupervised Learning Algorithms

Semi-supervised Learning

- Semi-supervised learning approach refers to:
  - when we have a large amount of input data (X) but **only some** of the data is labelled (Y)
  - e.g. a photo archive where only some of the images are labelled, (e.g. *dog, cat, person*) and the majority are unlabelled.
- Many real world problems adopt this method
  - It can be expensive or time-consuming to label data
  - A hybrid design often helps to bridge the gaps
- Algorithms:
  - A flexible combination of supervised and unsupervised algorithms

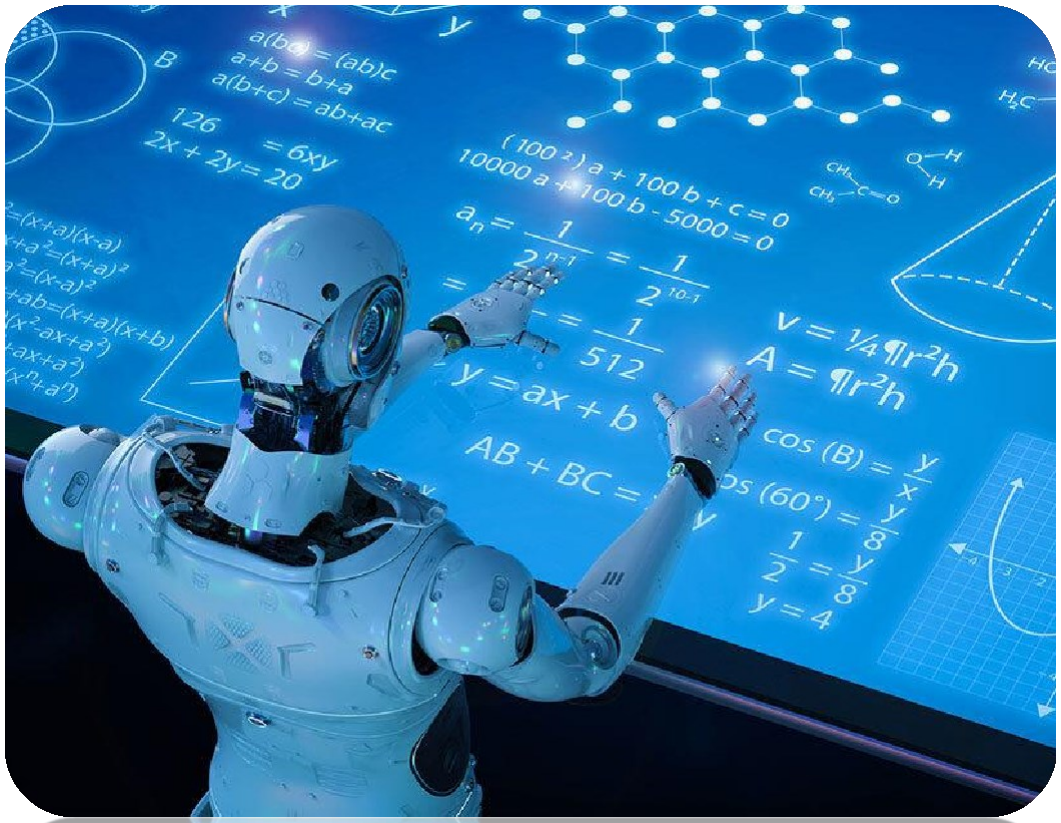


Semi-supervised Learning Algorithms

Machine Learning Summary

Today's Lecture

- Overview of Machine Learning
  - AI and ML, Definitions of ML, How to learn
- Types of Machine Learning
  - Supervised, unsupervised, semi-supervised
- Supervised Learning
  - Classification and regression
- Unsupervised Learning
  - Clustering and association



Coming up…

Lab: Introduction to Matlab

- Materials on Moodle

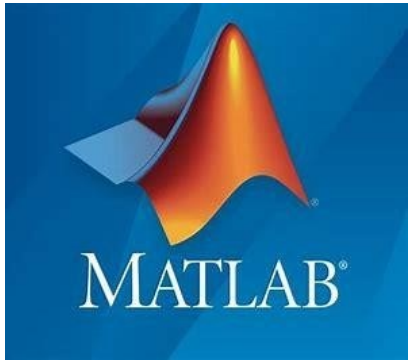
Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076

Coming up…

Lab: Introduction to Matlab

- Materials on Moodle

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076



Week 2 Lectures: Features in Machine Learning and Feature Extraction

- What are features?
- How to extract/represent features from text, images

Questions?

Lancaster University

SCC361: Artificial Intelligence

Week 2: Features and Feature Extraction

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

Attendance Check-in

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the “You are checked in” confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Lecture Plan

Weeks 1-5

1. Introduction to Artificial Intelligence and Machine Learning
2. **Features in Machine Learning and Feature Extraction**
3. Computer Vision and Natura Language Processing
4. Clustering and Classification
5. Artificial Neural Networks

Weeks 6-10:

6. Genetic Algorithms
7. Naïve Bayesian Classifier
8. Decision Tree Classifier
9. Introduction to Deep Neural Networks
10. Introduction to Convolutional Neural Networks

SCC361: Week 1

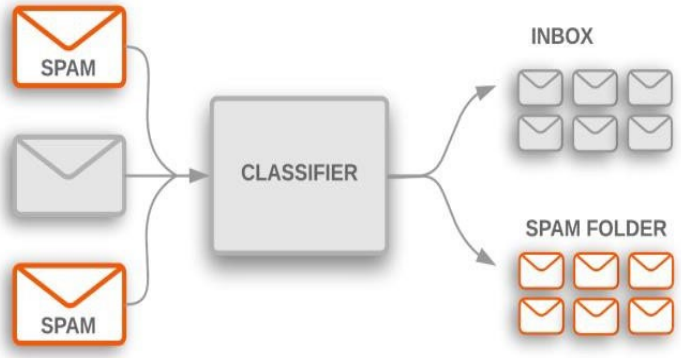
Artificial Intelligence

- AI Overview
- Definition of AI
- Goals of AI
- AI and Society

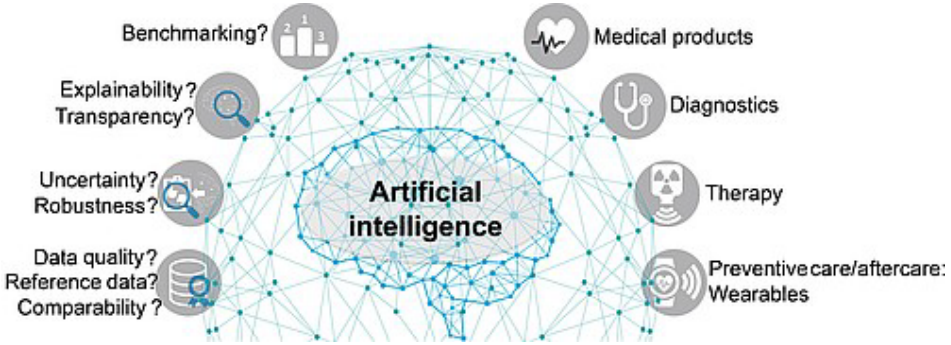
Introduction to Machine Learning

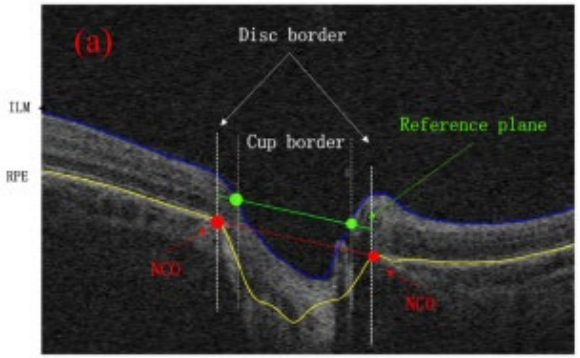
- Machine Learning Overview
- Types of Machine Learning
- Supervised Learning
- Unsupervised Learning

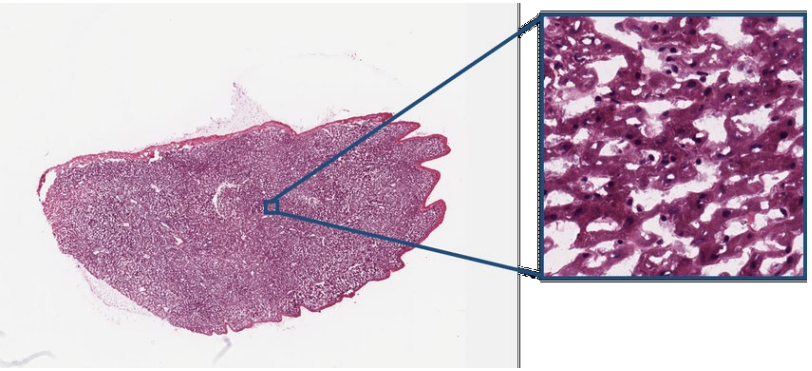
AI in Real Life

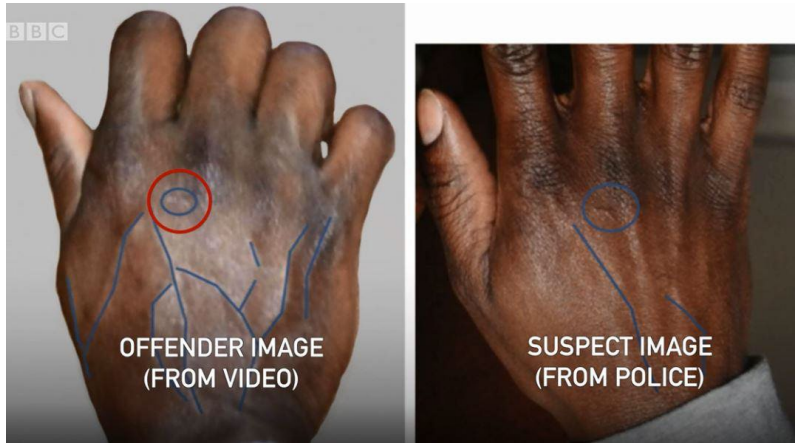


AI in Medicine and Security









AI in Forensic Identification

Definition of AI

A **computational agent** is an **agent** (something that **acts** within an environtment) whose decisions and actions can be explained in terms of computation.

Artificial intelligence, or **AI** is the field that studies the *synthesis and analysis of* ***computational agents*** *that* ***act rationally***

Artificial Intelligence: A Modern Approach, 2016, Russell & Norvig
Artificial Intelligence: Foundations of Computational Agents, 2017, Poole & Markworth

Goals of AI

Scientific Goals

- Understand the principles of intelligent behaviour:
  - Analysis of natural and artificial agents
  - Formulating and testing hypothesis
  - Designing, building and experimenting with computational agents
- Uses a general scientific approach
- Focuses on **empirical systems** and not **application**

VS

Engineering Goals

- Constructing intelligent agents
- Focuses on the **design** and **synthesis** of useful, intelligent artefacts.
- Builds agents that act intelligently
- Agents that are useful in many real-world applications

Benefits of AI

Business Benefits

Workflow/Process automation

- Bots for routine, repetitive tasks

Enhance creative tasks

- More time and tools to explore creative functions

Increased accuracy

- Human errors can be reduced

Better predictions & improved decisions

- Predictions of risks, performance targets, tailored product offerings etc

Social Benefits

Healthcare

Smart cities, transportation, security

- Maps, navigation systems, unmanned vehicles, route planning, security

Forecasts and predictions

- Weather, natural disasters, economics

Agriculture

- Real-time data analytics help farmers to maximise crop yields and profits

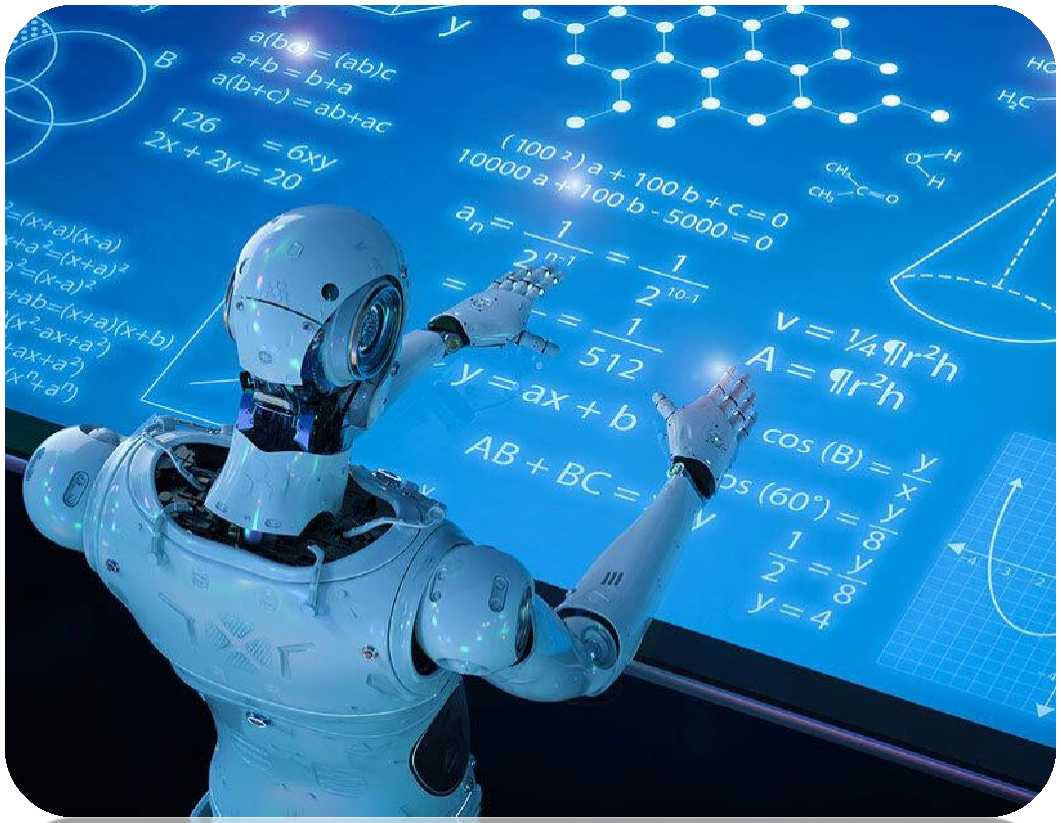
Overall lifestyle

Risks, Challenges and Ethical Concerns of AI

- Safety and security
- Trust and social manipulation
- Explainability of AI
- Possible job losses
- Accountability
- Accuracy, bias, privacy and inequality
- Human biases and prejudices
- Technological social responsibility (TSR)

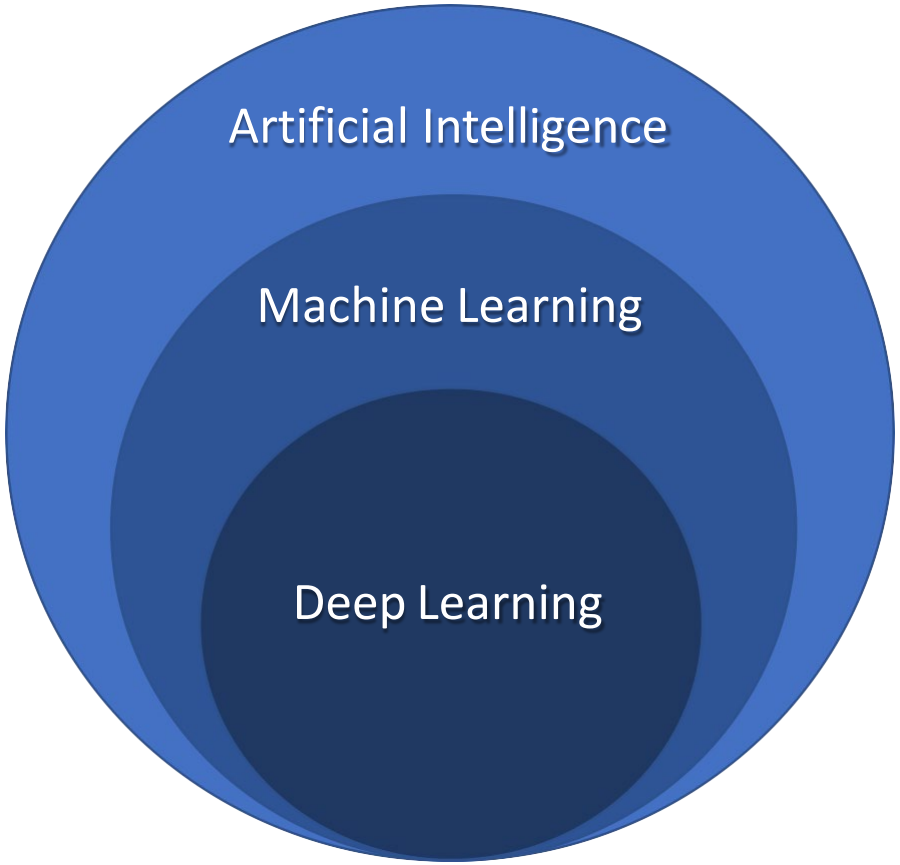
Introduction to Machine Learning

- Overview of Machine Learning
  - AI and ML, Definitions of ML, How to learn
- Types of Machine Learning
  - Supervised, unsupervised, semi supervised
- Supervised Learning
  - Classification and regression
- Unsupervised Learning
  - Clustering and association

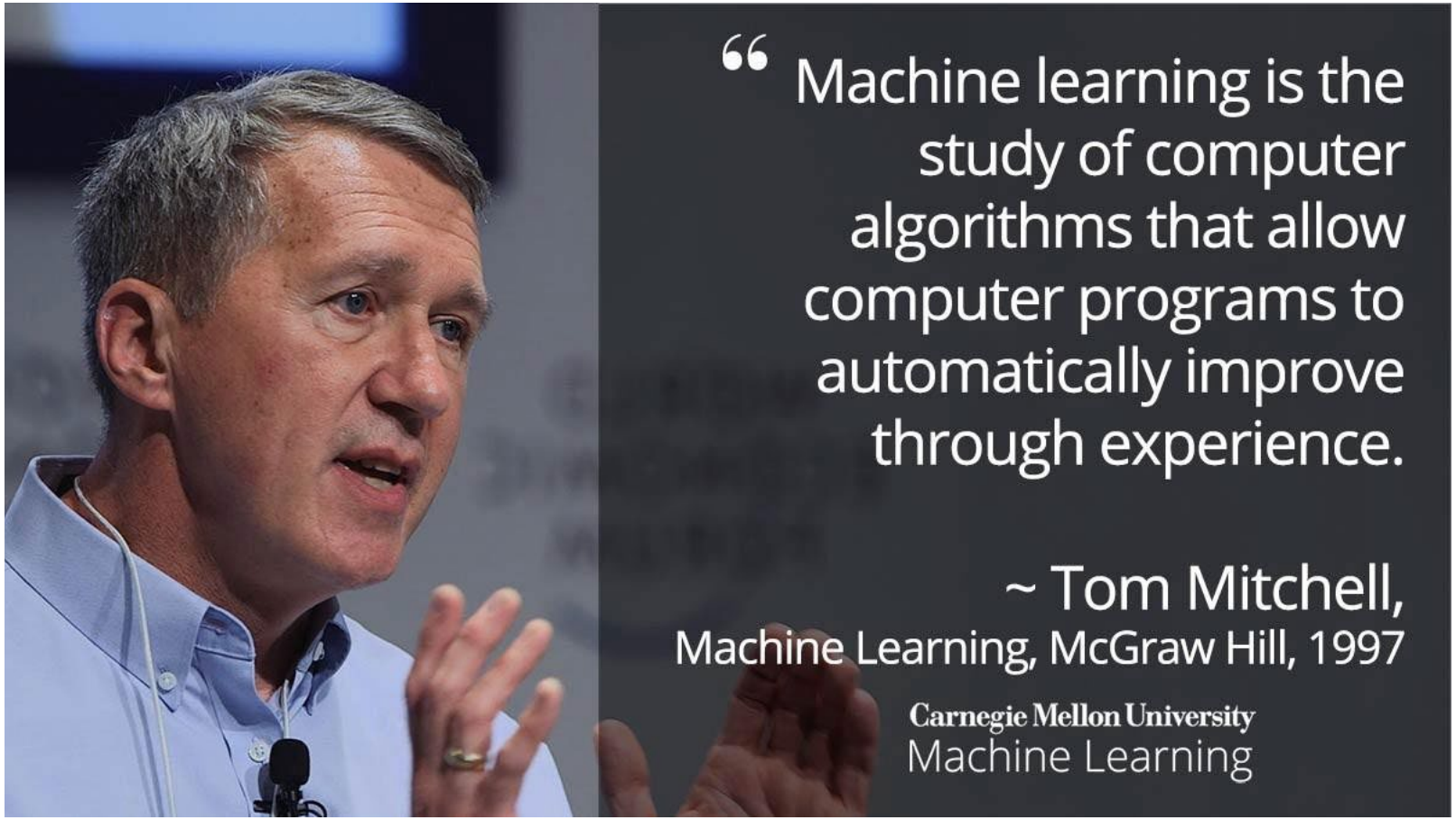


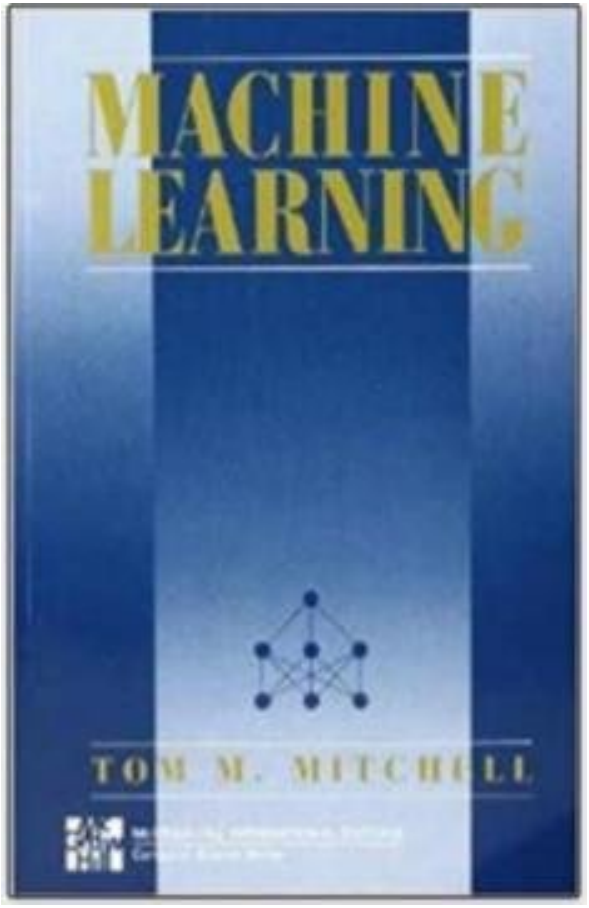
AI and Machine Learning

- AI systems were mostly **rule-based**
  - i.e. depended on hand-crafted rules
- Machine learning drives AI
  - Learning algorithms create a logical mapping from data to output
- Deep learning:
  - a subset of ML with additional layers to learn deeper representations data



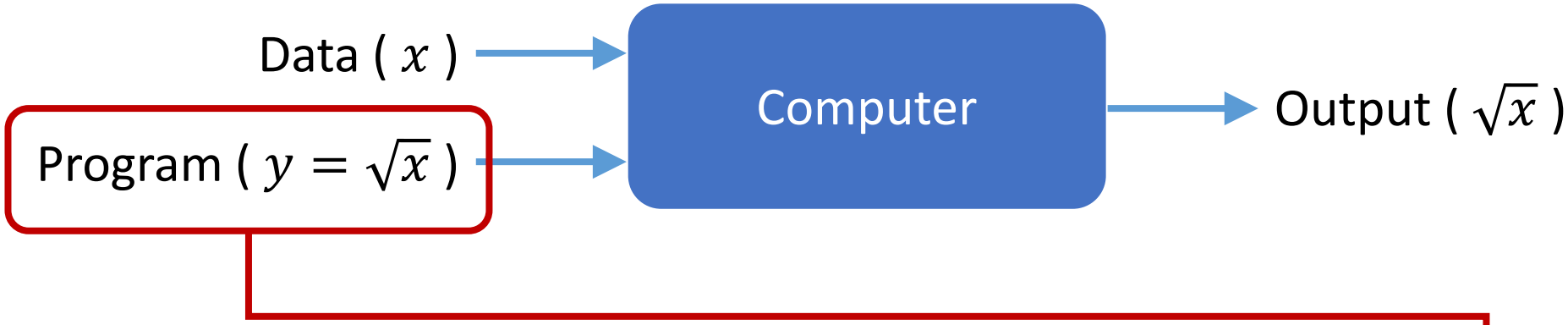
What is Machine Learning?





What is Machine Learning?

- Consider the function $y = f(x)$ (e.g. $f(x) = \sqrt{x}$)
- Traditional Programming (Software 1.0)

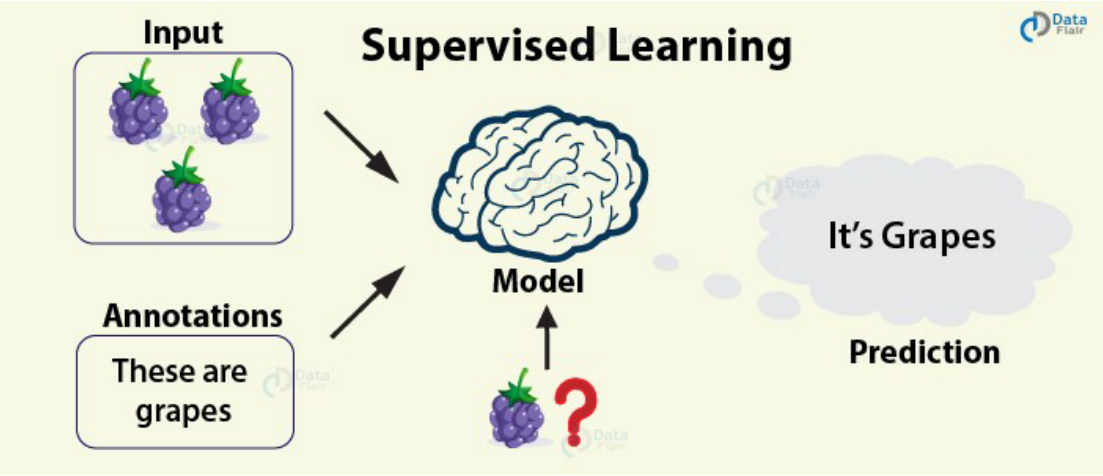


- Machine Learning (Software 2.0)

Types of Machine Learning

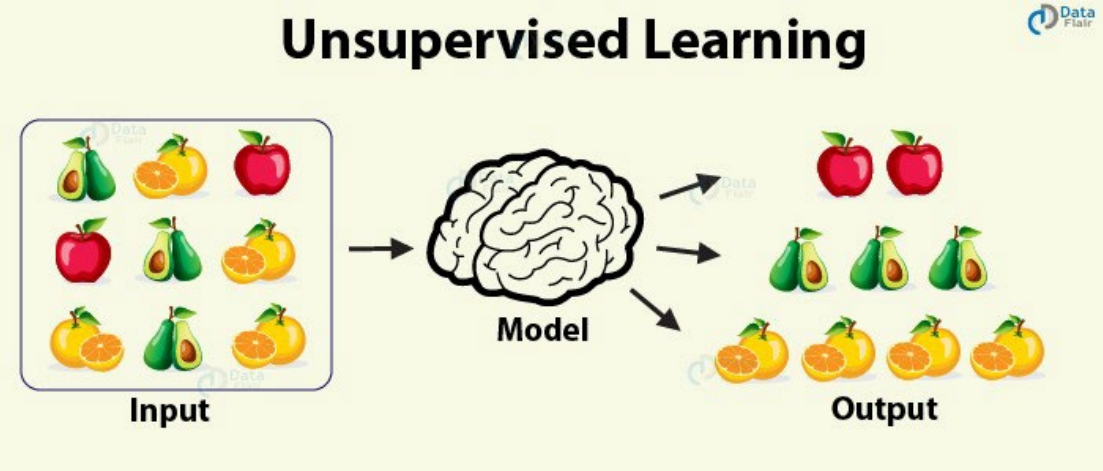
Supervised Learning

- Classification
- Regression



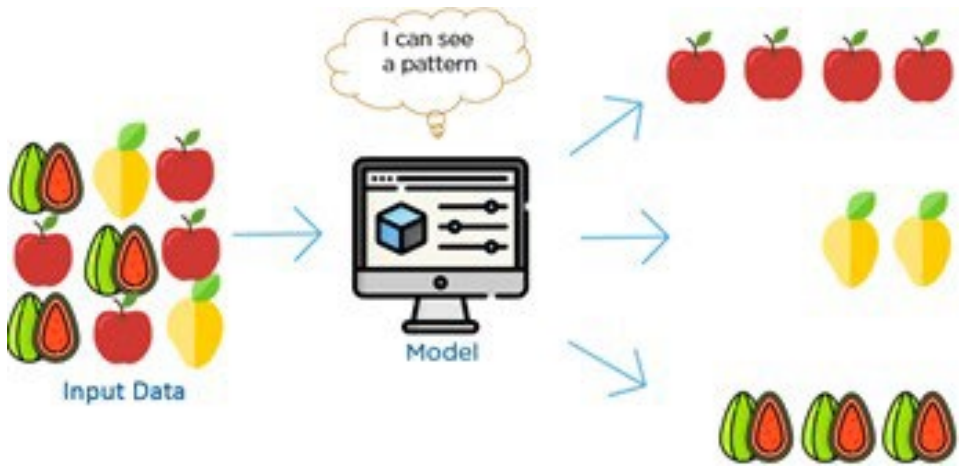
Unsupervised Learning

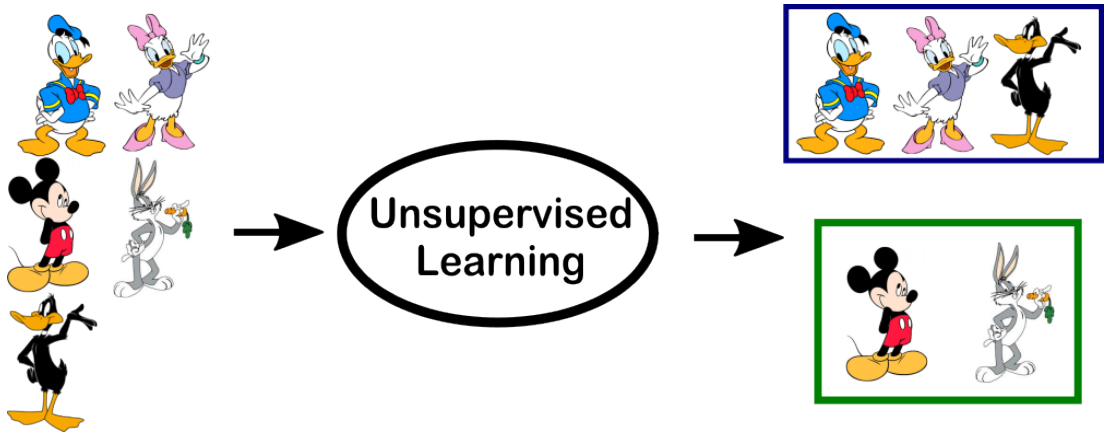
- Clustering
- Association



Unsupervised Learning

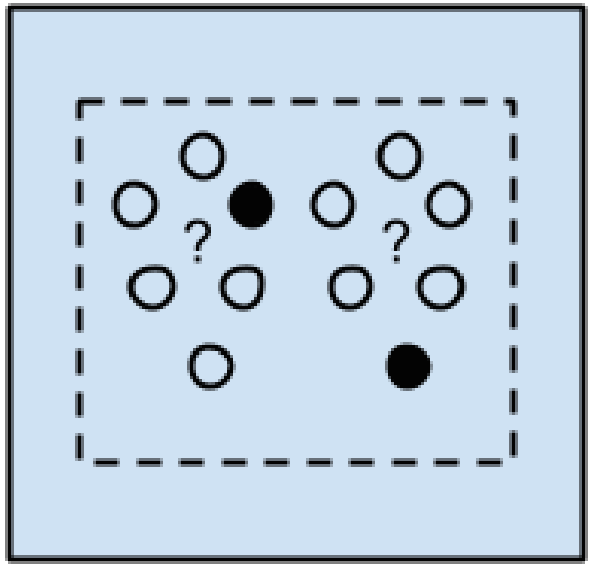
- Remember the function $y = f(x)$
- With unsupervised learning, only the input data, x, is available
- There are no corresponding labels (classes or categories) i.e. no output variable, y
- Aims at modelling the underlying structure of the data
- Two main categories
  - Clustering
  - Association





Semi-supervised Learning

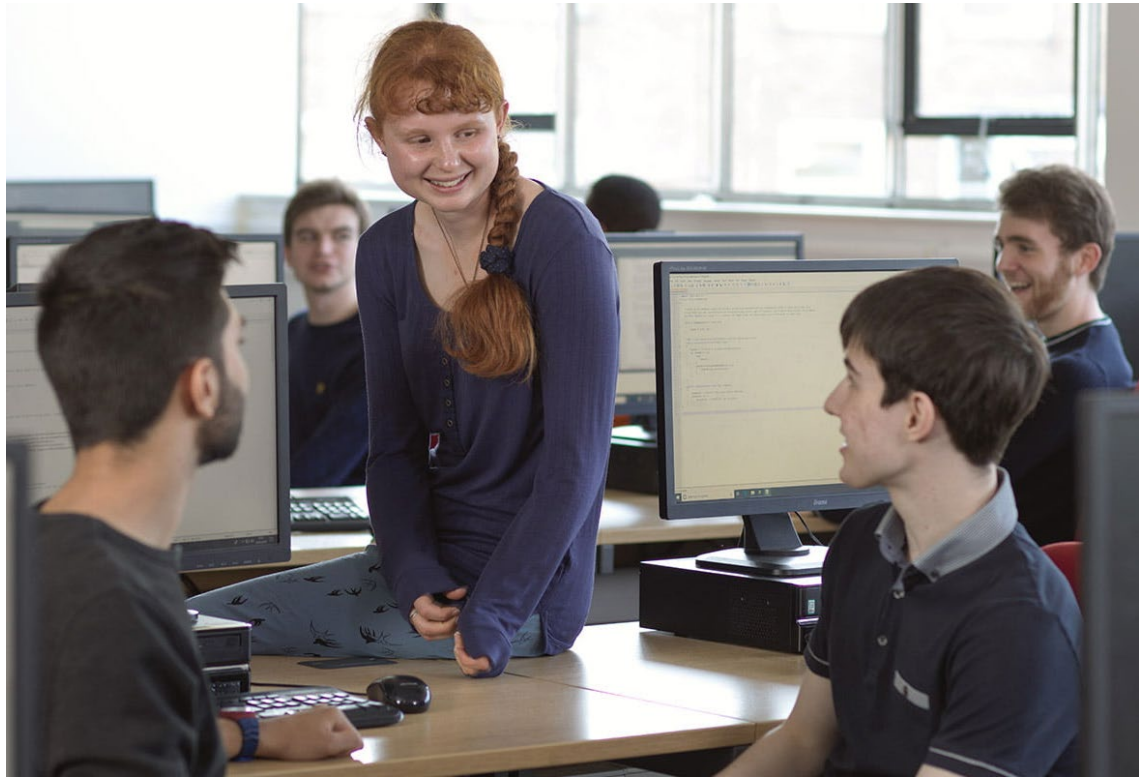
- Semi-supervised learning approach refers to:
  - when we have a large amount of input data (X) but **only some** of the data is labelled (Y)
  - e.g. a photo archive where only some of the images are labelled, (e.g. *dog, cat, person*) and the majority are unlabelled.
- Many real world problems adopt this method
  - It can be expensive or time-consuming to label data
  - A hybrid design often helps to bridge the gaps
- Algorithms:
  - A flexible combination of supervised and unsupervised algorithms



Semi-supervised Learning Algorithms

Lab Sessions and Assignment

- Common AI and CV tasks
  - Smoothing
  - Pooling
  - Blurring
  - Edge Detection
- Good attendance and good questions
- Solutions available on Moodle
  - .m / .mlx file
  - Video



This Week

Features in Machine Learning

Feature Extraction in Images and Text

SCC361: Artificial Intelligence

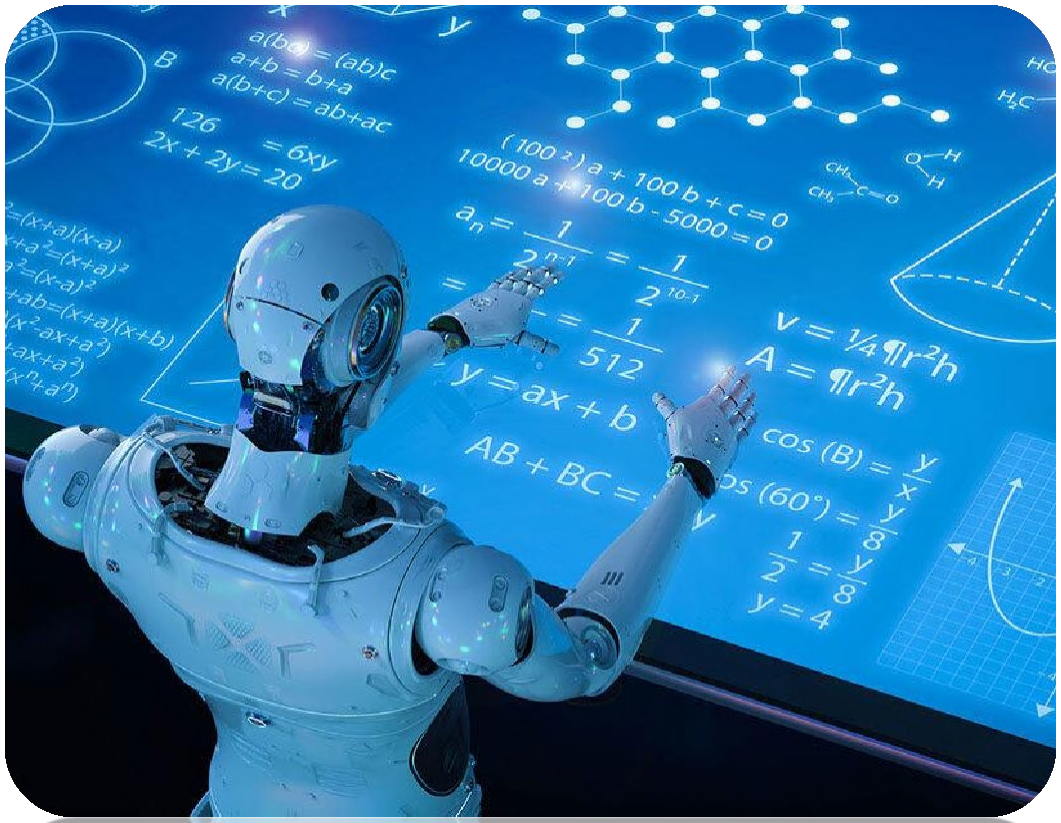
Features in Machine Learning

Feature Extraction in Images and Text

Today's Lecture

Features in Machine Learning

- Machine learning models are dependent on data
- Machine learning algorithms are designed to understand numbers
- How can real-world data be interpreted as numbers?
  - E.g. Images, video text.



Expected Learning Outcomes

By the end of this week:

- Understand what features and labels are
- Define feature extraction
- Understand the distinction between manual and automatic approaches to feature extraction
- Understand processes of feature extraction in images and text

Features and Labels

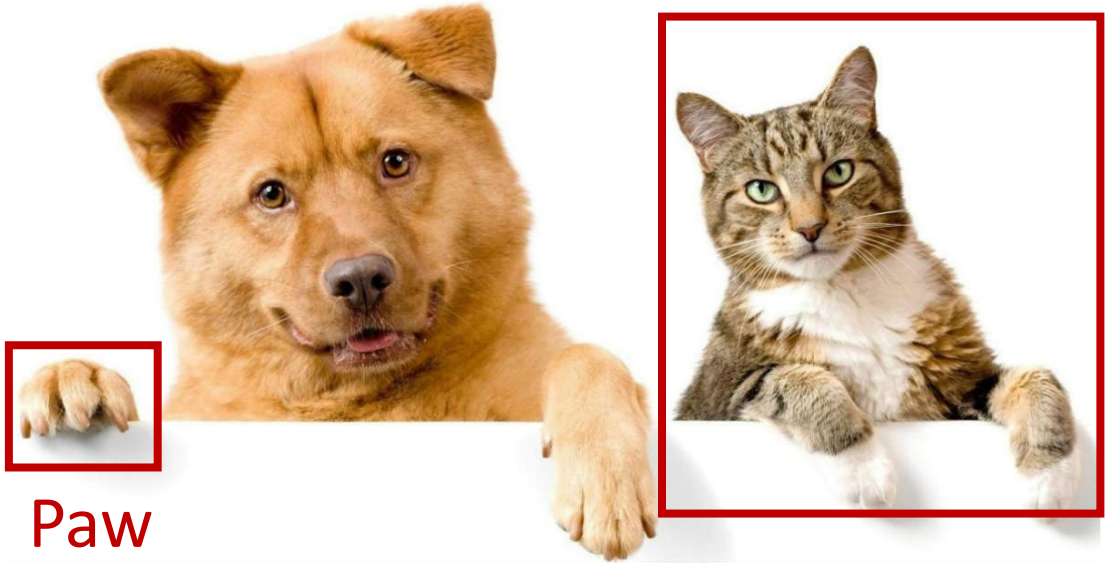
Feature:

A **feature** is an individual measurable property or characteristic of a phenomenon being observed

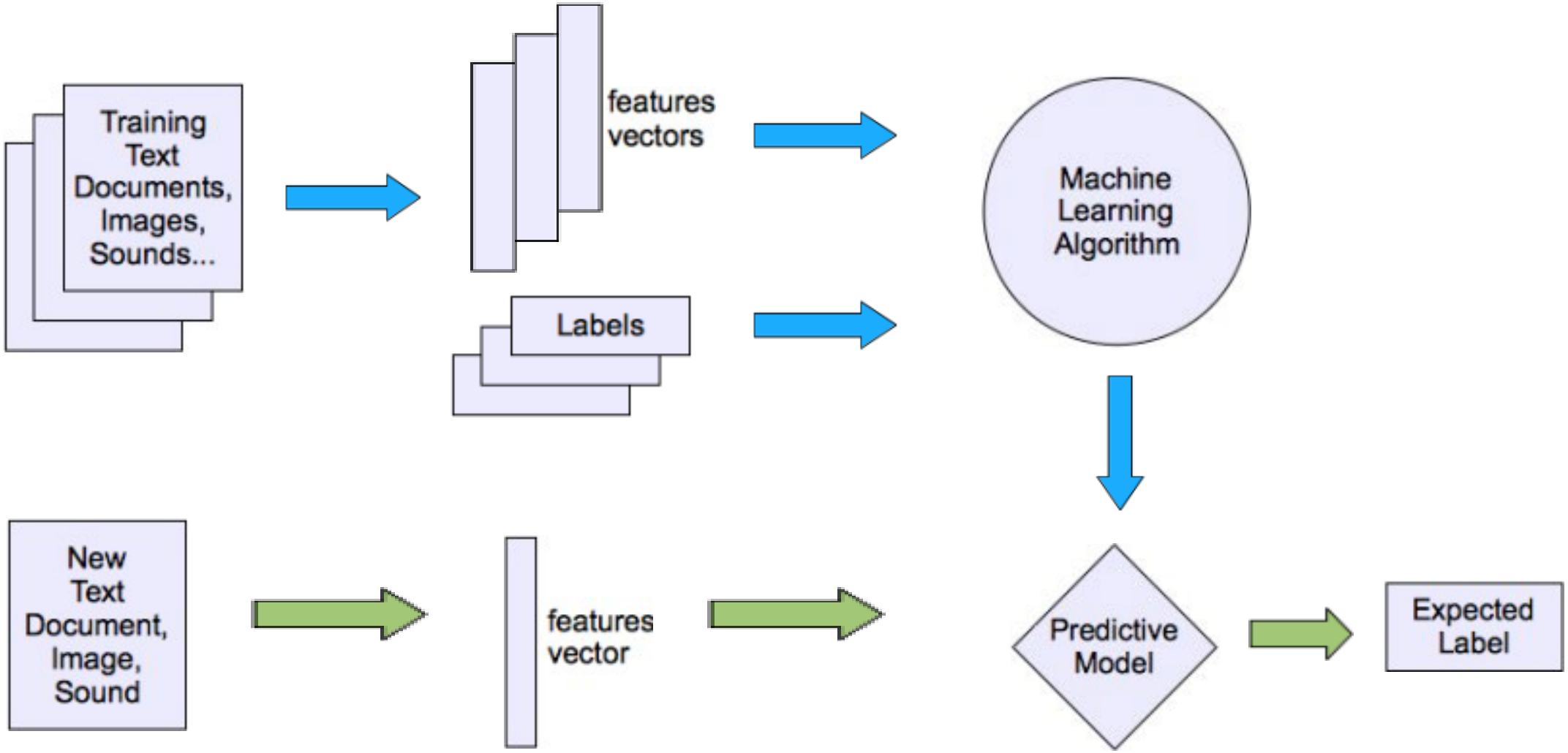
A set of features represent the information you can draw (extract) from data

Label:

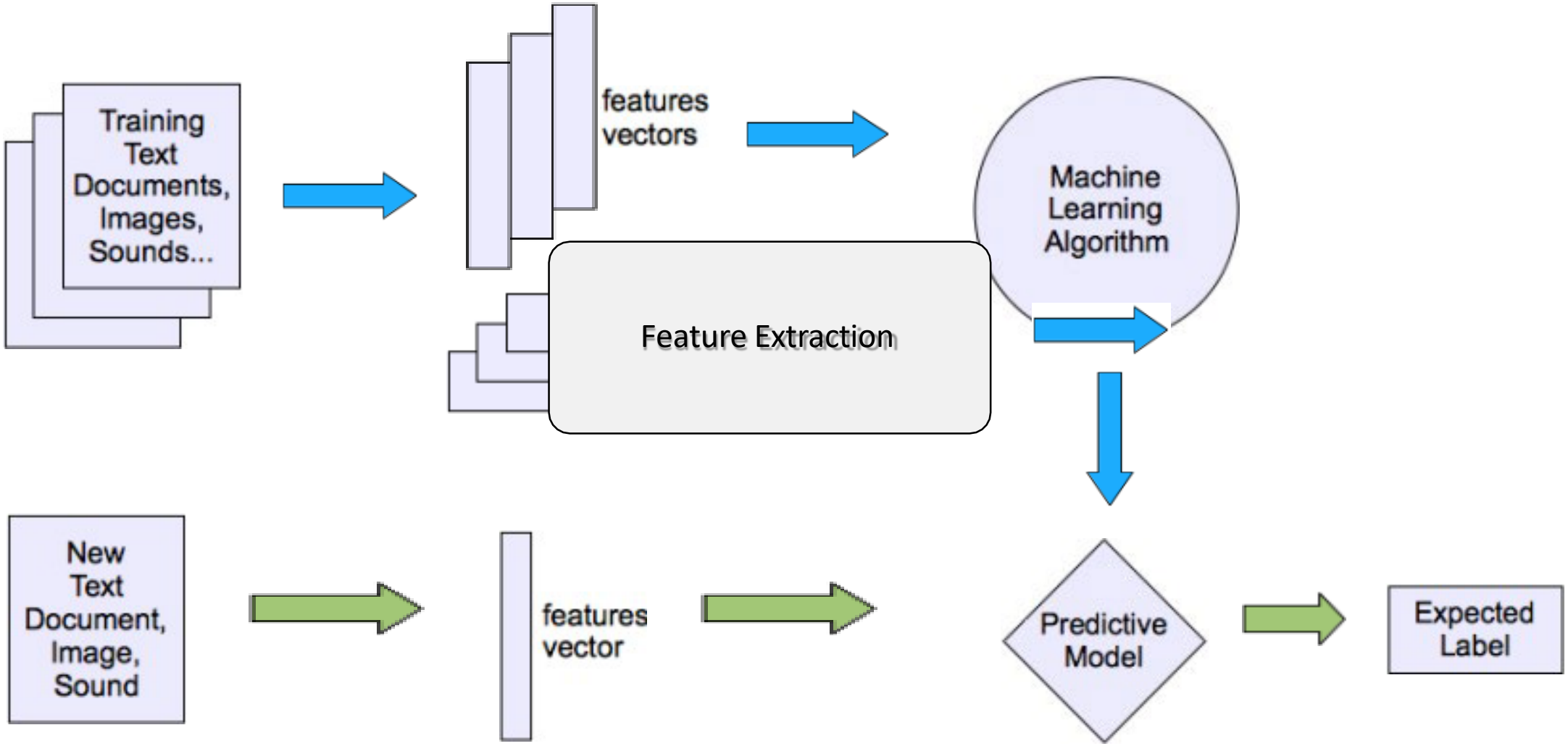
The label is the tag you wish to assign to a set of features



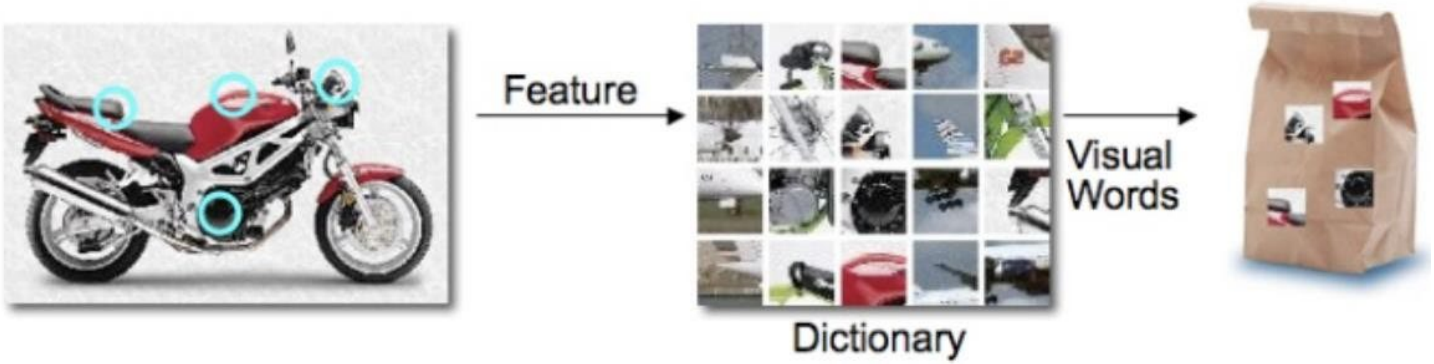
Features for ML



Features for ML



Feature Extraction



- A set of methods that map input features to new output features
- Any technique that transform raw data into features that can be used as input to a learning algorithm

Feature Extraction

- A set of methods that map input features to new output features
- Any technique that transform raw data into features that can be used as input to a learning algorithm
- The process of transforming **raw data** into **numerical features** that can be processed while **preserves** the information in the original data set.

Why Feature Extraction?

- Learning algorithms (and computers) prefer data in numeric format

Why Feature Extraction?

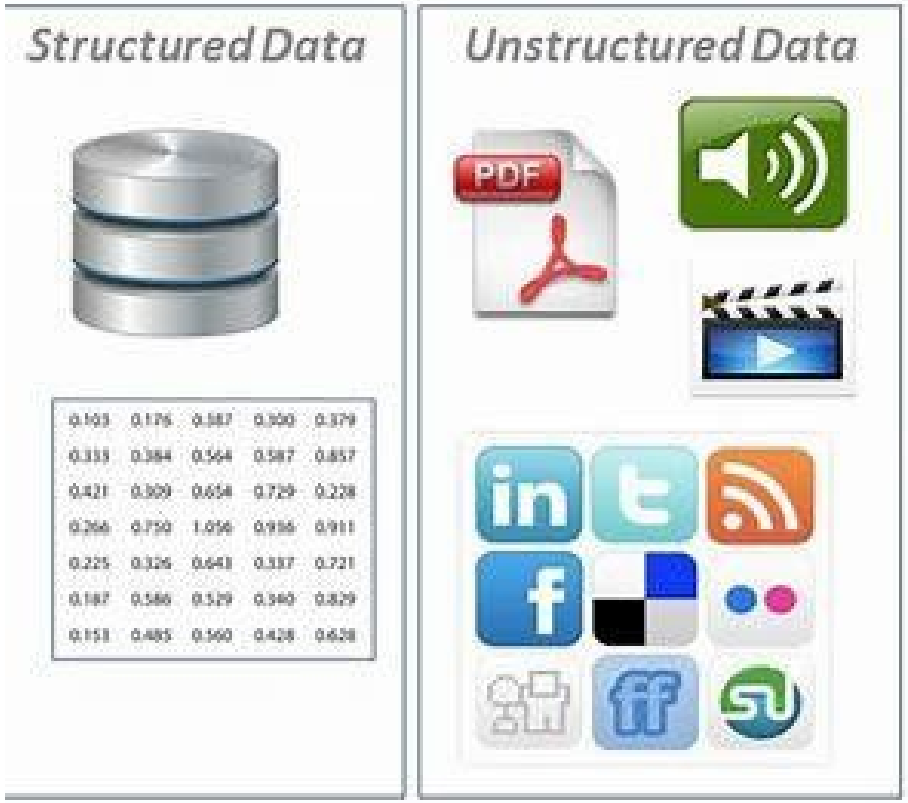
- Learning algorithms (and computers) prefer data in numeric format

Why Feature Extraction?

- Learning algorithms (and computers) prefer data in numeric format
- Real-word data are often generated in non-numeric format:
  - text, image, audio, video etc

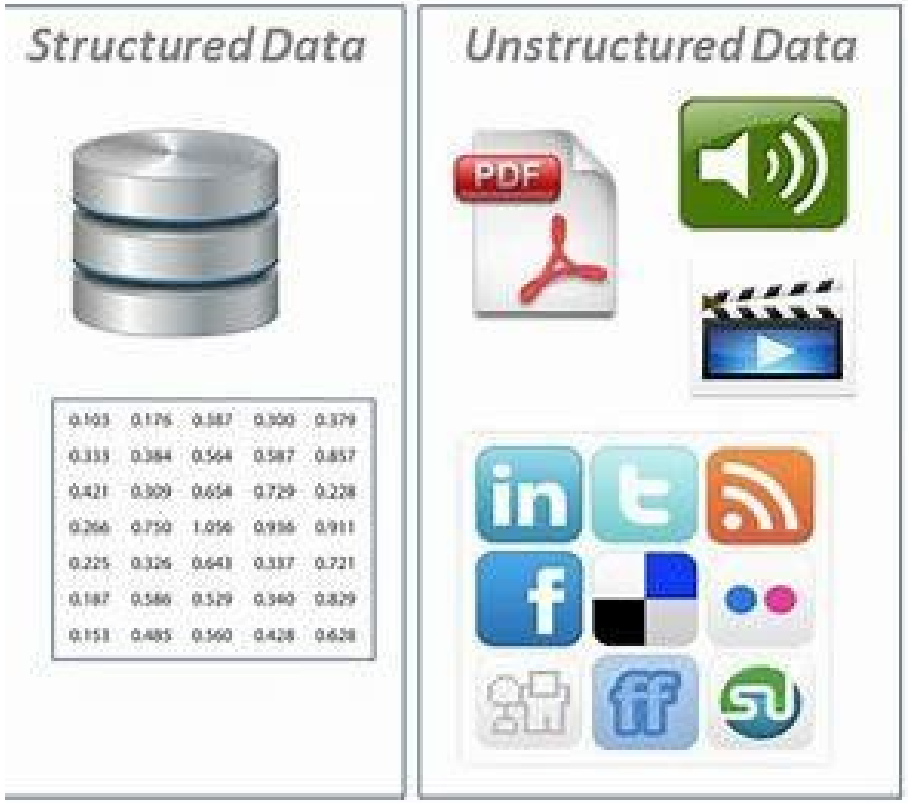
Why Feature Extraction?

- Learning algorithms (and computers) prefer data in numeric format
- Real-word data are often generated in non-numeric format:
  - text, image, audio, video etc
- Real-world data are mostly unstructured



Why Feature Extraction?

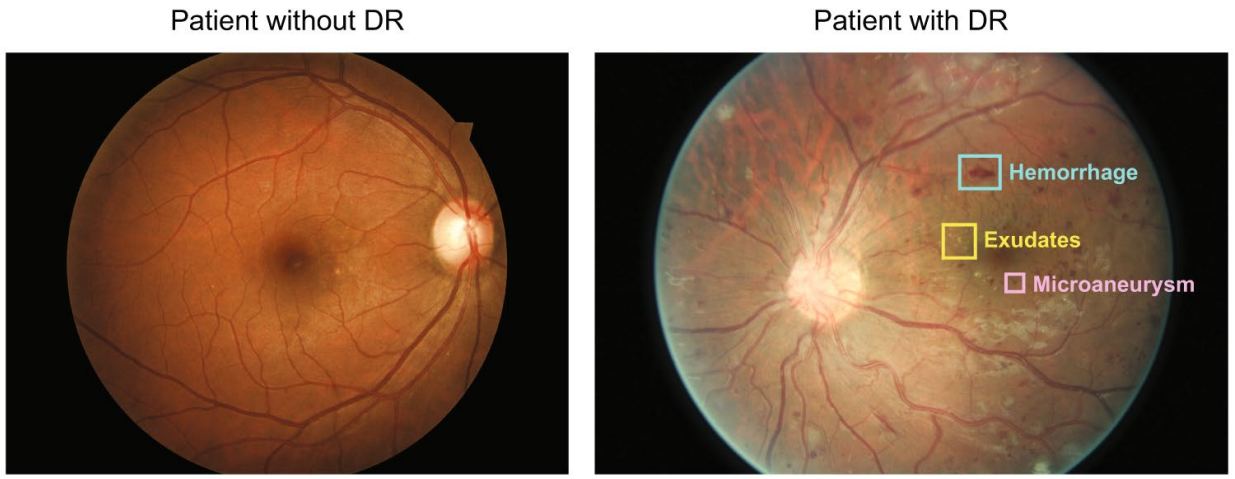
- Learning algorithms (and computers) prefer data in numeric format
- Real-word data are often generated in non-numeric format:
  - text, image, audio, video etc
- Real-world data are mostly unstructured
- Increases learning accuracy by extracting the most significant features

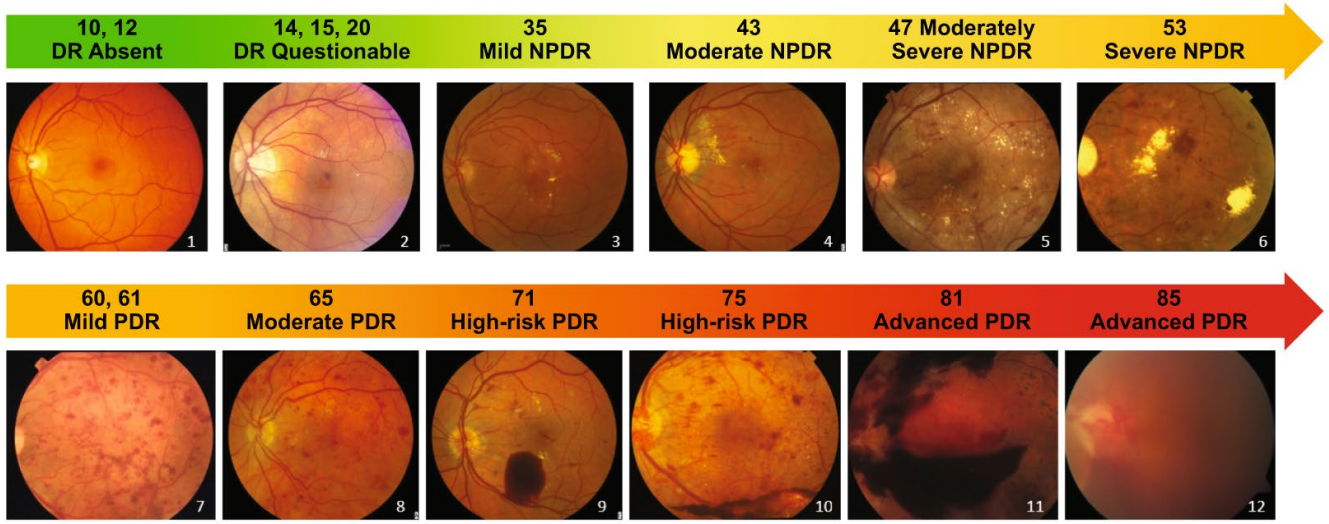


Examples of Feature Extraction

What are the features and labels?

- Image type?
- Patient Diagnosis?
- Image Diagnosis?
- Pathology?
- Anatomy?

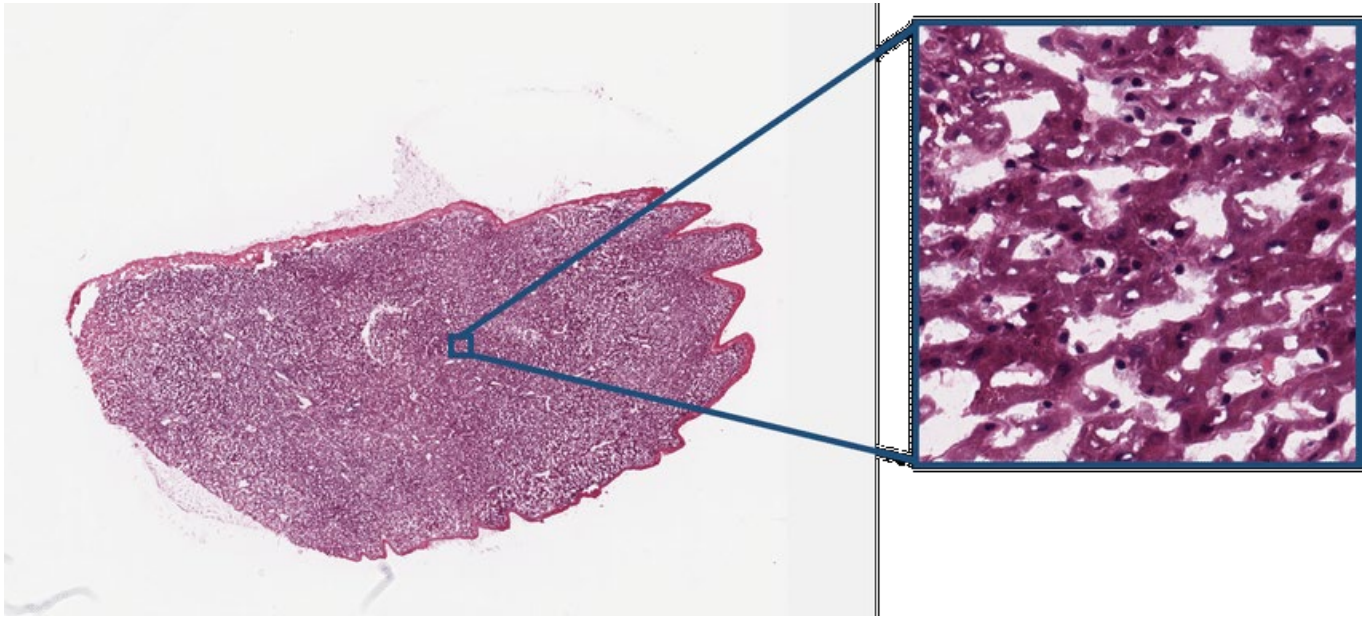




Examples of Feature Extraction

What are the features and labels?

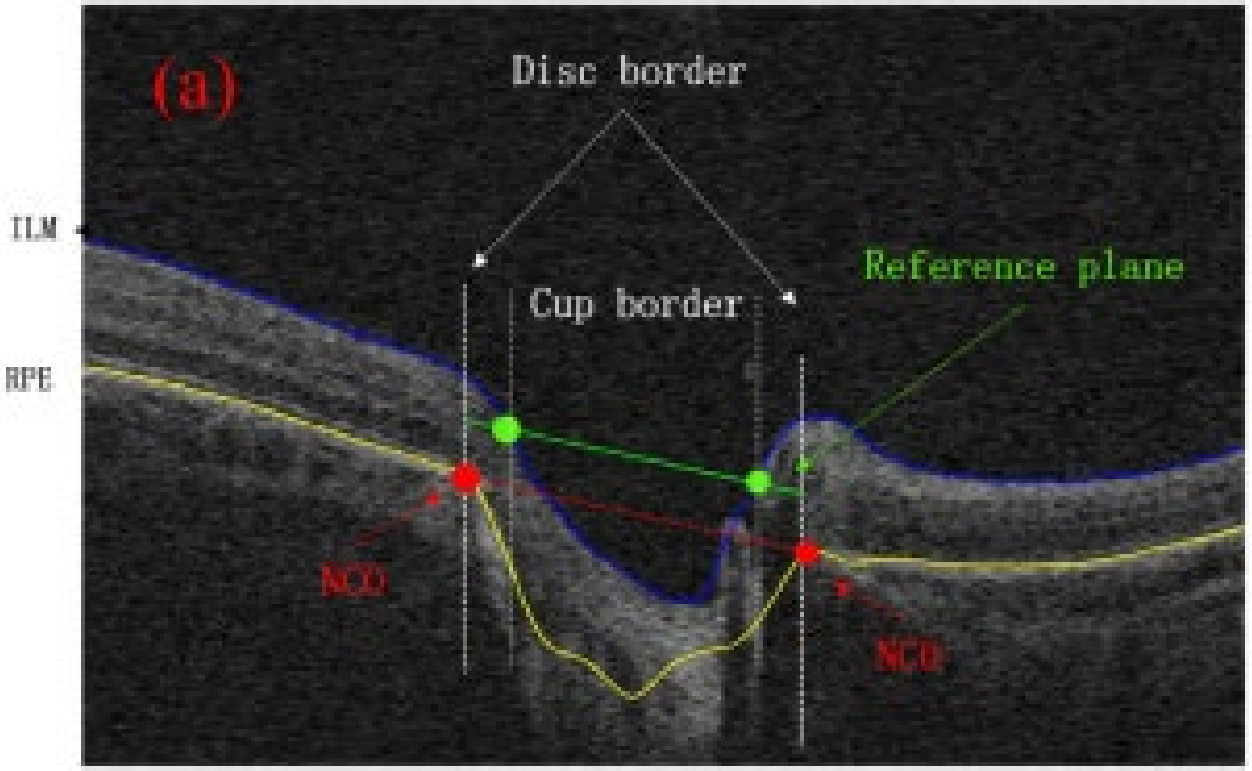
- Diagnosis?
- Region?



Examples of Feature Extraction

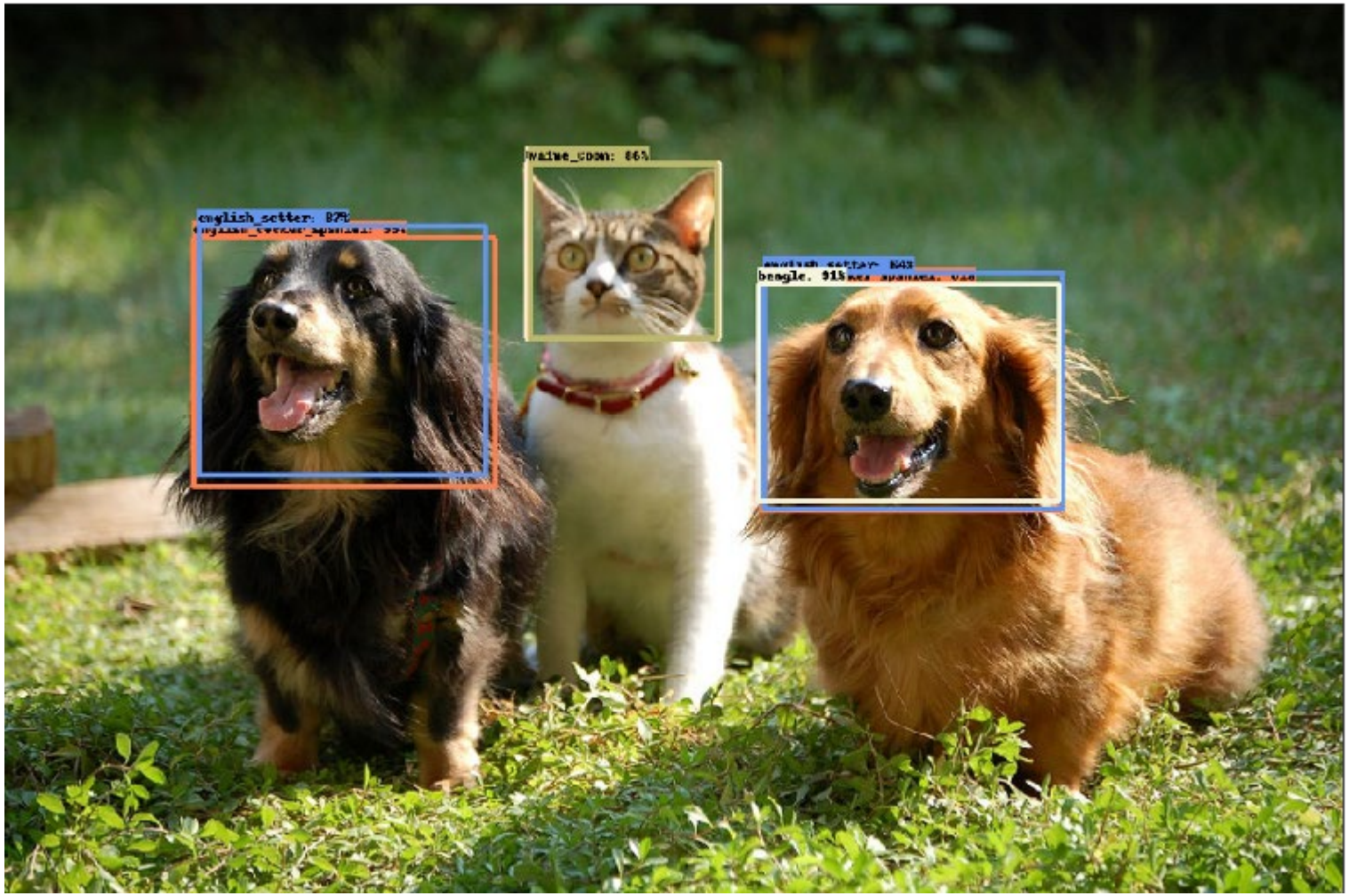
What type of label do you need?

- Dimension?
- Data type?
- Number of labels?



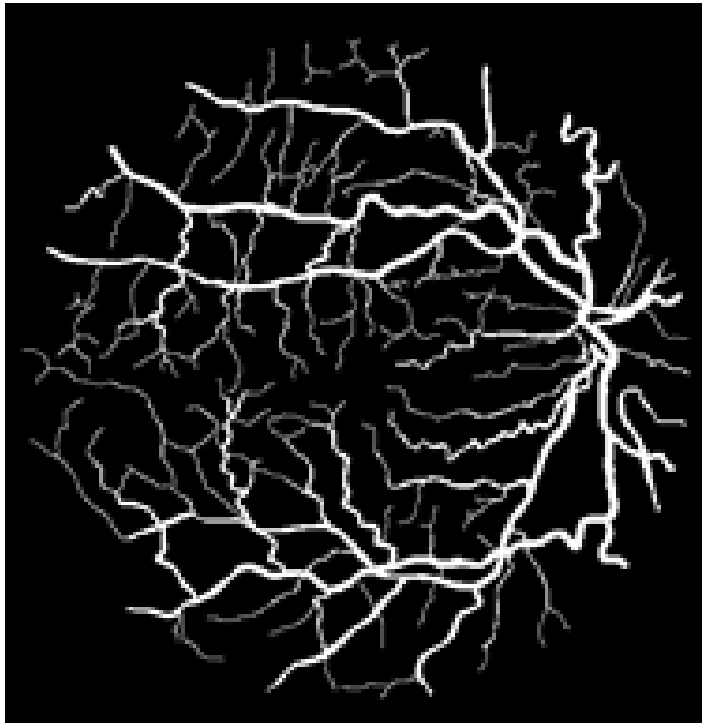
Examples of Feature Extraction

What is the label?

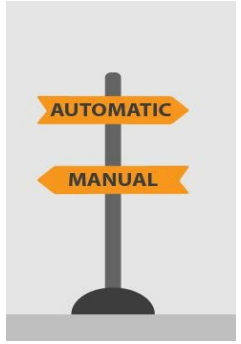


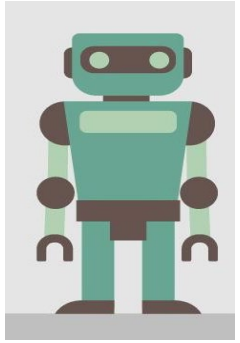
Examples of Feature Extraction

What is the label?



Approaches to Feature Extraction





Manual:

- Manually identifying and describing relevant features for a given problem and implementing a way to extract those features.
- Mostly requires experience or domain knowledge
- Eg: computing the mean of a window in signal processing

Automated:

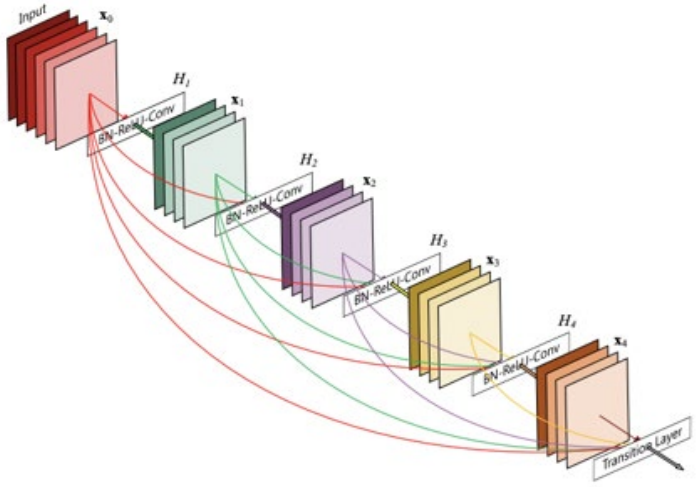
- uses specialized algorithms to extract features automatically from signals or images without the need for human intervention
- More efficient for raw data transformation
- Eg: Wavelet scattering

Feature Vector Example

How can we compare these images?

Look for meaningful attributes?

High-dimensional encoding?



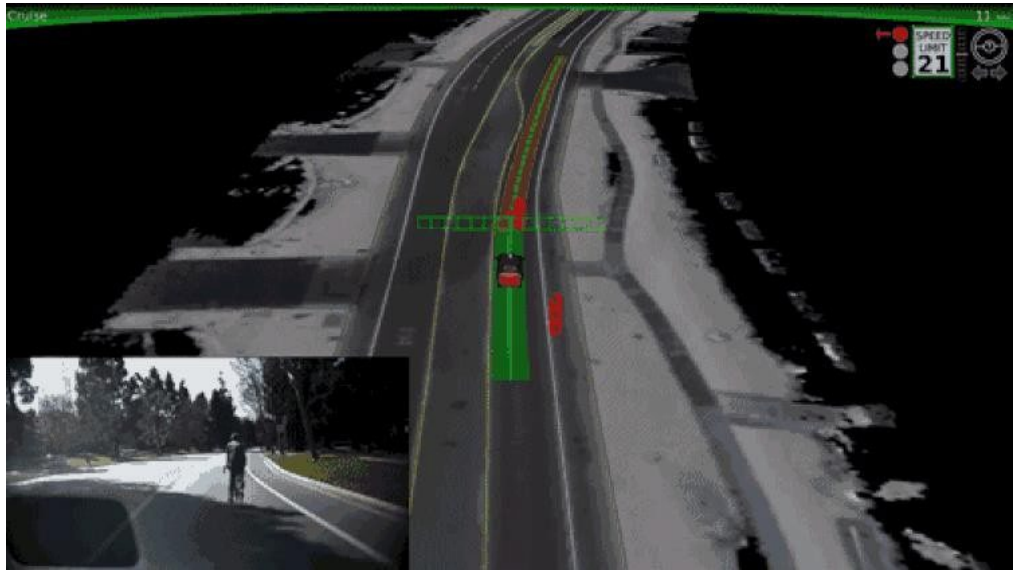
Definition

A **feature** is an individual measurable property or characteristic of a phenomenon.

A **feature vector** represents numeric or symbolic characteristics, called features, of an object in a mathematical analysable way

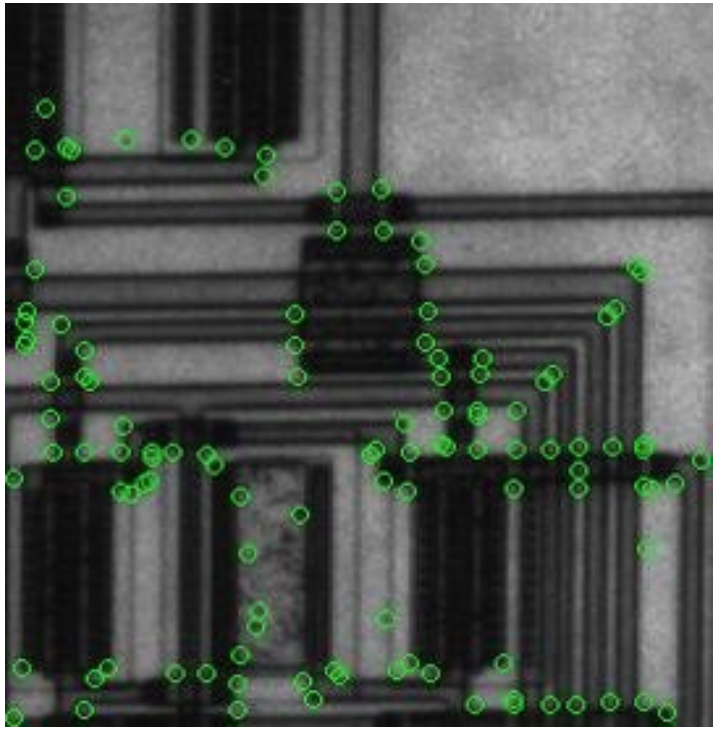
Feature Extraction from Images

- Detects and represents the interesting parts of an image as a compact feature vector
- A critical step in image processing and computer vision
- Transforms pictorial image to numeric data representation
- These representations can be used as inputs to classification algorithms



What counts as a “good” feature?

- Repeatable
  - Should be detectable at the same location in different images despite changes in viewpoint and illumination
- Saliency
  - Descriptive
  - Same points in different images should have similar features
- Compactness
  - Affect speed of matching
  - Fewer and smaller features are better



Steps to Image Feature Extraction

- There are two key steps:
- Feature detection
  - where to extract features from
  - Not all locations on the image are good
- Feature Extraction
  - Orientation of the extraction window
  - What to encode in the feature

How do we interpret an image

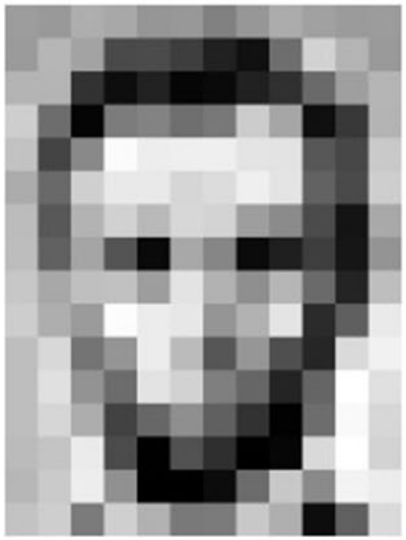
- An image is an array of pixels
- Pixel values can be extracted as feature vector representation for images
- It is important to know which **pixels** and which **interpretation** are relevant for the task

- 8-bit Image:

$$\boldsymbol{u} \in \{0,1,\dots,255\}^{m\times n\times p}$$

- Double Format:

$$\boldsymbol{u} \in [0,1]^{m\times n\times p}$$



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

How do we interpret an image

- An image is an array of pixels
- Pixel values can be extracted as feature vector representation for images
- It is important to know which **pixels** and which **interpretation** are relevant for the task

$$\boldsymbol{u} \in [0,1]^{682\times1024\times3}$$

$$\boldsymbol{u}^r(\Omega) = \begin{bmatrix} 0.55 & 0.72 & 0.44 \\ 0.23 & 0.56 & 0.23 \\ 0.67 & 0.52 & 0.93 \end{bmatrix}$$

$$\boldsymbol{u}^g(\Omega) = \begin{bmatrix} 0.45 & 0.23 & 0.76 \\ 0.76 & 0.34 & 0.76 \\ 0.94 & 0.98 & 0.93 \end{bmatrix}$$

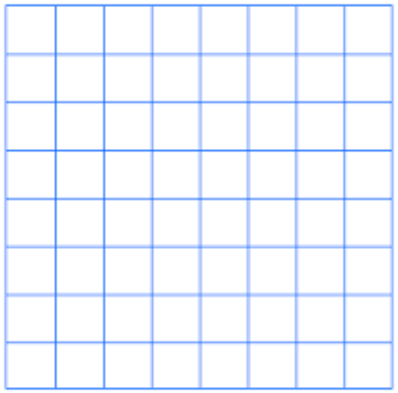
$$\boldsymbol{u}^b(\Omega) = \begin{bmatrix} 0.56 & 0.57 & 0.34 \\ 0.12 & 0.57 & 0.75 \\ 0.56 & 0.76 & 0.45 \end{bmatrix}$$

How do we interpret an image

Discrete Space vs Continuous Space

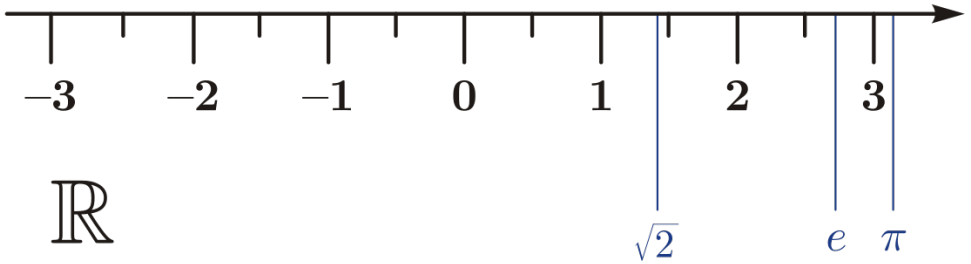
Discrete:

- Think of as a grid
- Limited Set of Values
- Practice



Continuous Space

- Think of the Real Line
- Infinite Set of Values
- Theory



How do we interpret an image

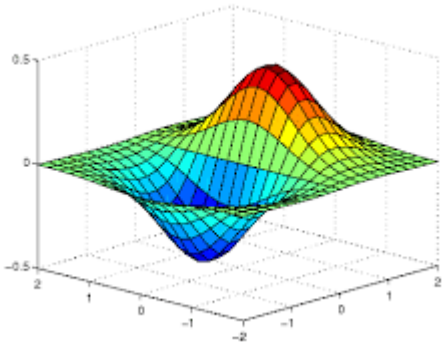
To develop computer vision theory:

We view an image as a function:

$$u: \underbrace{\mathbb{Z} \times \mathbb{Z}}_{pixel\ rows\ and\ columns} \rightarrow \underbrace{\mathbb{R}}_{intensity\ values}$$

We view an video as a function:

$$u: \underbrace{\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}}_{pixel\ rows\ and\ columns, frame} \rightarrow \underbrace{\mathbb{R}}_{intensity\ values}$$

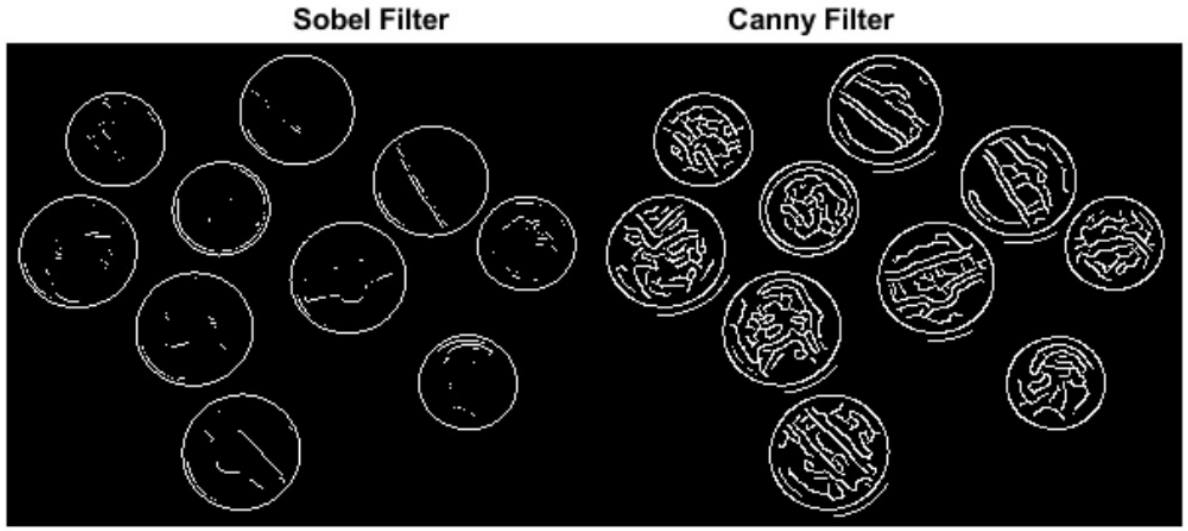


Try loading an image “im” in Matlab and typing

```
mesh(im(:,:,1))
```

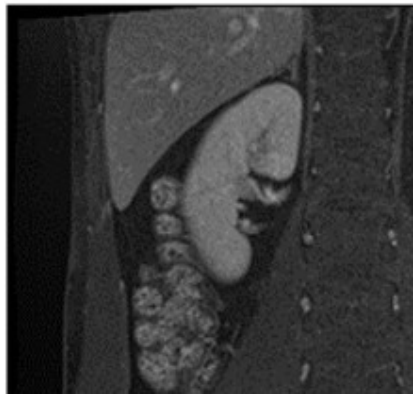
Edge Detection

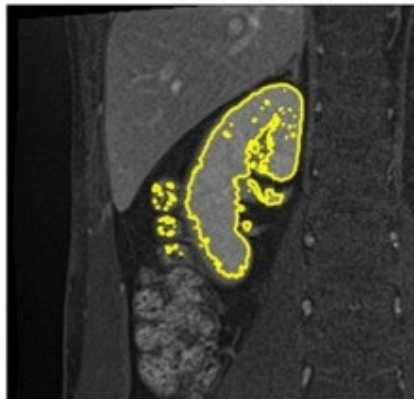
- an edge is a curve that follows a path of rapid change in image intensity
- Edge detection is used to identify the edges in an image
- The most well-known edge-detection method is Canny filter – based on gradients



Edge Detection

- an edge is a curve that follows a path of rapid change in image intensity
- Edge detection is used to identify the edges in an image
- The most well-known edge-detection method is Canny filter – based on gradients
- But more complex approaches do exist





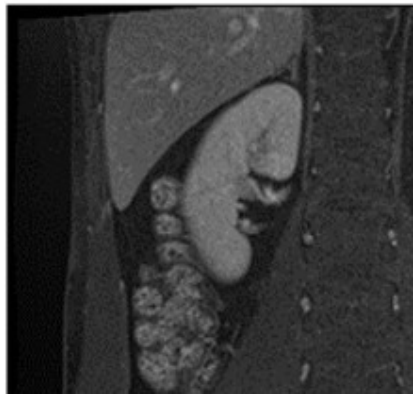
This is a research example for interest. You do **<u>not</u>** need to know this for this module:

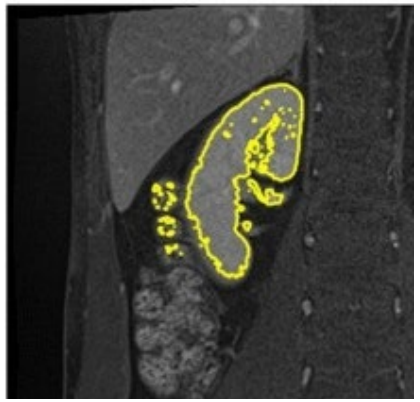
$$\min_{\Omega} \int_{\Omega} |f(x) - m(\Omega)|^2 \mathrm{d}x + \int_{\Omega^c} |f(x) - m(\Omega^c)|^2 \mathrm{d}x + \mathrm{Perim}(\Omega)$$

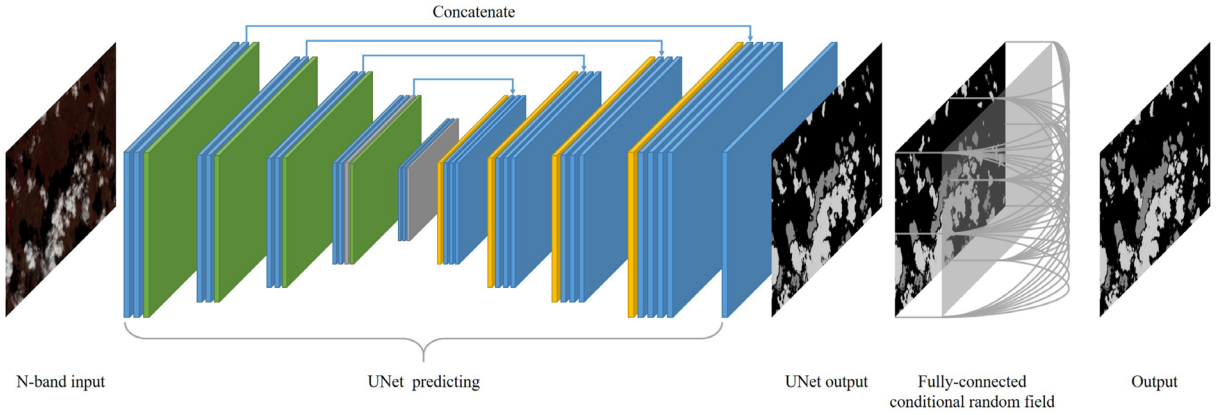
$$\text{where} \quad m(\Omega) \stackrel{\text{def.}}{=} \frac{1}{|\Omega|} \int_{\Omega} f(x) \mathrm{d}x$$

Edge Detection

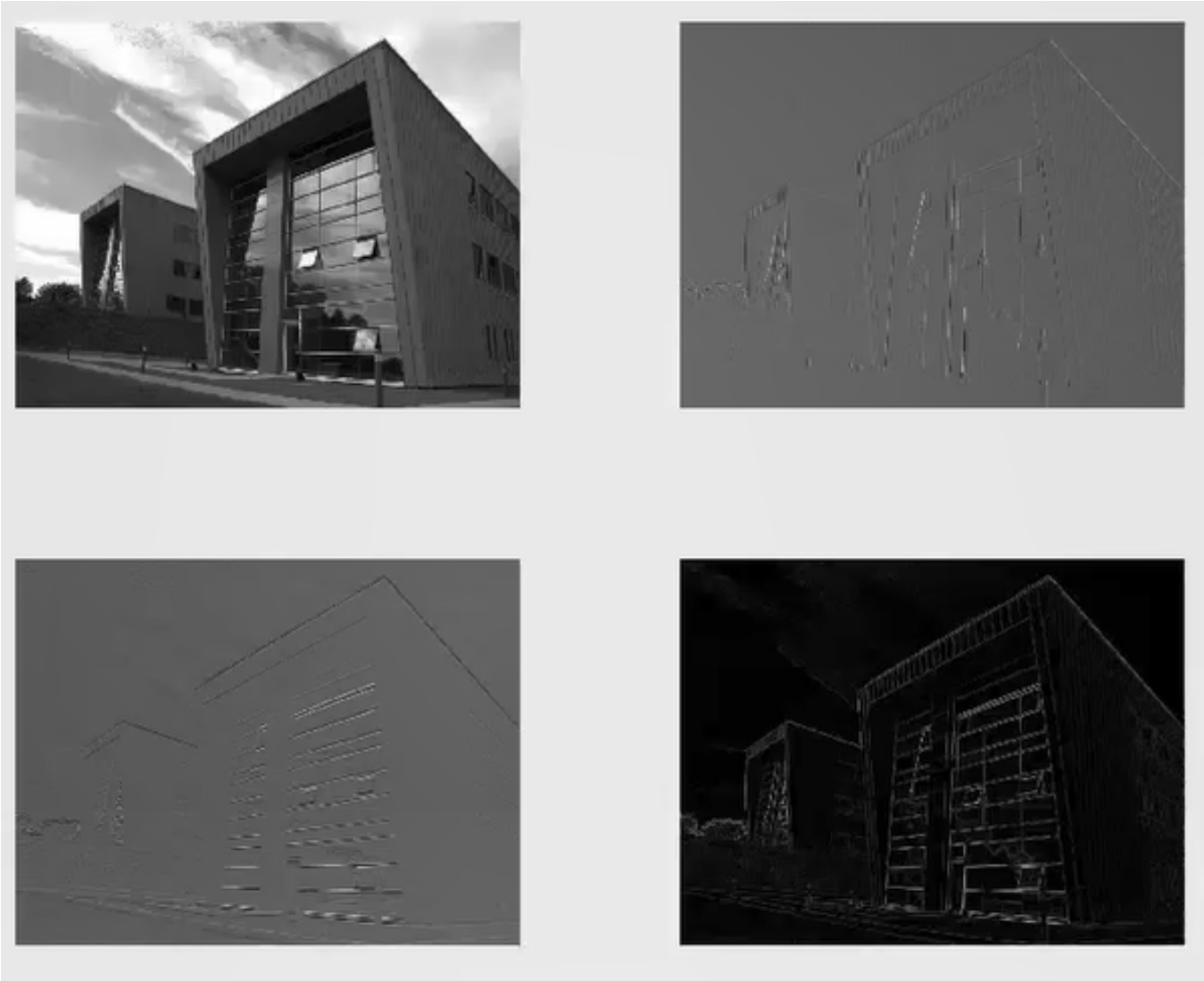
- an edge is a curve that follows a path of rapid change in image intensity
- Edge detection is used to identify the edges in an image
- The most well-known edge-detection method is Canny filter – based on gradients
- But more complex approaches do exist







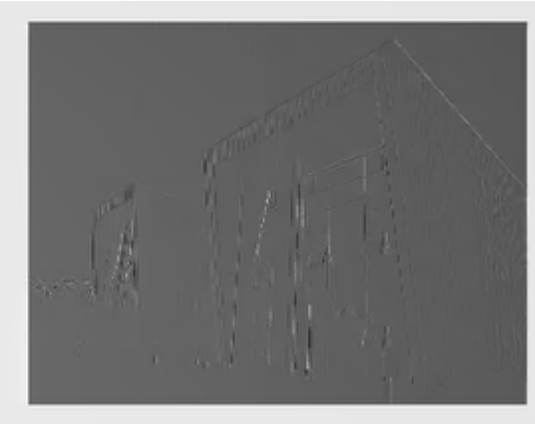
Meaning of Edge Detection Approaches?

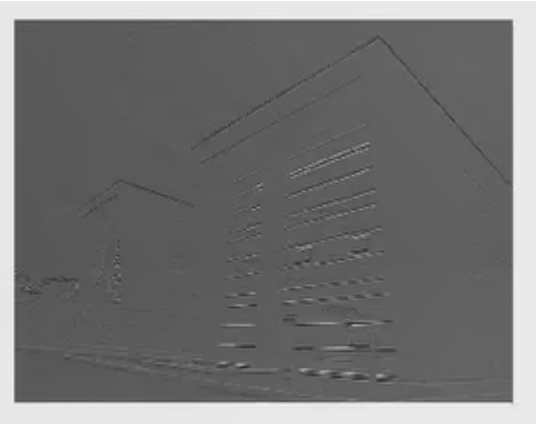


Edge Detection

Sobel Filter

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

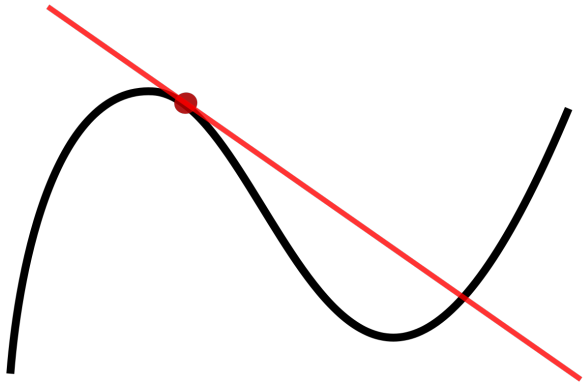




$$\mathbf{G} = \sqrt{\mathbf{G}_x{}^2 + \mathbf{G}_y{}^2}$$

Edge Detection

Differentiation



1st and 2nd derivative wrt x: $\frac{\partial u(x,y)}{\partial x}, \frac{\partial^2 u(x,y)}{\partial x^2}$

2nd derivative wrt y: $\frac{\partial u(x,y)}{\partial y}, \frac{\partial^2 u(x,y)}{\partial y^2}$

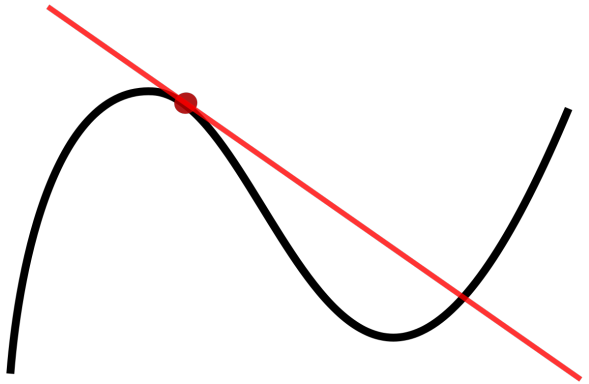
Edge Detector:

$$E(x,y) = \sqrt{\left(\frac{\partial^2}{\partial x^2}\frac{\partial u(x,y)}{\partial y}\right)^2 + \left(\frac{\partial^2}{\partial y^2}\frac{\partial u(x,y)}{\partial x}\right)^2}$$

Achieve matrix representation through discretisation with finite differences

Edge Detection

Finite Differences



1st order wrt x:

$$\frac{\partial u(x,y)}{\partial x} = \lim_{h \to 0} \frac{u(x+h,y) - u(x)}{h}$$

Set $h = 1$:

$$\left.\frac{\partial u(x,y)}{\partial x}\right|_{h=1} = u(x+1,y) - u(x)$$

In the discrete case, we have

$$diff(\boldsymbol{u}) = \boldsymbol{u}[\boldsymbol{i}+\boldsymbol{1},\boldsymbol{j}] - \boldsymbol{u}[i,j]$$

What about the second derivative? ***HINT:*** *apply 1st order twice*

Edge Detection

2nd Order:

$$diff\big(diff(\boldsymbol{u})\big) = \mathbf{1} \cdot \boldsymbol{u}[\boldsymbol{i}+\mathbf{1},\boldsymbol{j}] - \mathbf{2} \cdot \boldsymbol{u}[i,j] + \mathbf{1} \cdot \boldsymbol{u}[\boldsymbol{i}-\mathbf{1},\boldsymbol{j}]$$

Edge Detector:

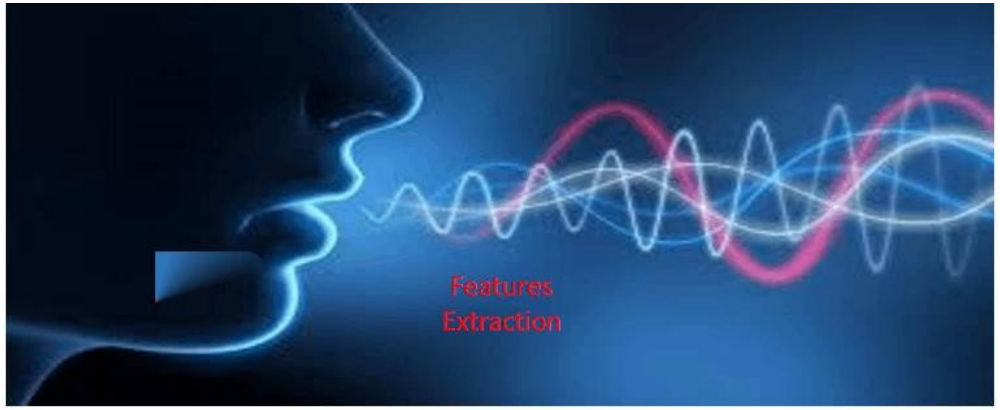
$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ -2 & 0 & +2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & -2 & +1 \\ 0 & 0 & 0 \\ -1 & +2 & -1 \end{bmatrix} * \mathbf{A}$$

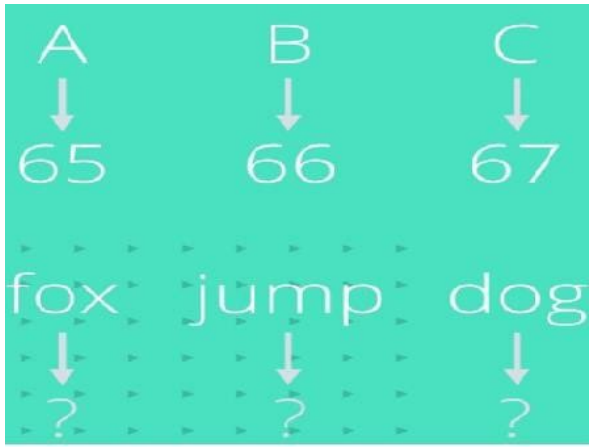
$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

Questions?

Feature Extraction from Text

- How do we extract features from text?
- Text data = a sequence of words
- Words are made up of characters
- Characters are represented with ASCII or Unicode
- The machine do not really make any meaning of from that
- **So, how do we encode meaning representations from text?**





Meaning in Context

Foundational assumption

- You shall know a word by the company it keeps
  - Firth J.R (1957)
- You gain a lot by representing a word in terms of its neighbours
- One of the most successful ideas in text processing

Vector representation

- aims at capturing the similarities between words

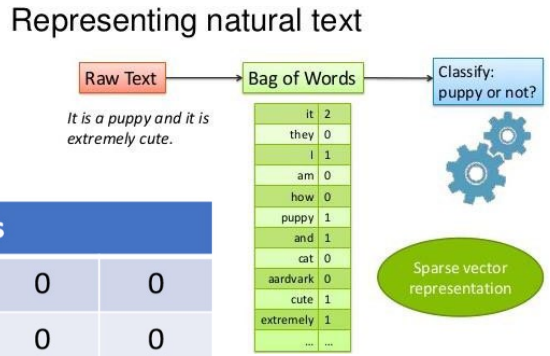
Vector Representation

Common approaches

- Localist approach (1-Hot Vector)
- Bag of Word (BoW) Model
  - TF-IDF, PPMI
- Word Embedding
  - Word2Vec
  - Glove

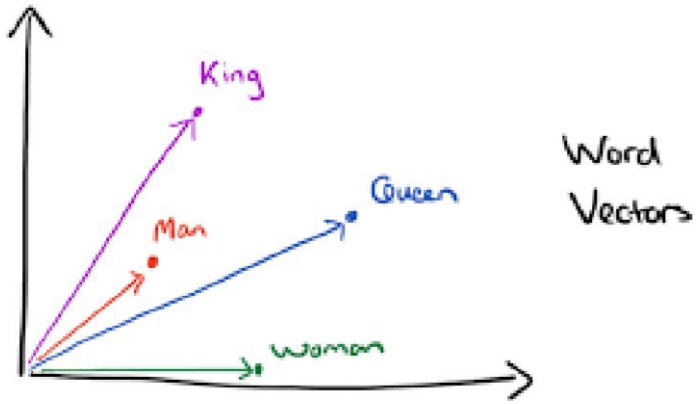
Similarity (distance) measure

- Cosine distance
- Euclidean



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

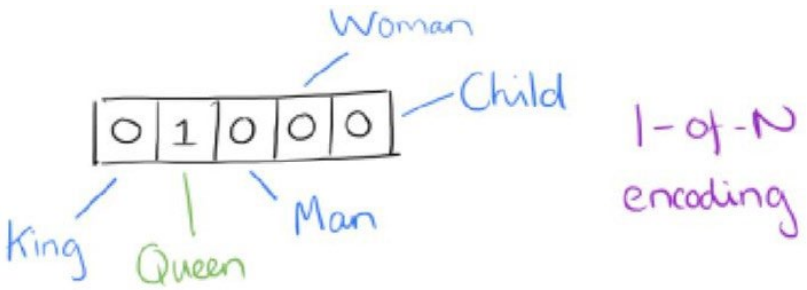
$$\sqrt{\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\|x\| \cdot \|y\|}}$$



1-Hot Encoding

Localist representation

- Vocabulary: [king, queen, man, woman, child]
- Each word is assigned a vector
- 1: in the position of word
- 0: everywhere else



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

English Vocabulary Size?

What is the approximate size of the English vocabulary i.e. how many unique words does the English language have?

a. About 50,000
b. About 500,000
c. About 1,000,000
d. Well over 1 million

English Vocabulary Size?

What is the approximate size of the English vocabulary i.e. how many unique words does the English language have?

a. About 50,000
b. About 500,000 (≈470,000)
c. About 1,000,000
d. Well over 1 million

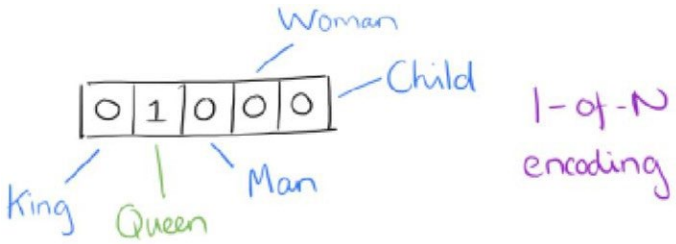
1-Hot for English Vocabulary?

Given what we know about 1-Hot encoding what problem are we likely to have if we use it to encode our text data?

a. No problems at all, it is good method

b. Dimension of vectors will be too low

c. Dimension of vectors will be too high



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

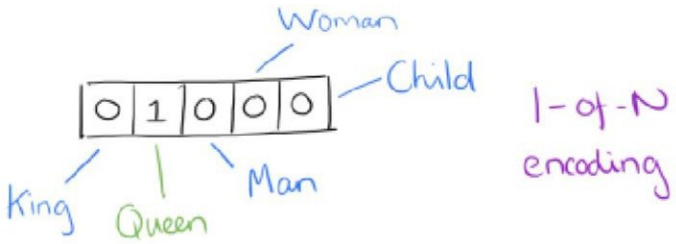
1-Hot for English Vocabulary?

Given what we know about 1-Hot encoding what problem are we likely to have if we use it to encode our text data?

a. No problems at all, it is good method

b. Dimension of vectors will be too low

c. Dimension of vectors will be too high



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

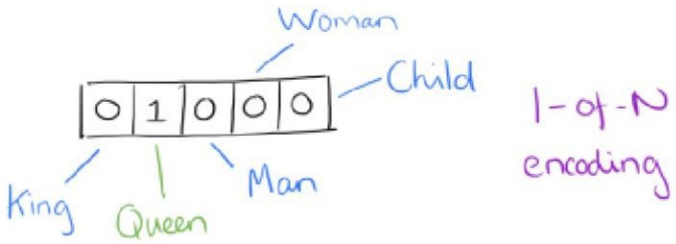
1 – Hot Encoding

Localist representation

- Vocabulary: [king, queen, man, woman, child]
- Each word is assigned a vector
- 1: in the position of word
- 0: everywhere else

Problems

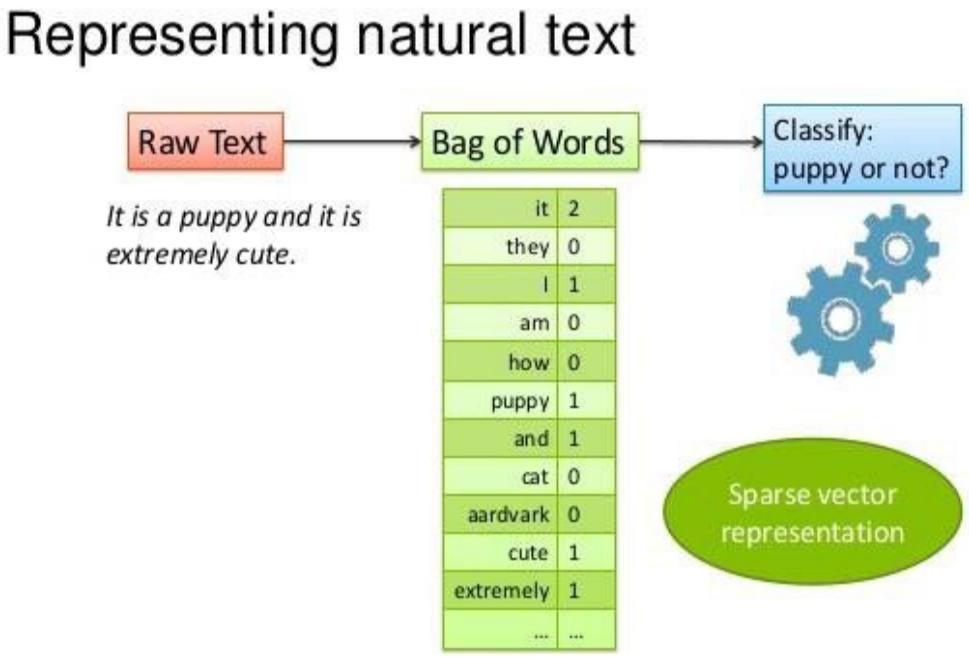
- $|V|$ can be very large 100s of thousands
- Sparse matrix, shape $|V| \times |V|$
  - Memory and compute inefficient
- Orthogonal vectors
  - $similarity(\text{'}king\text{'}, \text{'}queen\text{'}) = 0$



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence



Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.
```

Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.
```

Bag of Words (BoW)

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.

BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"Mary":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

If we have a union of documents e.g.

```
(3) John likes to watch movies. Mary likes movies too. Mary also likes to watch football games.
```

what will the counts for the BoW model be?

```
BoW3 = {"John":□,"likes":□,"to":□,"watch":□,"movies":□,"Mary":□,"too":□,"also":□,"football":□,"games":□};
```

Bag of Words (BoW)

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.

BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"Mary":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

If we have a union of documents e.g.

```
(3) John likes to watch movies. Mary likes movies too. Mary also likes to watch football games.
```

what will the counts for the BoW model be?

```
BoW3 = {"John":1,"likes":3,"to":2,"watch":2,"movies":2,"Mary":2,"too":1,"also":1,"football":1,"games":1};
```

Bag of Words (BoW)

- Advantage
  - Better similarity representation than 1-hot
  - Can be extended to n-grams
- Disadvantage
  - Most common words dominate the counts
    - the, a, is
  - High dimensional vectors
- Solution
  - TF-IDF
  - PPMI
  - Word embedding

```
[
    "John likes",
    "likes to",
    "to watch",
    "watch movies",
    "Mary likes",
    "likes movies",
    "movies too",
]
```

TF-IDF

- Term Frequency – Inverse Document Frequency
  - For a corpus of documents D:
    - *Term Frequency* (TF)
      - Frequency counts (log transformed)
      - $TF_{t,d} = \begin{cases} \mathbf{1 + \log_{10} c(t,d)} & \text{if } c(t,d)>0 \\ \mathbf{0} & \text{otherwise} \end{cases}$
    - *Inverse Document Frequency* (IDF)
      - $|D|$ = # all documents
      - $|\{d \in D : t \in d\}|$ = # documents with t
      - $IDF_t = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$
      - Words like 'the' or '*of*' have low IDF
    - **TF-IDF:** $TF \times IDF$

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

Word Vector (Passage Vector)

Document Vector

PPMI

PPMI

- Pointwise Mutual Information
  - *Do x and y co-occur more than if they were independent?*
  - PMI = $\log_2 \frac{P(x,y)}{P(x) \cdot P(y)}$
- PMI range: $-\infty$ to $+\infty$
- -ves are problematic
  - Use Positive PMI (PPMI)
  - PPMI $\max(\log_2 \frac{P(x,y)}{P(x) \cdot P(y)}, 0)$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w,context) **p(w)**

	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

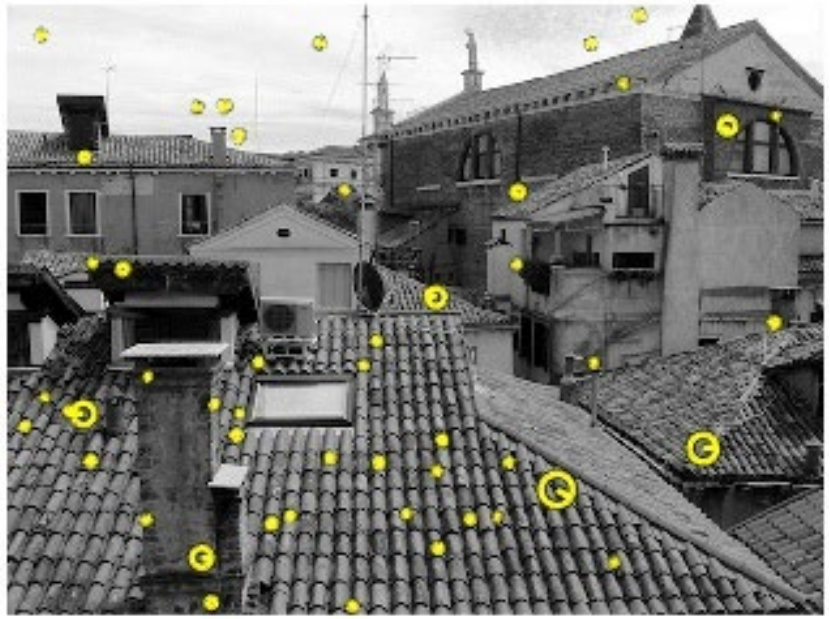
PPMI(w,context)

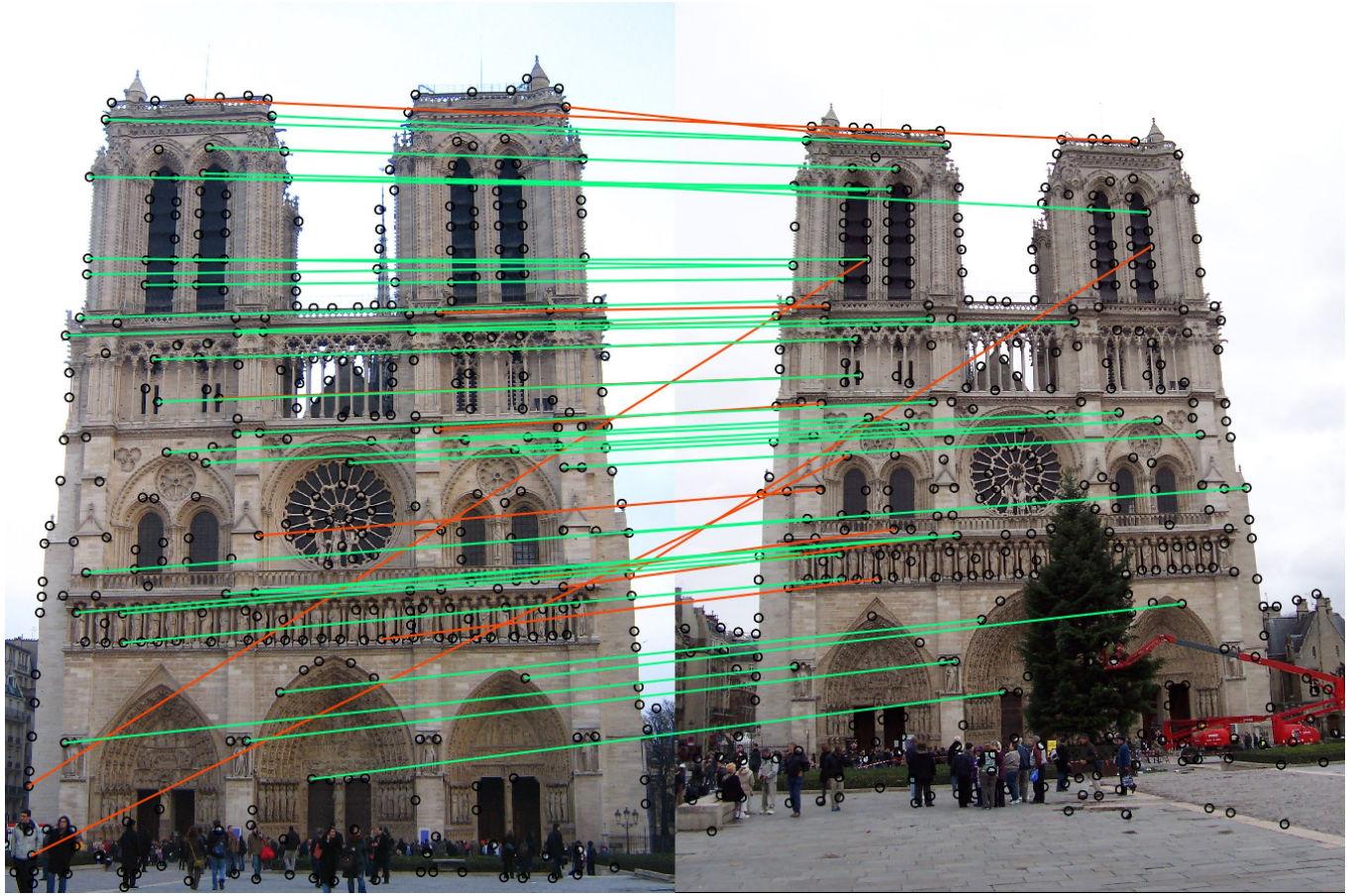
	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Questions?

Feature Matching and Comparison

- Largely problem dependent
- Simple vs Complex
- Supervised vs unsupervised

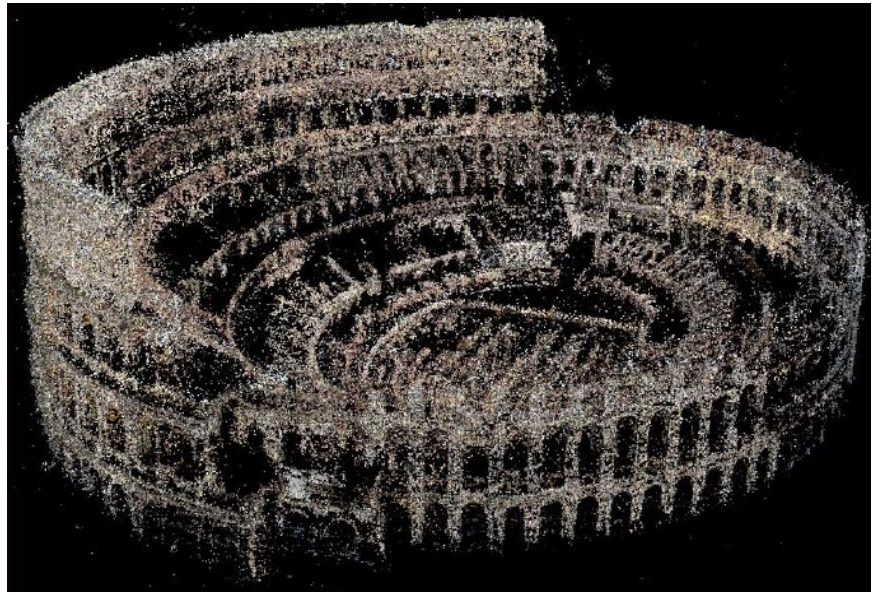




Some applications of Feature Matching

- Image alignment and stitching
- 3D structure from images
- Motion tracking
- Robot navigation
- Image retrieval
- Object and face recognition
- Human action recognition

Some applications of Feature Matching



Lancaster University

SCC361: Artificial Intelligence

Week 3: Computer Vision and Natural Language Processing

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

Attendance Check-in

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the “You are checked in” confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Lecture Plan

Weeks 1-5

1. Introduction to Artificial Intelligence and Machine Learning
2. Features in Machine Learning and Feature Extraction
3. **Computer Vision and Natural Language Processing**
4. Clustering and Classification
5. Artificial Neural Networks

Weeks 6-10:

6. Genetic Algorithms
7. Naïve Bayesian Classifier
8. Decision Tree Classifier
9. Introduction to Deep Neural Networks
10. Introduction to Convolutional Neural Networks

Expected Learning Outcomes

By the end of this week:

- Understand some core concepts of computer vision
- Understand some core concepts of natural language processing
- Prepare for moving to clustering and classification

This Week

Computer Vision

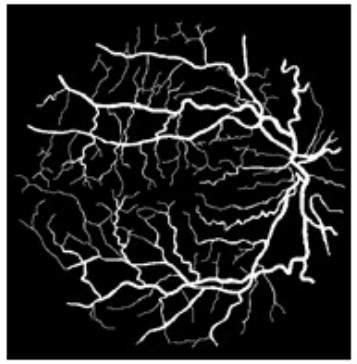
Natural Language Processing

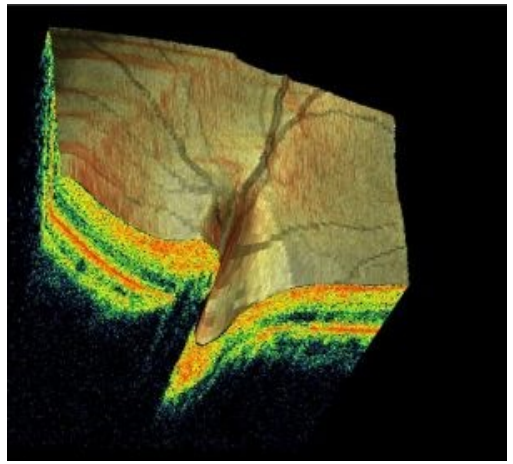
Last Week

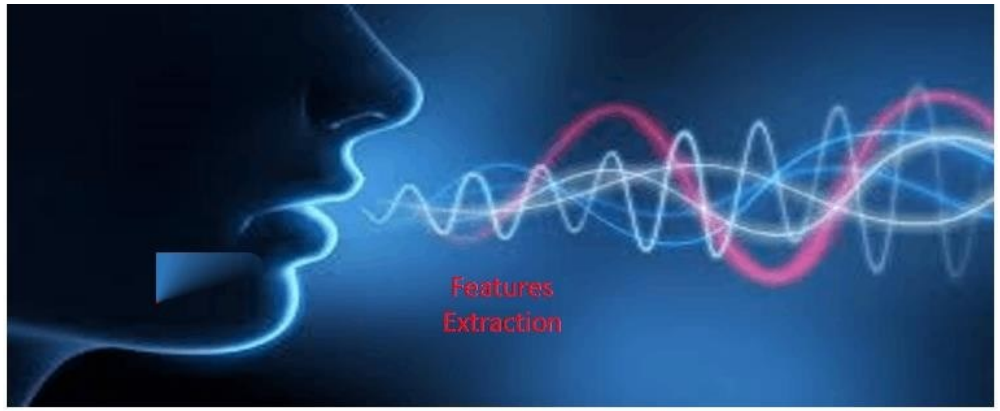
A **feature** is an individual measurable property or characteristic of a phenomenon.

A **feature vector** represents numeric or symbolic characteristics, called features, of an object in a mathematical analysable way

Features and Labels

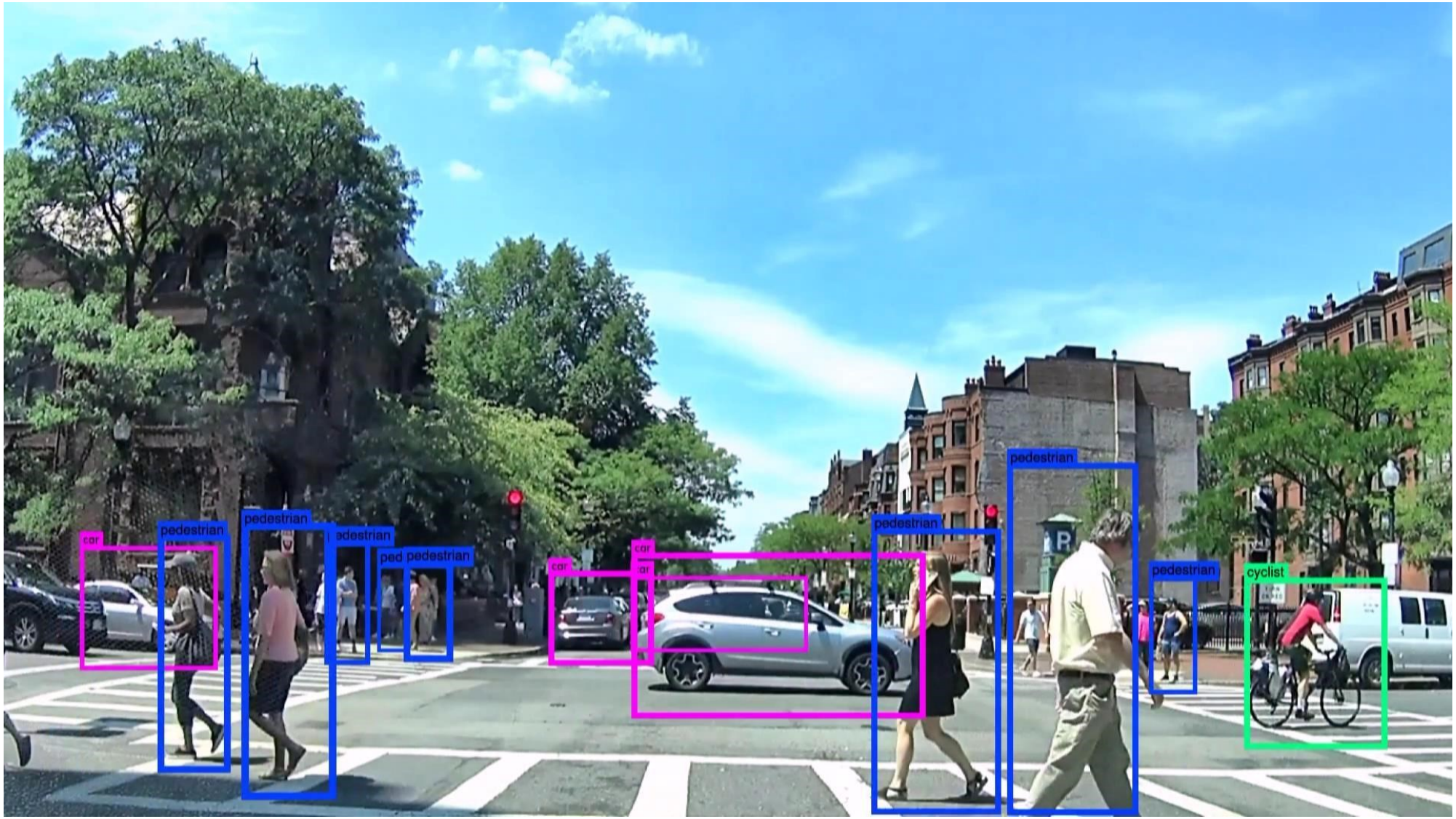






Measure1	Measure2	Measure3	Measure4	Class1	Class2
5.1	3.5	1.4	0.2	1	setosa
4.9	3	1.4	0.2	1	setosa
4.7	3.2	1.3	0.2	1	setosa
4.6	3.1	1.5	0.2	1	setosa
5	3.6	1.4	0.2	1	setosa
5.4	3.9	1.7	0.4	1	setosa
4.6	3.4	1.4	0.3	1	setosa
5	3.4	1.5	0.2	1	setosa
4.4	2.9	1.4	0.2	1	setosa

Encoding the world around us



Encoding Image Data

Image maps integer coordinates to intensity values

$$u: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{R}$$

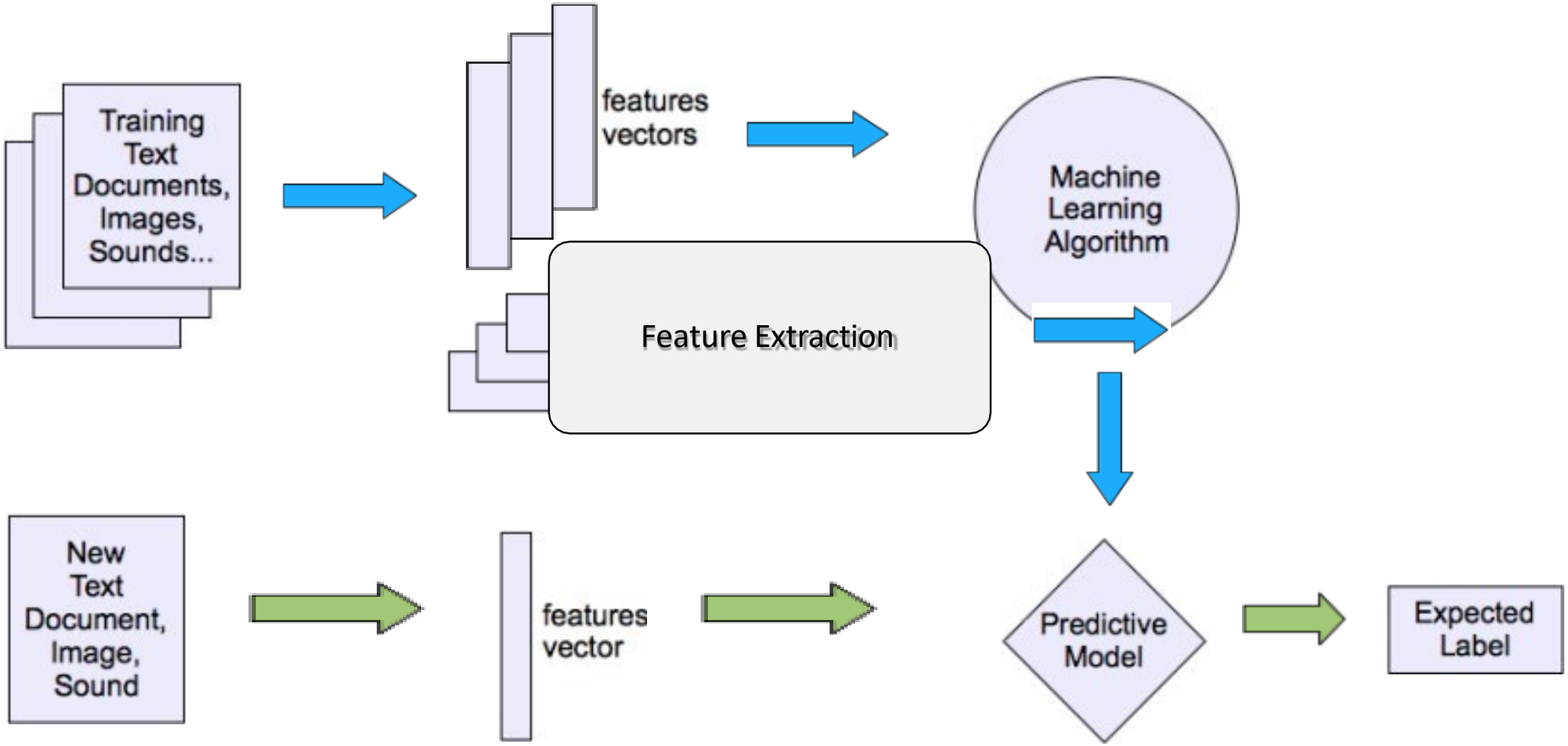
$$\boldsymbol{u} \in [0,1]^{682 \times 1024 \times 3}$$

$$\boldsymbol{u}^r(\Omega) = \begin{bmatrix} 0.55 & 0.72 & 0.44 \\ 0.23 & 0.56 & 0.23 \\ 0.67 & 0.52 & 0.93 \end{bmatrix}$$

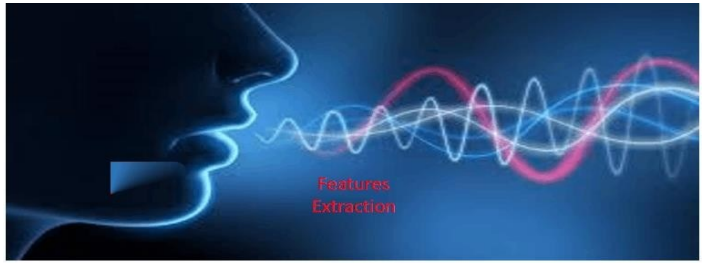
$$\boldsymbol{u}^g(\Omega) = \begin{bmatrix} 0.45 & 0.23 & 0.76 \\ 0.76 & 0.34 & 0.76 \\ 0.94 & 0.98 & 0.93 \end{bmatrix}$$

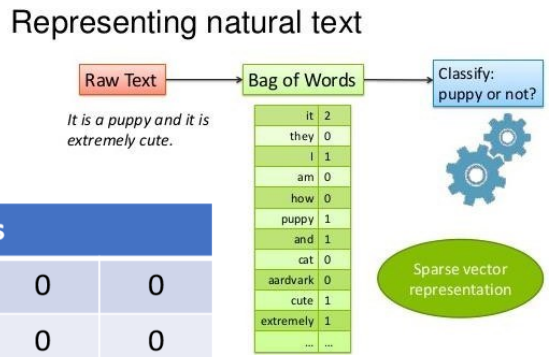
$$\boldsymbol{u}^b(\Omega) = \begin{bmatrix} 0.56 & 0.57 & 0.34 \\ 0.12 & 0.57 & 0.75 \\ 0.56 & 0.76 & 0.45 \end{bmatrix}$$

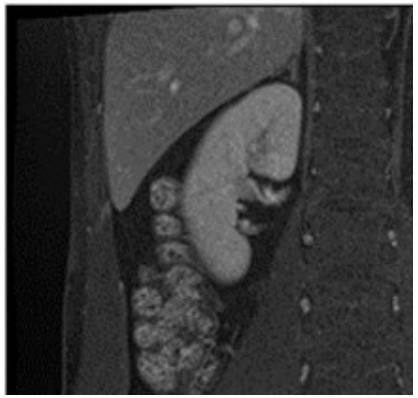
Features for ML

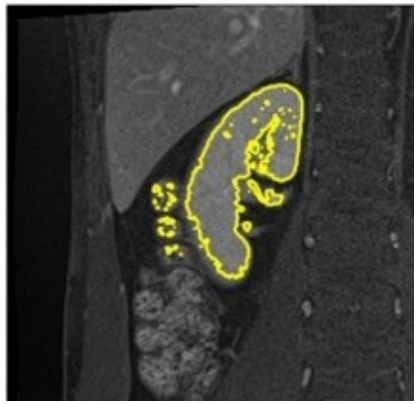


Feature Extraction

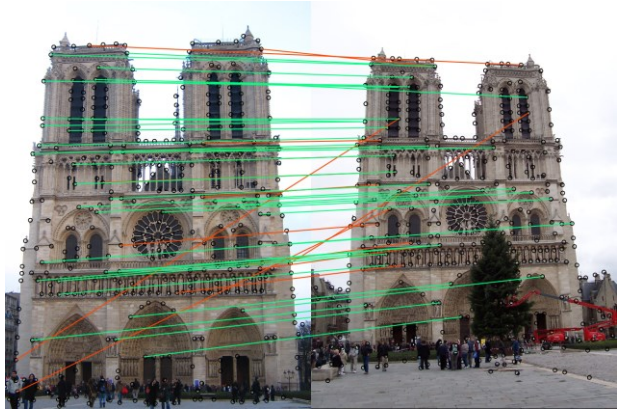


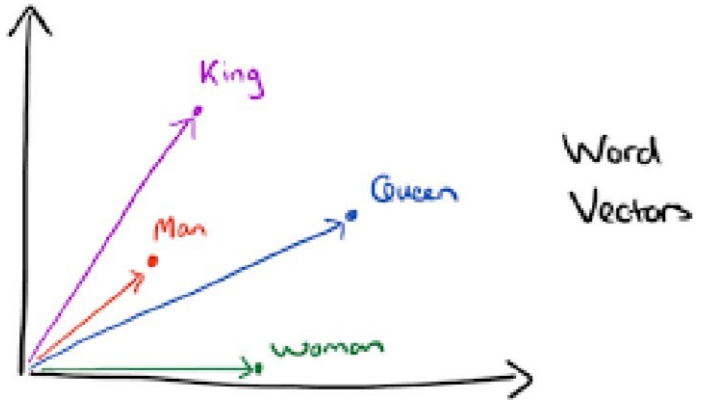






word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

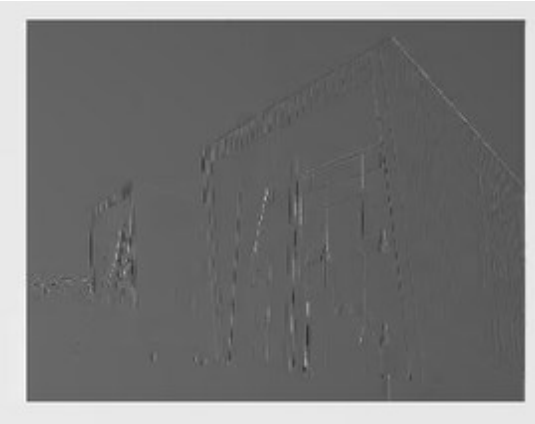


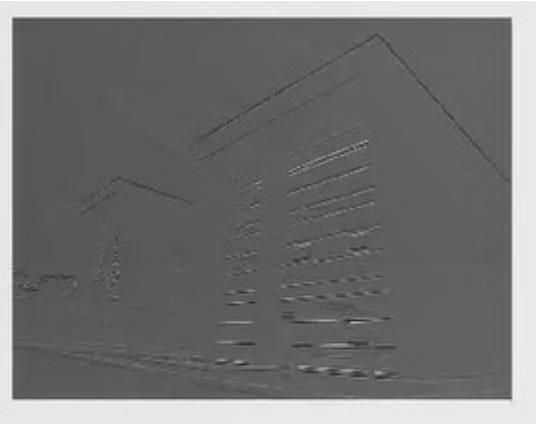


Feature Extraction

Sobel Filter

$$\mathbf{G}_x = \begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A} \quad \text{and} \quad \mathbf{G}_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

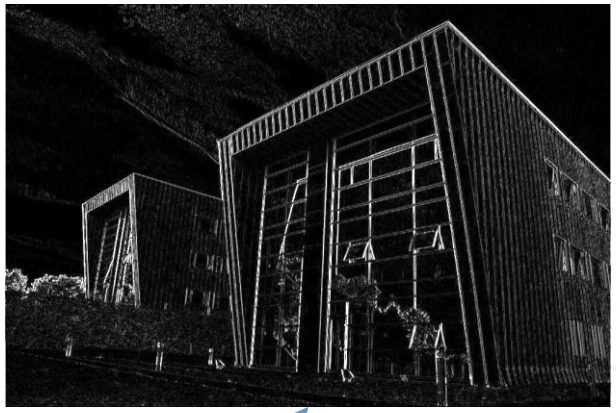




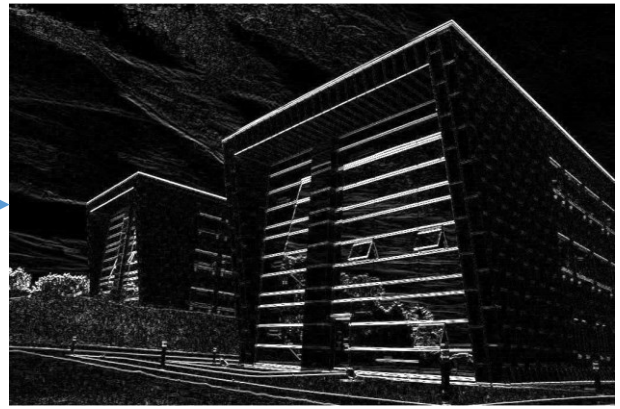
$$\mathbf{G} = \sqrt{{\mathbf{G}_x}^2 + {\mathbf{G}_y}^2}$$

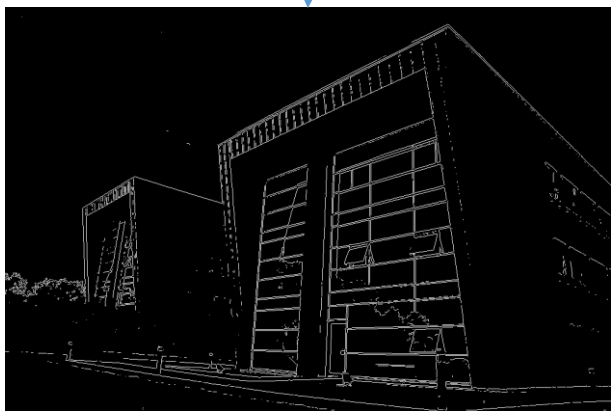
Feature Extraction

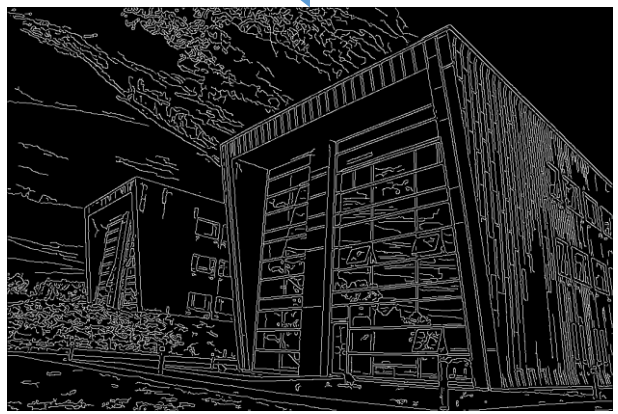
$$\begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

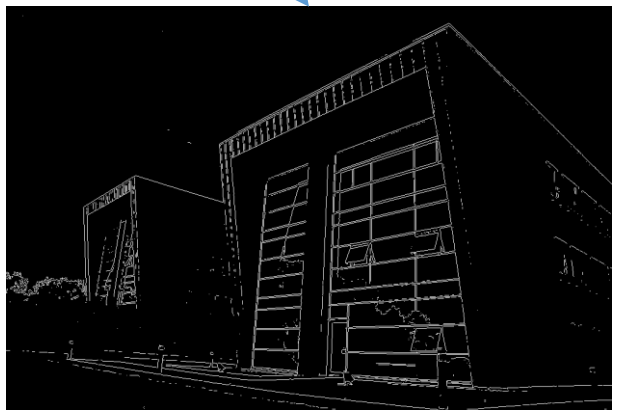


$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix}$$



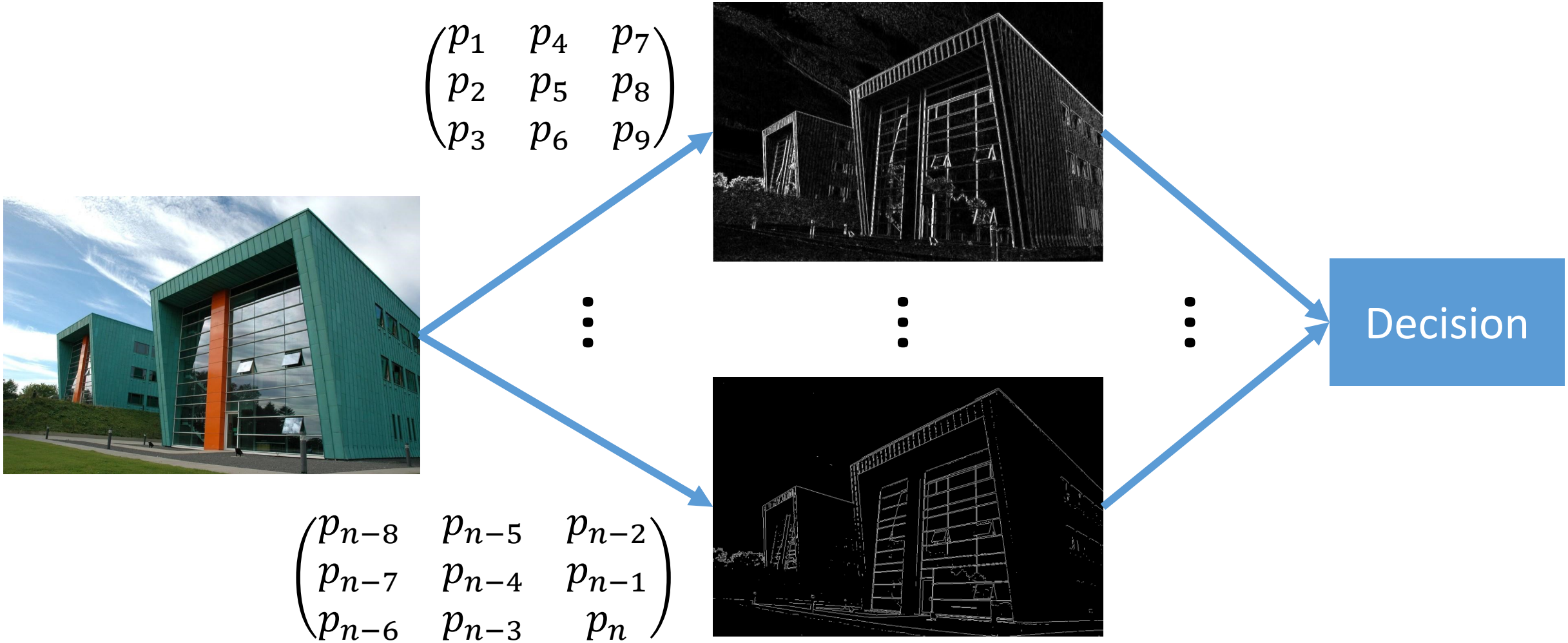






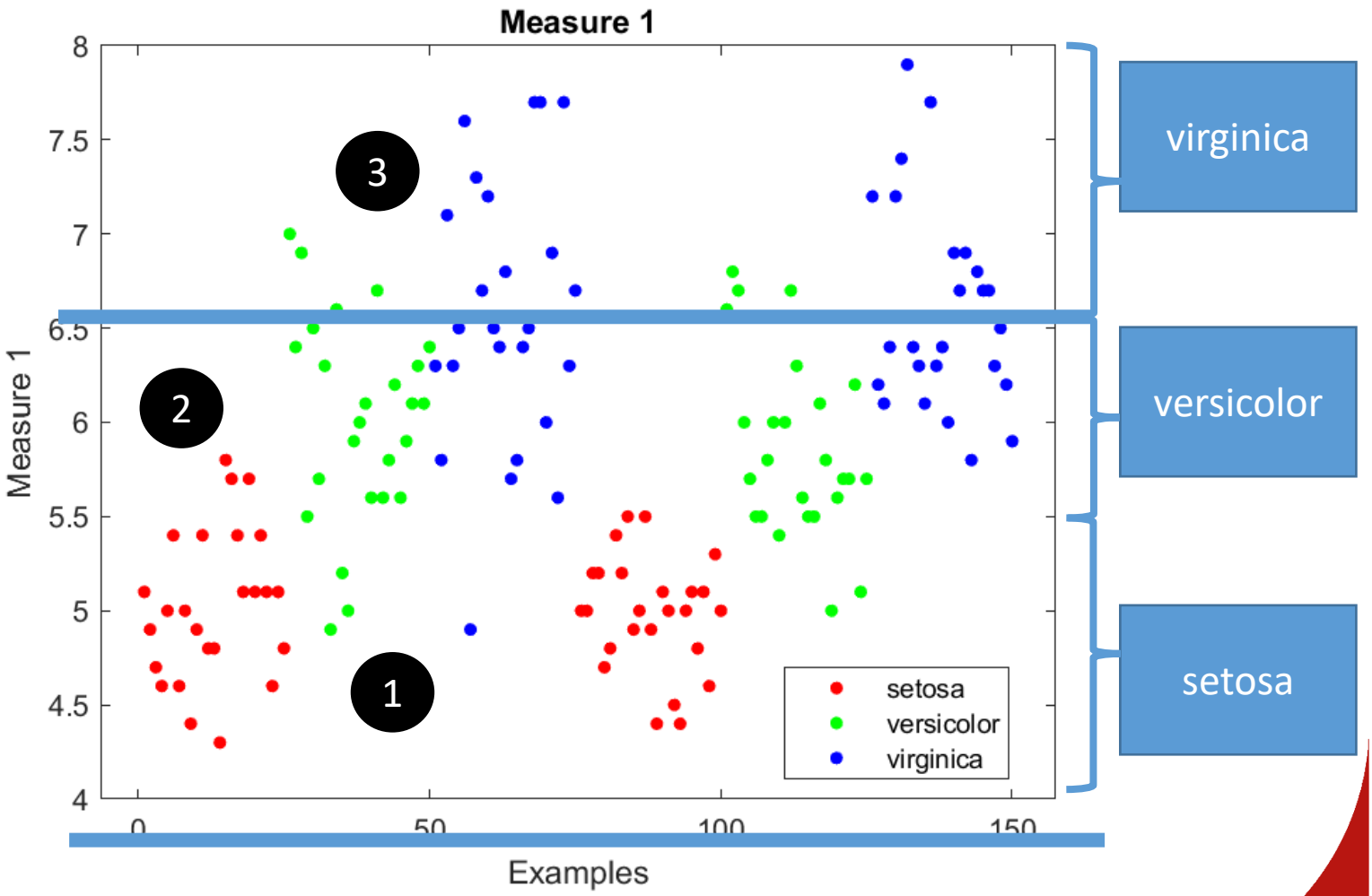
$$\sqrt{(a * u)^2 + (b * u)^2}$$

Feature Extraction

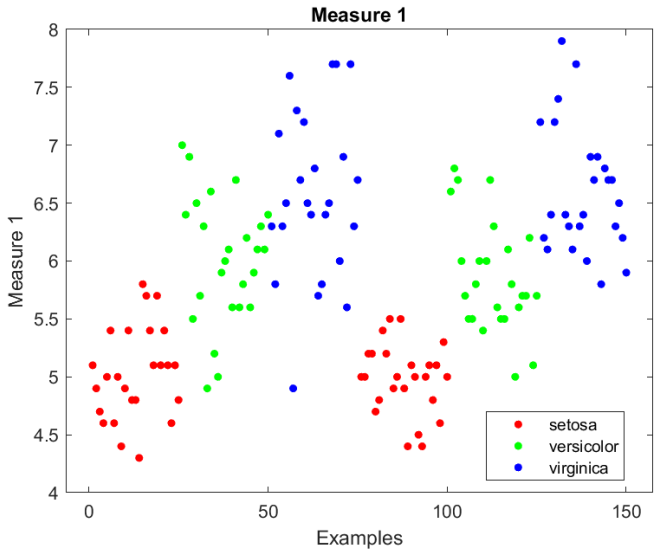


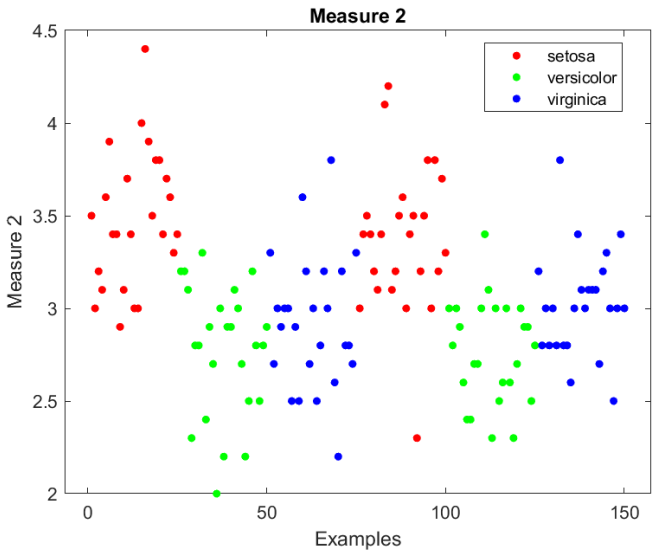
Decision Making

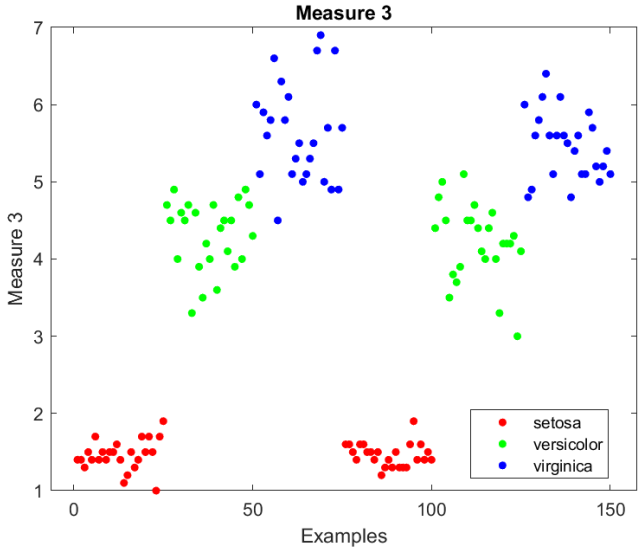
Measure1	Measure2	Measure3	Measure4	Class1	Class2
5.1	3.5	1.4	0.2	1	setosa
4.9	3	1.4	0.2	1	setosa
4.7	3.2	1.3	0.2	1	setosa
4.6	3.1	1.5	0.2	1	setosa
5	3.6	1.4	0.2	1	setosa
5.4	3.9	1.7	0.4	1	setosa
4.6	3.4	1.4	0.3	1	setosa
5	3.4	1.5	0.2	1	setosa
4.4	2.9	1.4	0.2	1	setosa

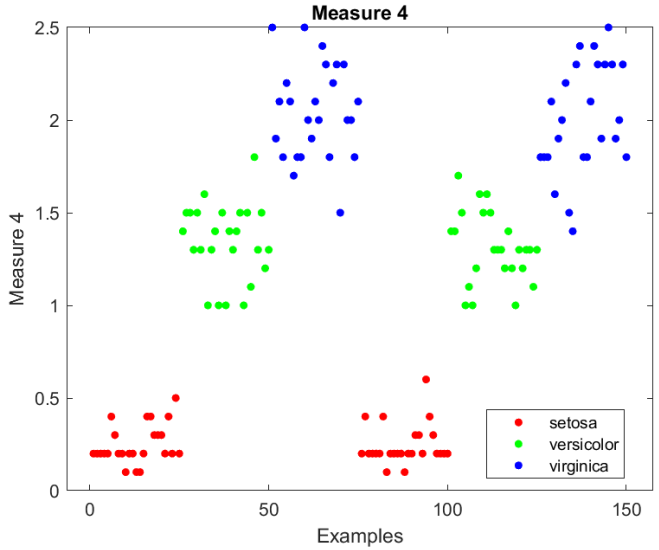


Decision Making

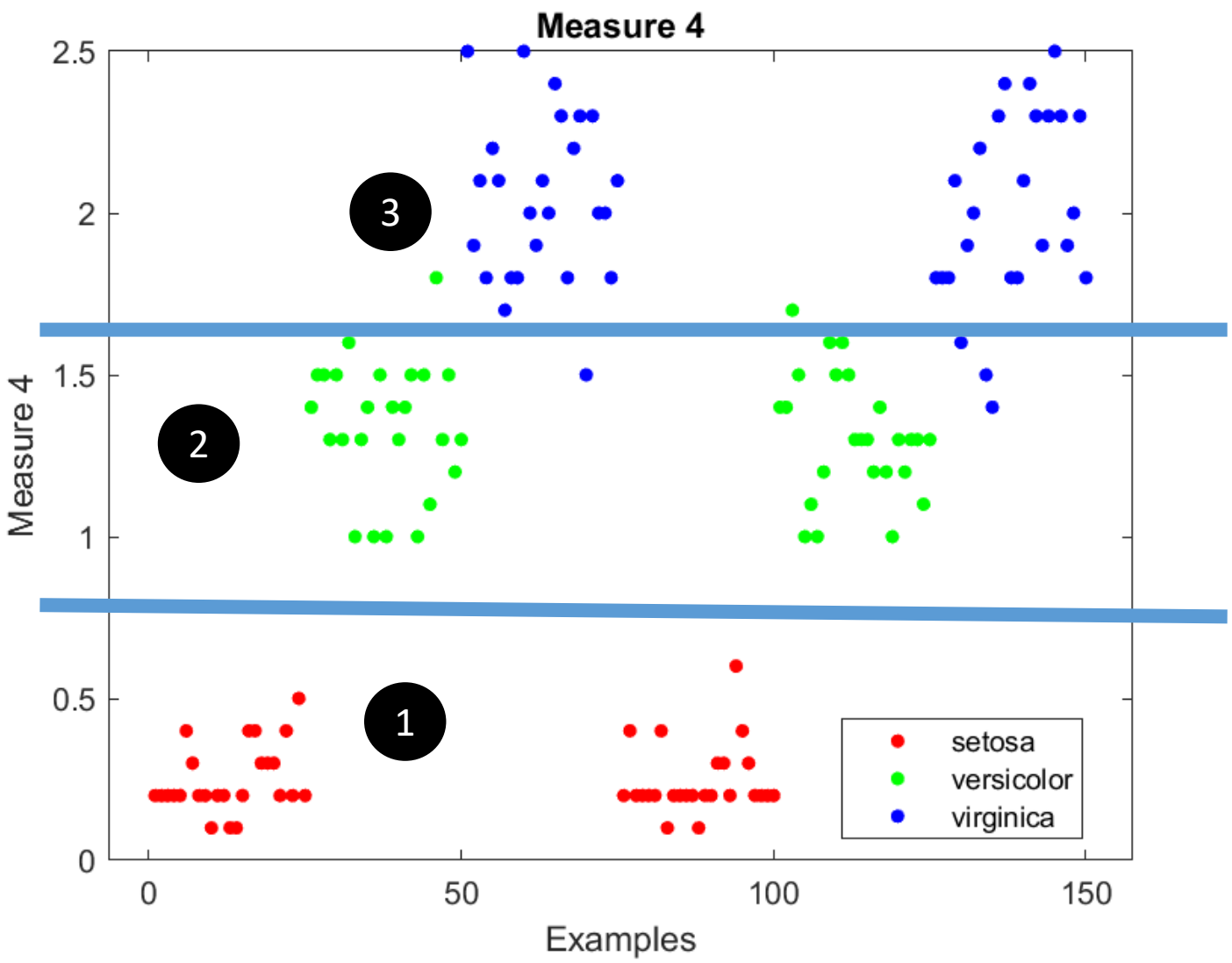




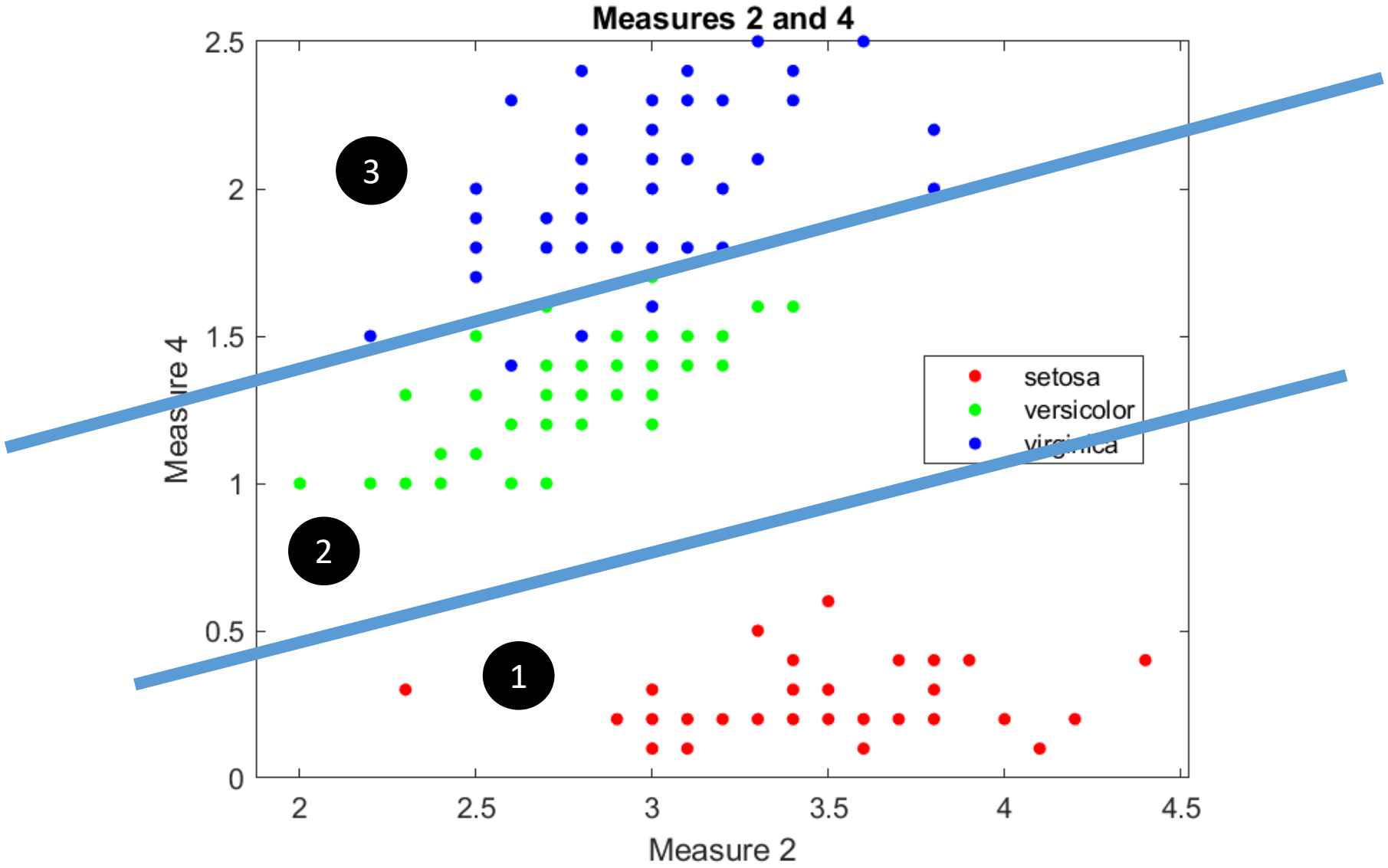




Decision Making



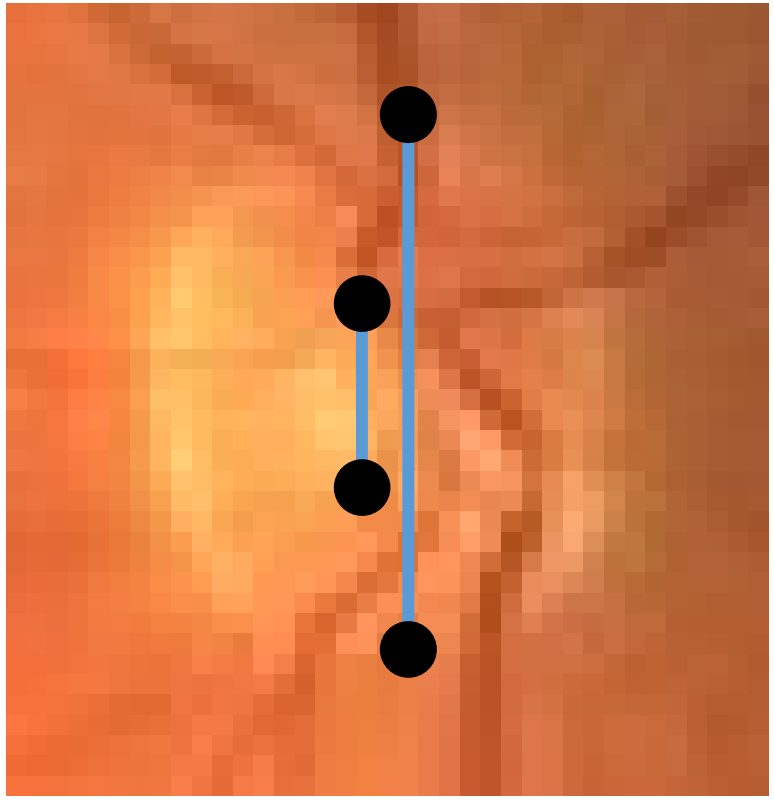
Decision Making



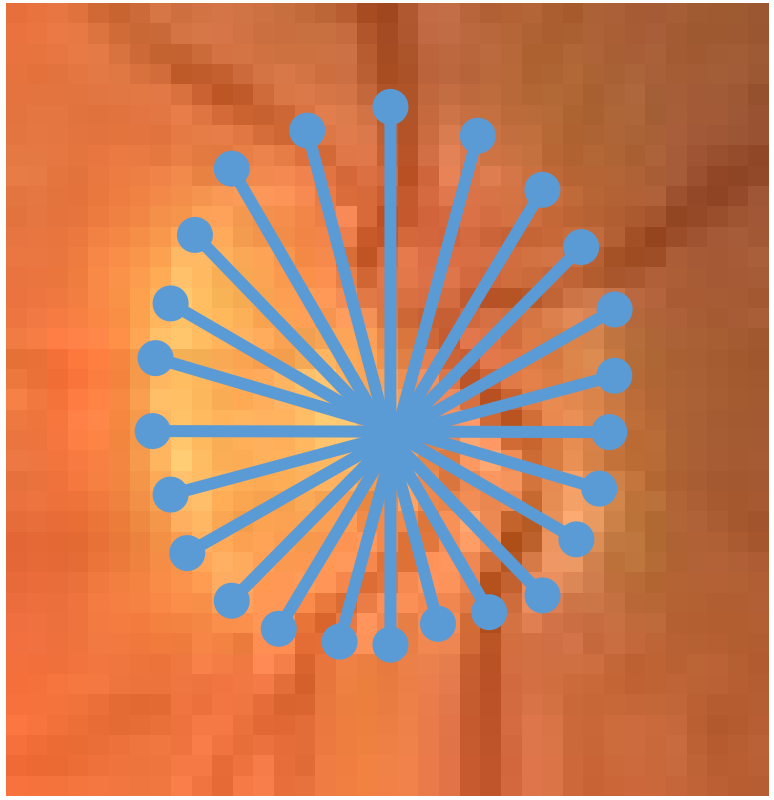
Getting the Info

Measure1	Measure2	Measure3	Measure4	Class1	Class2
5.1	3.5	1.4	0.2	1	setosa
4.9	3	1.4	0.2	1	setosa
4.7	3.2	1.3	0.2	1	setosa
4.6	3.1	1.5	0.2	1	setosa
5	3.6	1.4	0.2	1	setosa
5.4	3.9	1.7	0.4	1	setosa
4.6	3.4	1.4	0.3	1	setosa
5	3.4	1.5	0.2	1	setosa
4.4	2.9	1.4	0.2	1	setosa

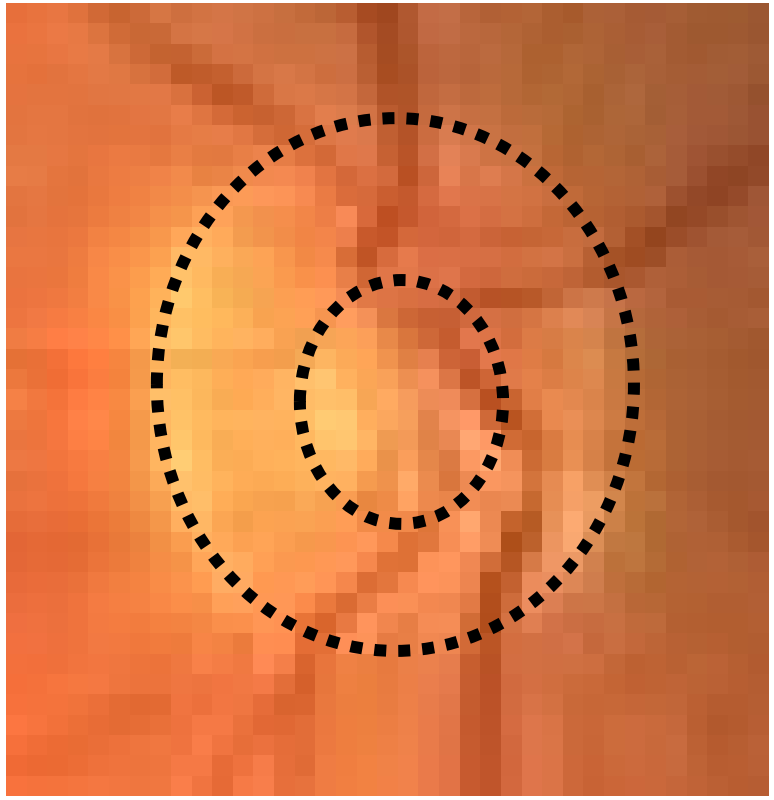
Getting the Info



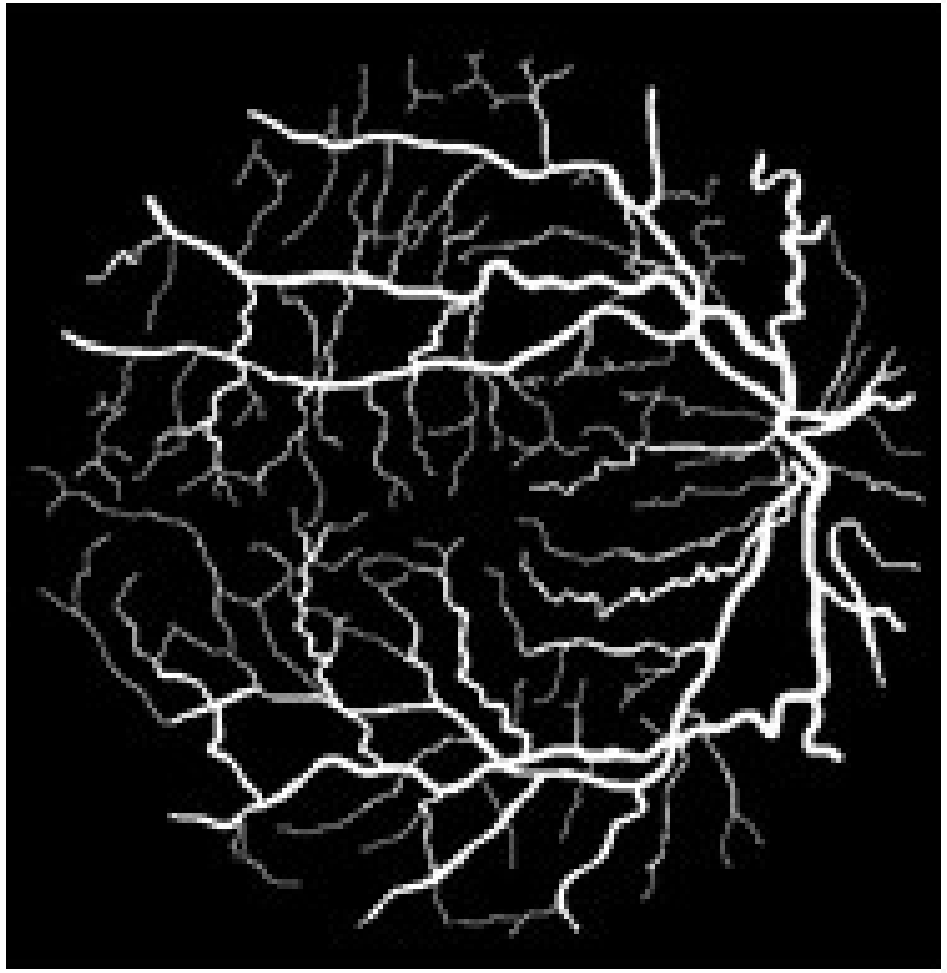
Getting the Info



Getting the Info

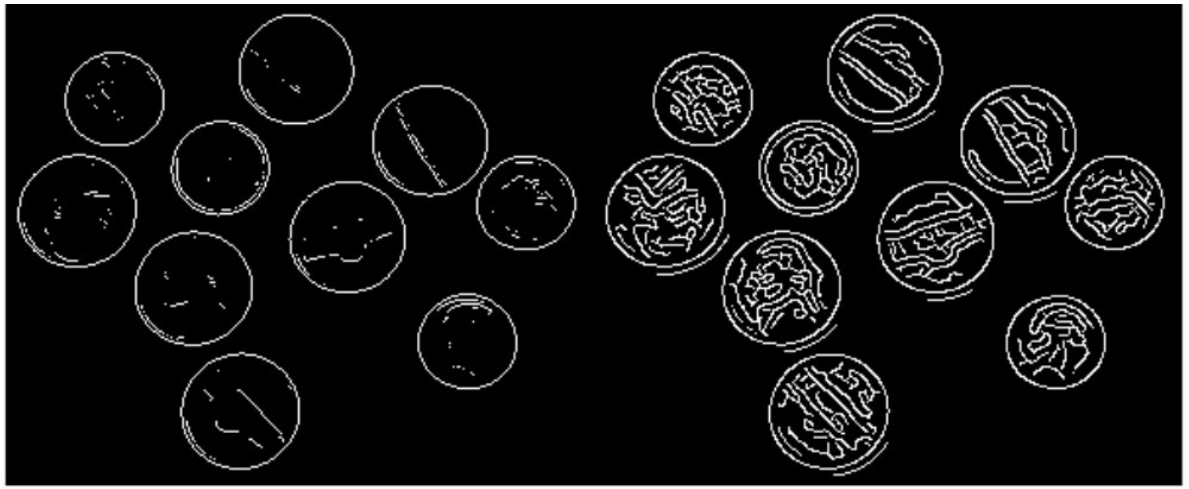


Getting the Info

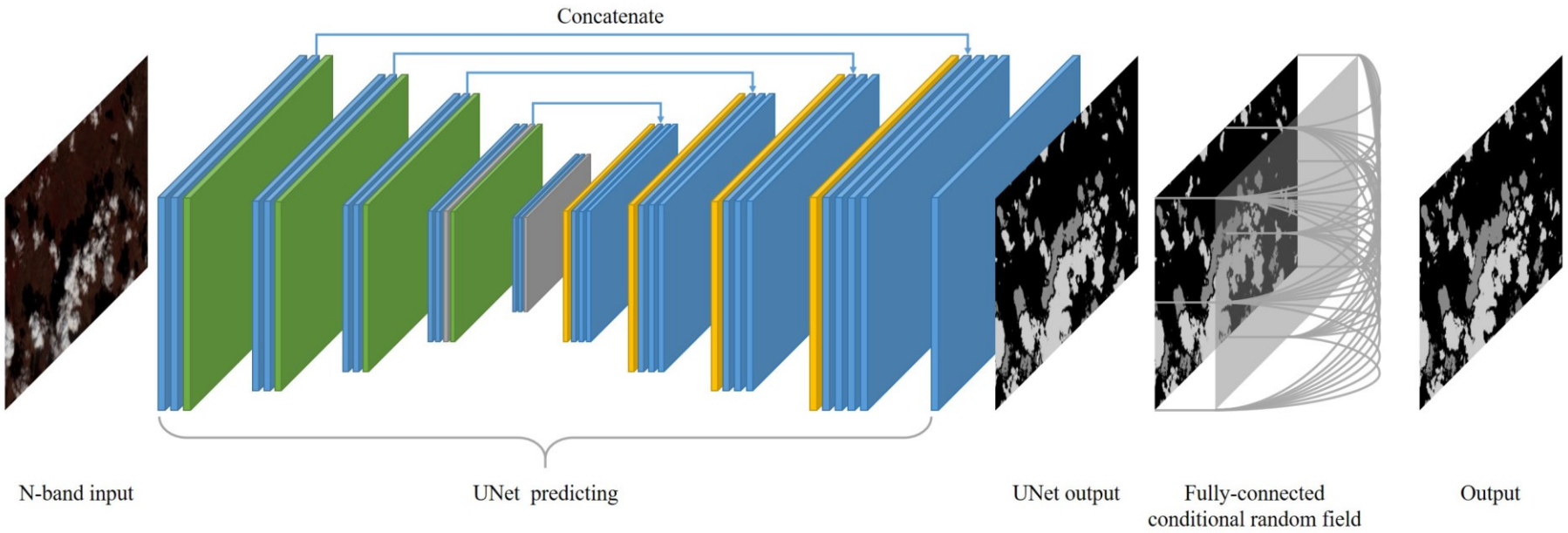


Getting the Info

Extract features automatically



Embed feature extraction

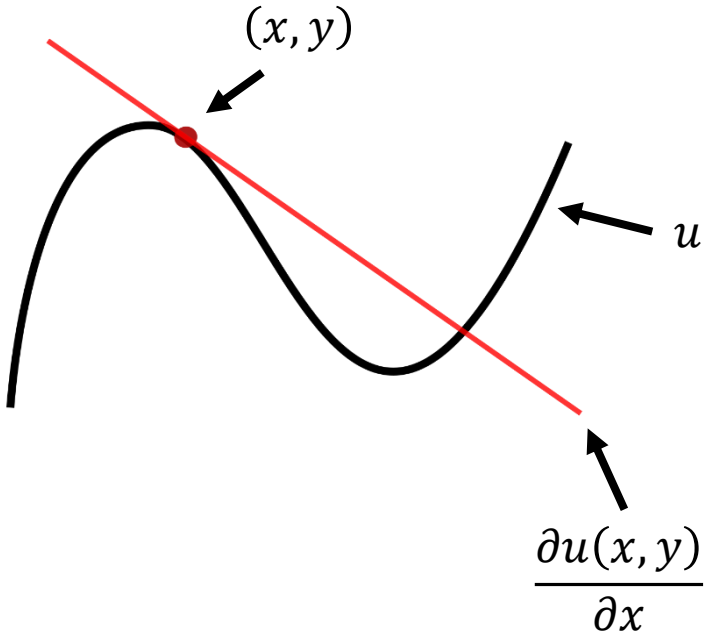


Last Week: Edge Detection

Finite Differences

1st order wrt x:

$$\frac{\partial u(x,y)}{\partial x} = \lim_{h \to 0} \frac{u(x+h,y) - u(x)}{h}$$



Set $h = 1$:

$$\left.\frac{\partial u(x,y)}{\partial x}\right|_{h=1} = u(x+1,y) - u(x)$$

In the discrete case, we have

$$diff(\boldsymbol{u}) = \boldsymbol{u}[\boldsymbol{i}+\mathbf{1},\boldsymbol{j}] - \boldsymbol{u}[i,j]$$

What about the second derivative? ***HINT:*** *apply 1st order twice*

Last Week: Edge Detection

Key take home message:

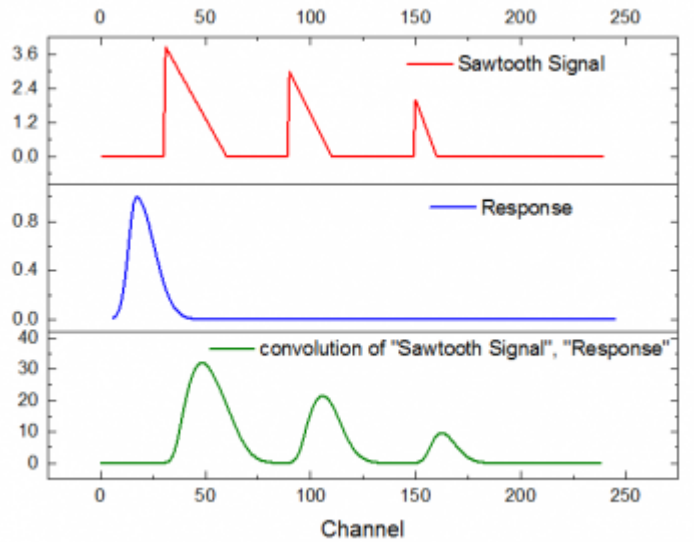
	Theory:	Implementation
First order derivative wrt x	$\frac{\partial u(x,y)}{\partial x}$	$\boldsymbol{u}[i+1,j]-\boldsymbol{u}[i,j]$
First order derivative wrt y	$\frac{\partial u(x,y)}{\partial y}$	$\boldsymbol{u}[i,j+1]-\boldsymbol{u}[i,j]$
Second order derivative wrt x	$\frac{\partial^2 u(x,y)}{\partial x^2}$	$\boldsymbol{u}[i-1,j]-\mathbf{2}\boldsymbol{u}[i,j]+\boldsymbol{u}[i+1,j]$
Second order derivative wrt y	$\frac{\partial^2 u(x,y)}{\partial y^2}$	$\boldsymbol{u}[i,j-1]-\mathbf{2}\boldsymbol{u}[i,j]+\boldsymbol{u}[i,j+1]$

$$(f * g)(x_2, y_2)$$

$$= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, y_1)\, g(x_2 - x_1, y_2 - y_1) \mathrm{d}y_1\, \mathrm{d}x_1$$

Convolution

Convolution is a **mathematical operation** on two functions (f and g) that produces a third function ($f * g$) that expresses how the shape of one is modified by the other.



Let's break this down

Real Domain

$$(k * u)(x_2, y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x_1, y_1)\, u(x_2 - x_1, y_2 - y_1) \mathrm{d}y_1\, \mathrm{d}x_1$$

1. Notation
2. Summation
3. Integration

Notation

$$(k * u)(x_2, y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x_1, y_1)\, u(x_2 - x_1, y_2 - y_1) \mathrm{d}y_1\, \mathrm{d}x_1$$

f	:	a letter in italics usually denotes a variable or a function
$f(x)$	:	f is a function of x
$f(x, y)$	:	f is a function of x and y
$\mathbf{u}$	:	a letter in bold usually denotes a vector or matrix/array
$\mathbf{u}[i, j]$	:	the value stored in the ith row and jth column of $\mathbf{u}$
$\approx$	:	approximately equal to

Summation

Let $\mathbf{u} \in \mathbb{R}^n$

The summation of $\mathbf{u}$ from 1 to n

$$\sum_{x=1}^{n} \mathbf{u}[x] = \mathbf{u}[1] + \mathbf{u}[2] + \cdots + \mathbf{u}[n]$$

Can think of this as a *for* loop:

```
count = 0;
for x = 1:n
    count = count + u(x);
end
```

or we could form a vector and use a summation

```
count = sum(u(1:n))
```

Summation

Let $\mathbf{u} \in \mathbb{R}^n$ and assume $1 \leq a, b \leq n$

i.e. a and b and greater than or equal to 1 and less than or equal to n

The summation of $\mathbf{u}$ from a to b

$$\sum_{x=a}^{b} \mathbf{u}[x] = \mathbf{u}[a] + \mathbf{u}[a+1] + \cdots + \mathbf{u}[b]$$

Can think of this as a *for* loop:

```
count = 0;
for x = a:b
    count = count + u(x);
end
```

or we could form a vector and use a summation

```
count = sum(u(a:b))
```

Summation

Let $\mathbf{u} \in \mathbb{R}^n$ and $A = \{1,2,\dots,n\}$

A is a set of values

The summation of $\mathbf{u}$ over the set A

$$\sum_{x \in A} \mathbf{u}[x] = \mathbf{u}[1] + \mathbf{u}[2] + \cdots + \mathbf{u}[n]$$

Can think of this as a *for* loop:

```
count = 0;
for x = 1:length(A)
    count = count + u(A(x));
end
```

or we could form a vector and use a summation

```
count = sum(u(A))
```

Summation

Let $\mathbf{u} \in \mathbb{R}^{m \times n}$ and assume $1 \leq a, b \leq m$ and $1 \leq c, d \leq n$

The summation of $\mathbf{u}$ from a to b

$$\sum_{y=c}^{d} \sum_{x=a}^{b} \mathbf{u}[x, y] = \sum_{y=c}^{d} \left(\sum_{x=a}^{b} \mathbf{u}[x, y] \right) = \sum_{y=c}^{d} (\mathbf{u}[a, y] + \mathbf{u}[a+1, y] + \cdots + \mathbf{u}[b, y])$$

$$\begin{aligned}
&= \mathbf{u}[a, c] + \mathbf{u}[a+1, c] + \cdots + \mathbf{u}[b, c] && \Big\} \; y = c \\
&+ \mathbf{u}[a, c+1] + \mathbf{u}[a+1, c+1] + \cdots + \mathbf{u}[b, c+1] && \Big\} \; y = c+1 \\
&+ \ \ldots \\
&+ \underbrace{\mathbf{u}[a, d]}_{x = a} + \underbrace{\mathbf{u}[a+1, d]}_{x = a+1} + \cdots + \underbrace{\mathbf{u}[b, d]}_{x = b} && \Big\} \; y = d
\end{aligned}$$

Summation

Let $\mathbf{u} \in \mathbb{R}^{m \times n}$ and assume $1 \leq a, b \leq m$ and $1 \leq c, d \leq n$

The summation of $\mathbf{u}$ from a to b

$$\sum_{y=c}^{d} \sum_{x=a}^{b} \mathbf{u}[x, y] = \sum_{y=c}^{d} \left(\sum_{x=a}^{b} \mathbf{u}[x, y] \right) = \sum_{y=c}^{d} (\mathbf{u}[a, y] + \mathbf{u}[a+1, y] + \cdots + \mathbf{u}[b, y])$$

Can think of this as a nested *for* loop:

```
count = 0;
for x = a:b
    for y = c:d
        count = count + u(x,y);
    end
end
```

or we could form a vector and use a summation

```
count =
sum(u(a:b,c:d),'all')
```

Integration

Integration is the inverse of differentiation

Examples:

1. If

$$f(x) = x^3$$

Then

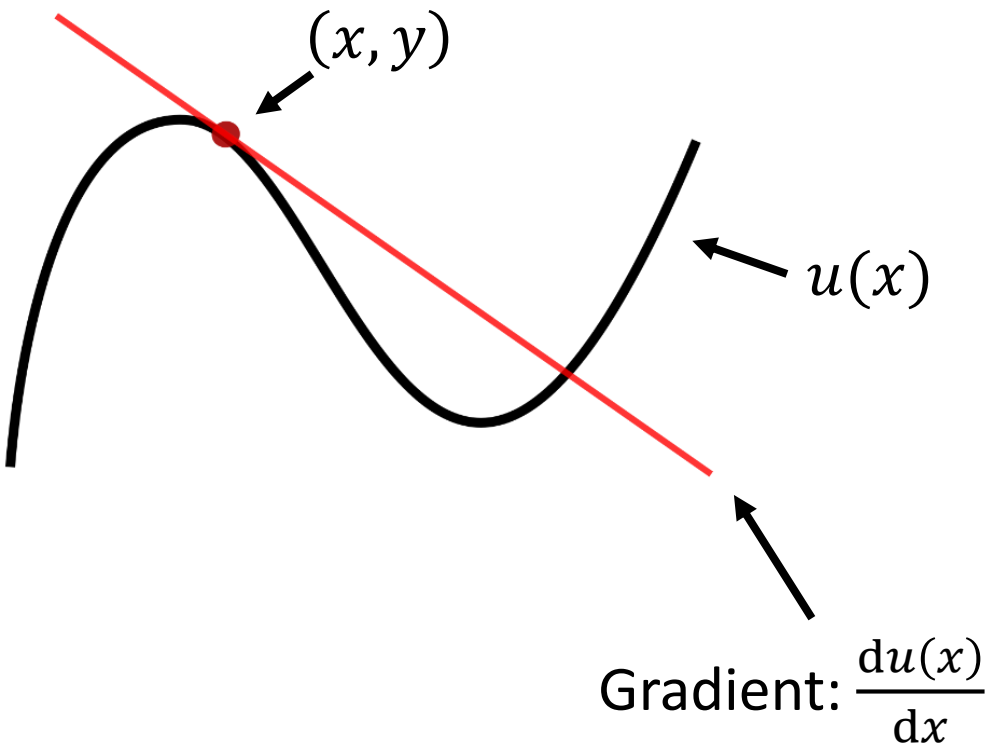
$$\frac{\mathrm{d}f(x)}{\mathrm{d}x} = 3x^2$$

2. If

$$f(x) = 3x^2$$

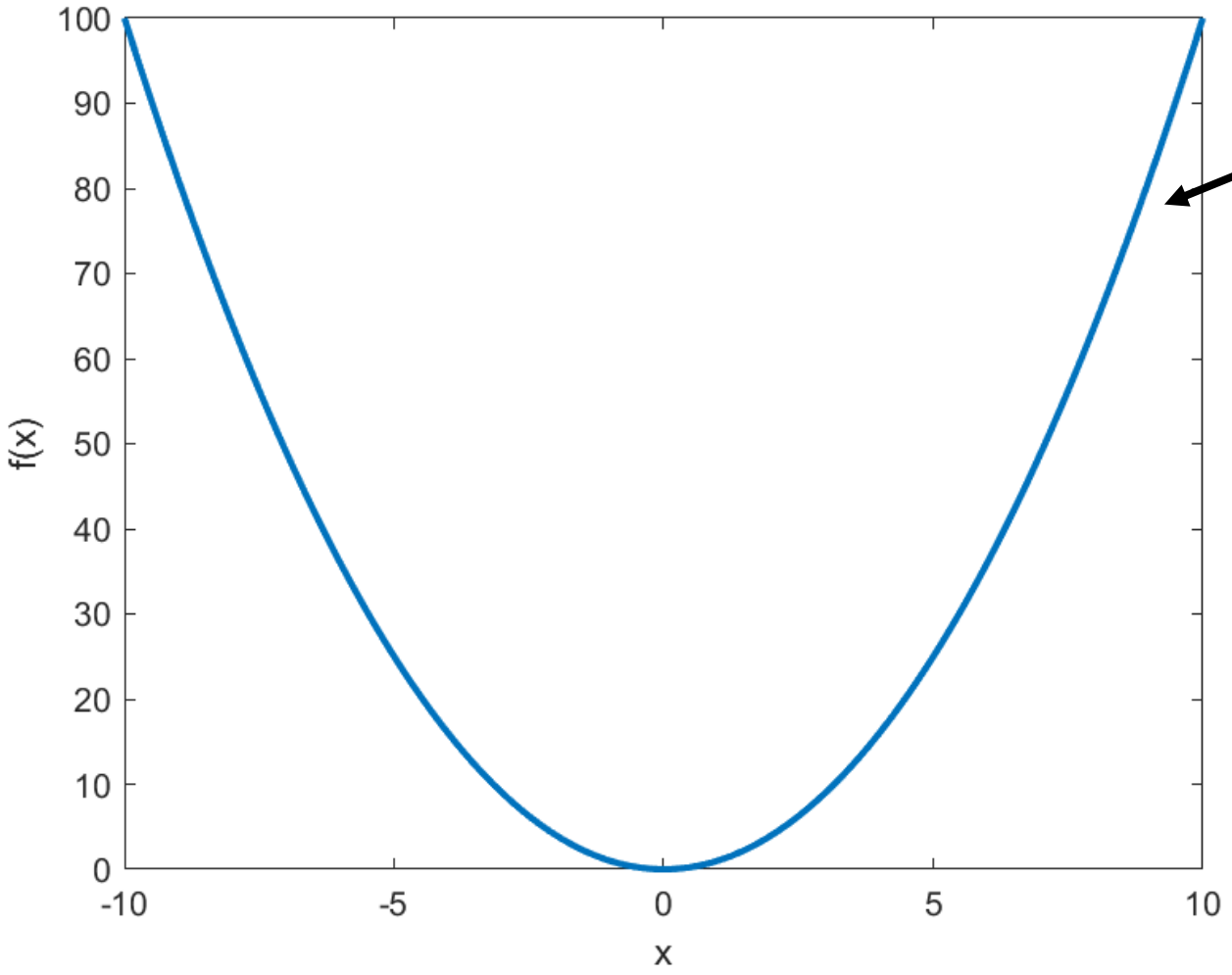
Then

$$\int f(x)\,\mathrm{d}x = x^3 + c$$



Integration

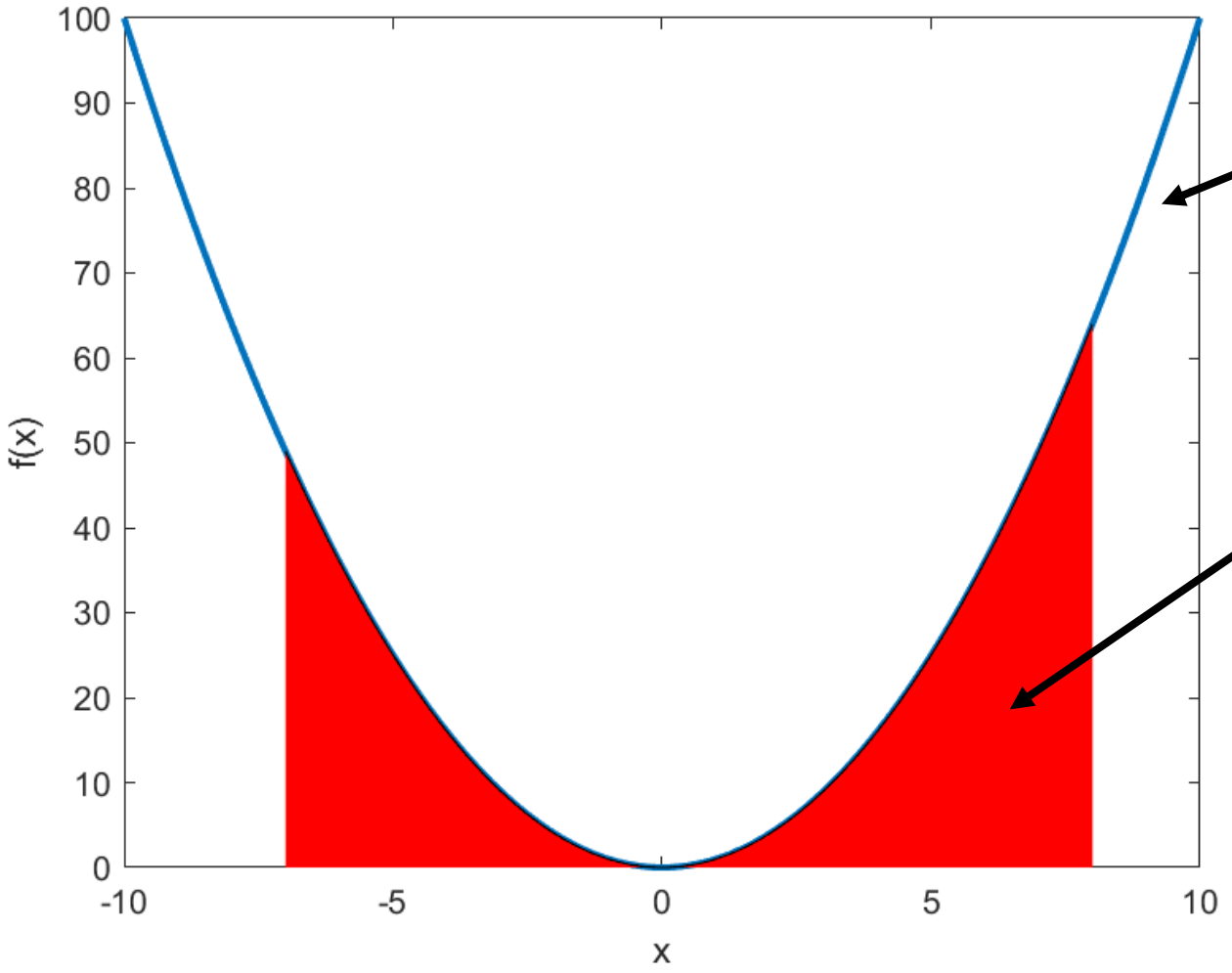
We can view integration as the area under the graph



$f(x) = x^2$

Integration

We can view integration as the area under the graph

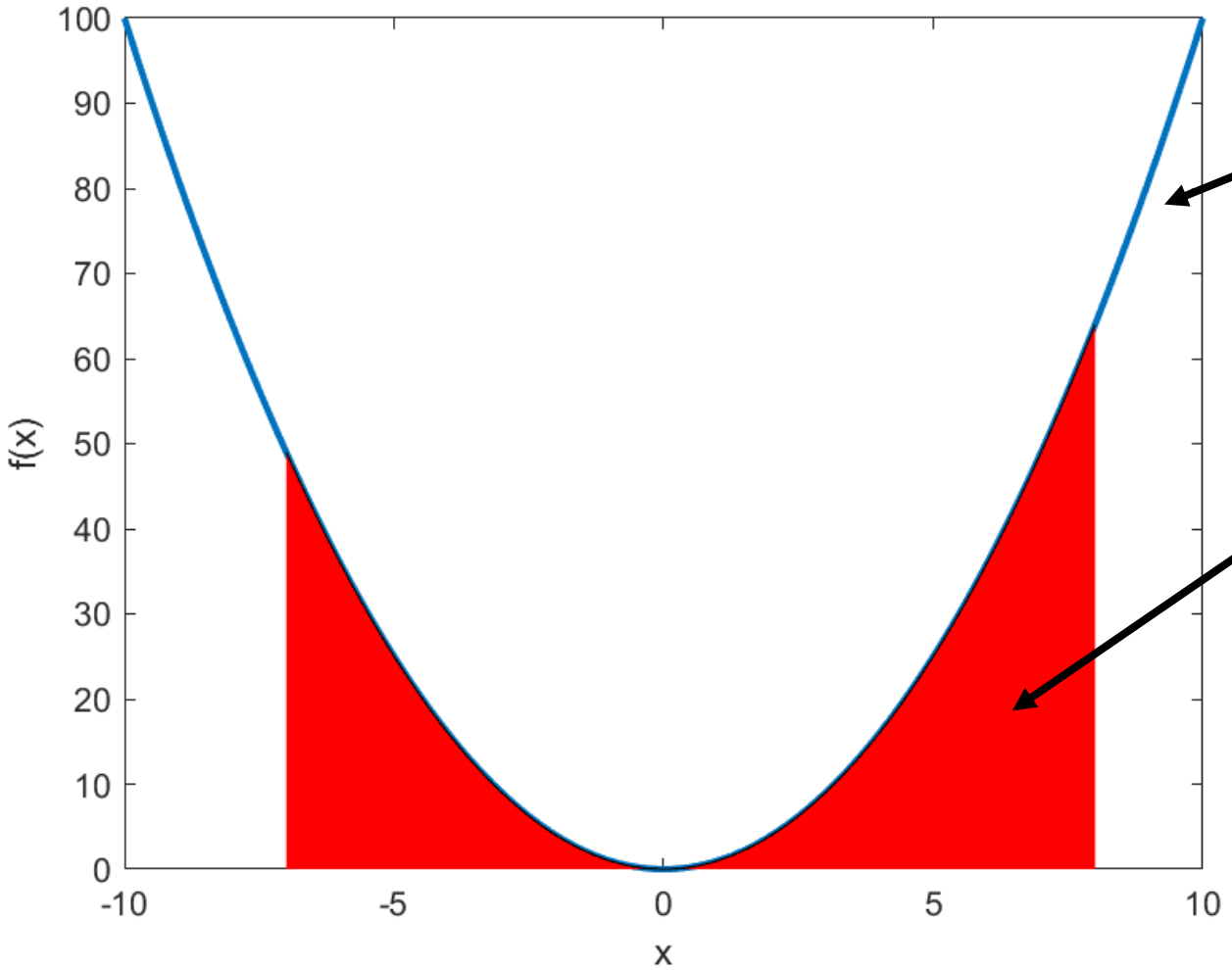


$$f(x) = x^2$$

$$\int_{-7}^{8} f(x)\,\mathrm{d}x$$

Integration

We can view integration as the area under the graph



$$f(x) = x^2$$

$$\int_{-7}^{8} x^2 \, \mathrm{d}x = \left[\frac{x^3}{3}\right]_{-7}^{8} = \frac{8^3}{3} - \frac{(-7)^3}{3}$$
$$= 285$$

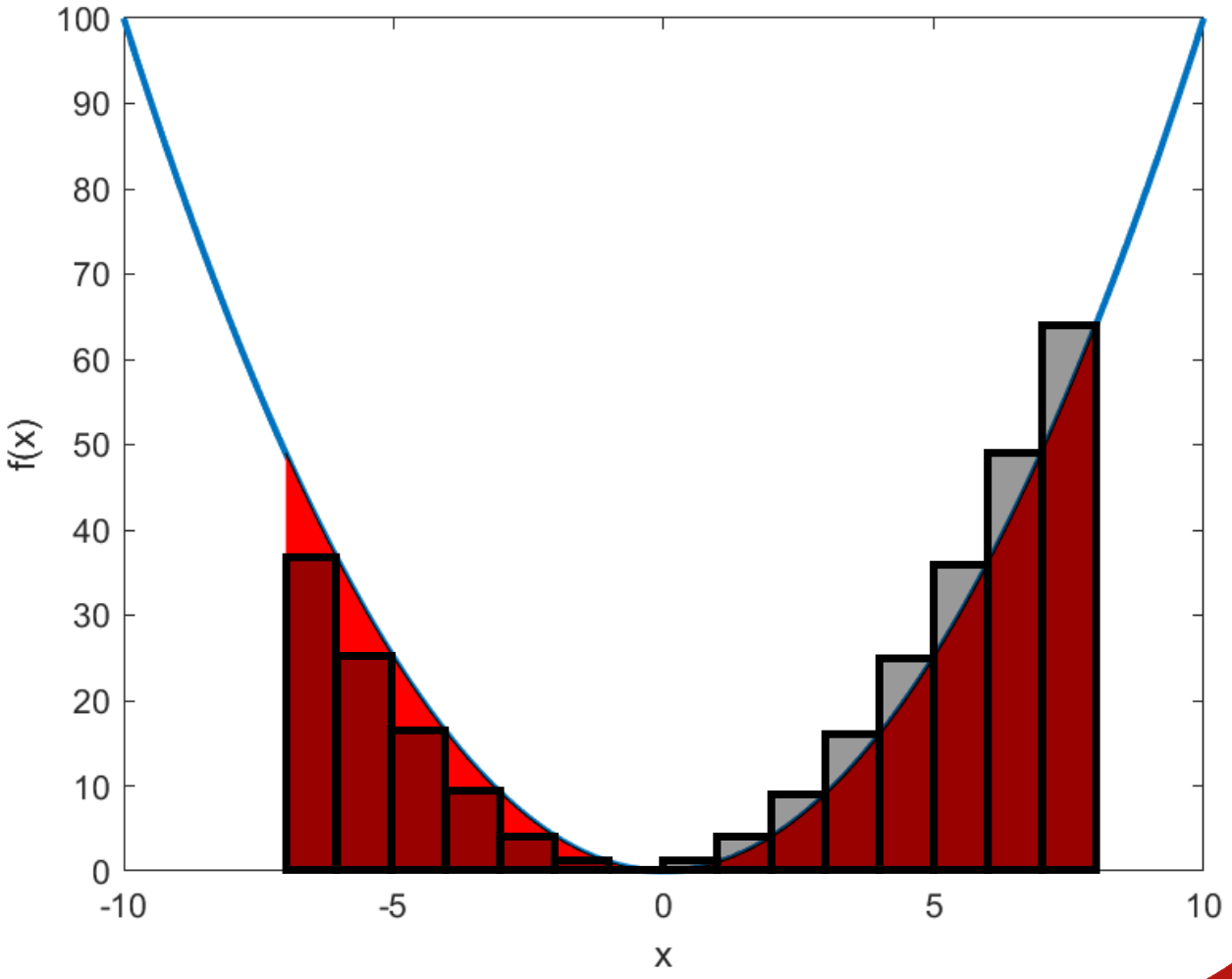
NB: you don't need to know how to integrate functions for this module

Integration

BUT… all we know is $f: \mathbb{Z}^p \to \mathbb{R}$

i.e. we know the values of f but not the equation

So, we could approximate the integral with **areas of rectangles**, which gives 295



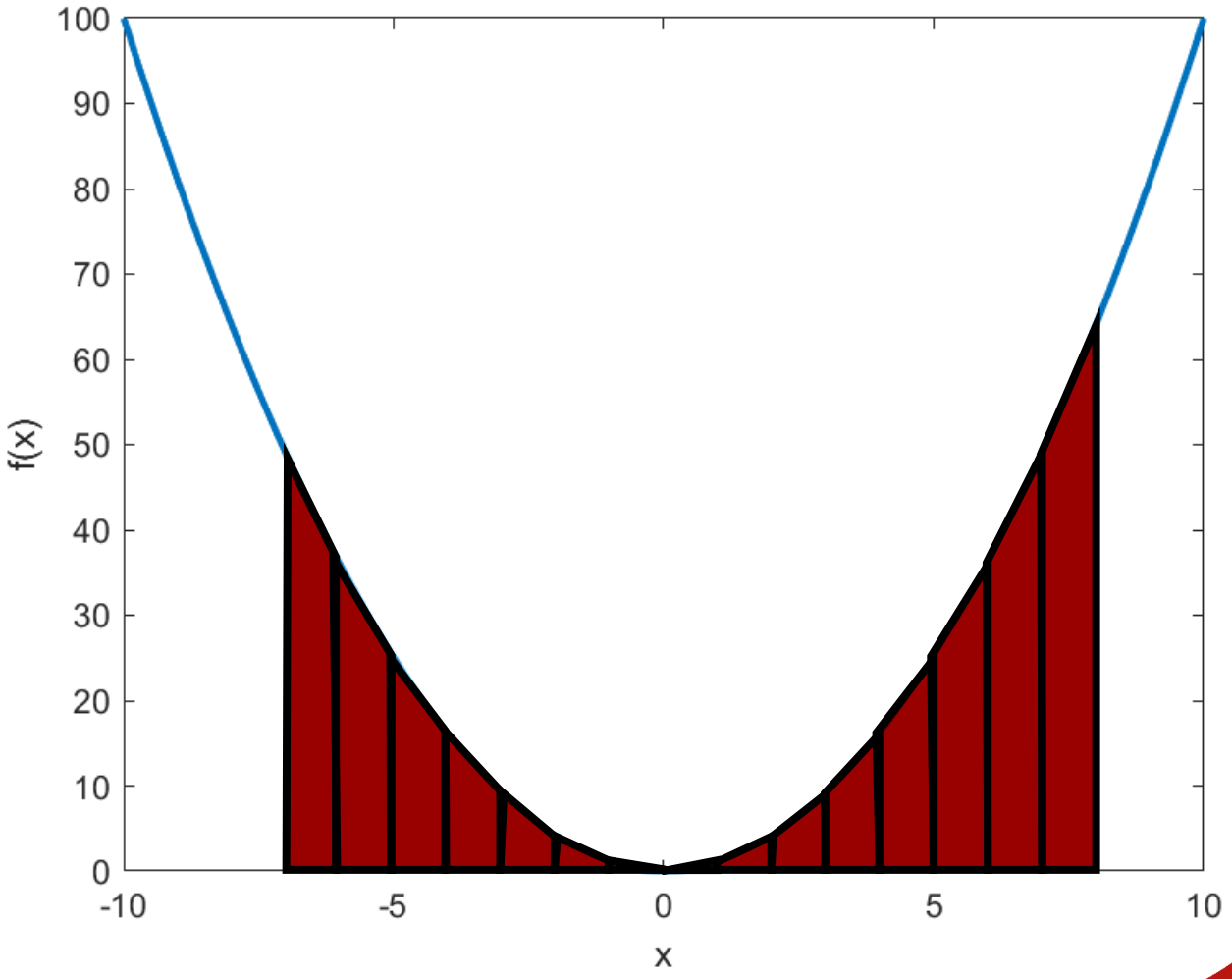
Integration

BUT… all we know is $f: \mathbb{Z}^p \to \mathbb{R}$

i.e. we know the values of f but not the equation

So, we could approximate the integral with **areas of rectangles**, which gives 295

For a better approximation, we can use **trapeziums**, which gives 287.5



Integration

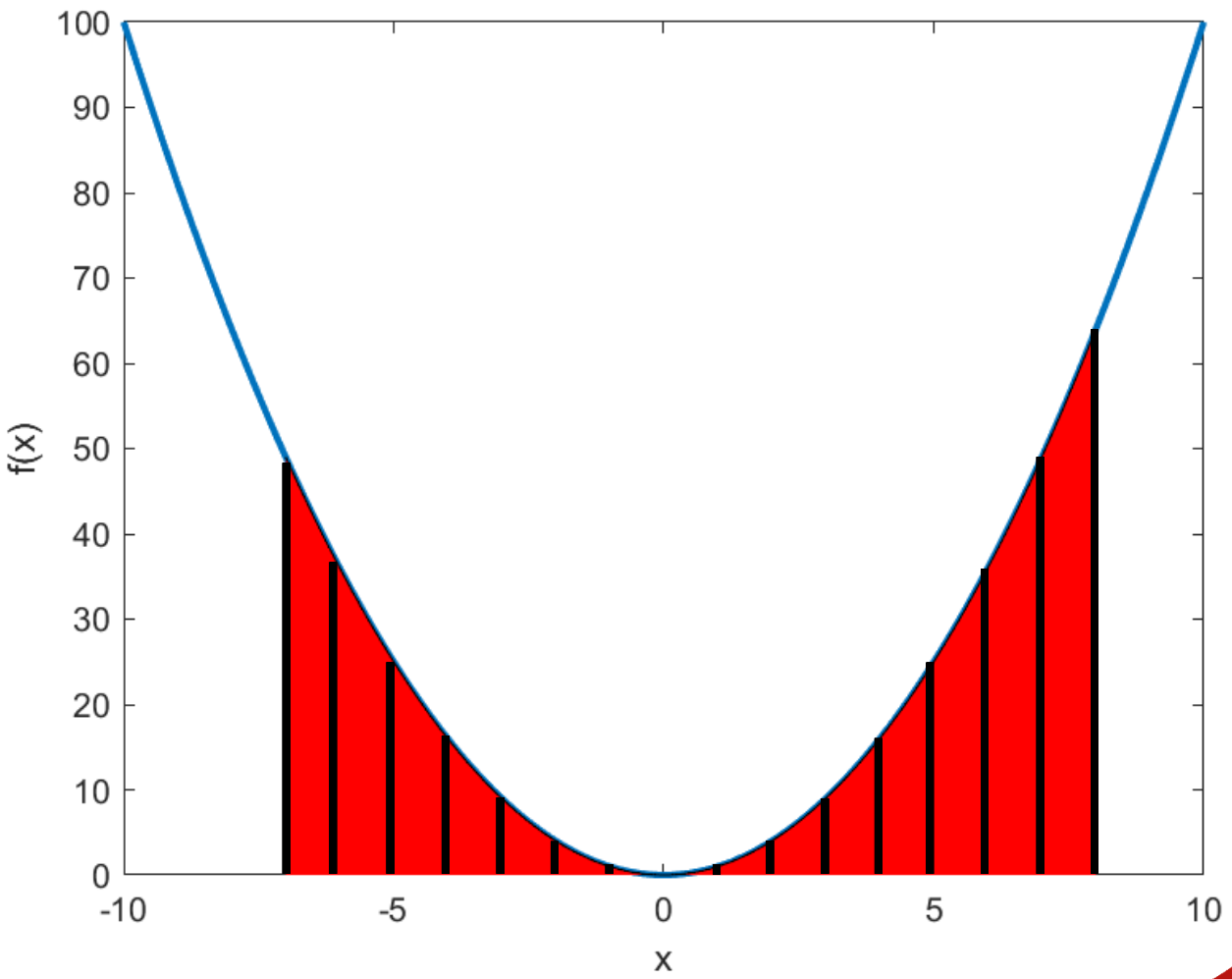
BUT… all we know is $f: \mathbb{Z}^p \rightarrow \mathbb{R}$

i.e. we know the values of f but not the equation.

BUT… we are mapping from **integers**, so we can only use this information.

So, our approximation becomes:

$$\sum_{x=-7}^{8} f(x)$$



Integration

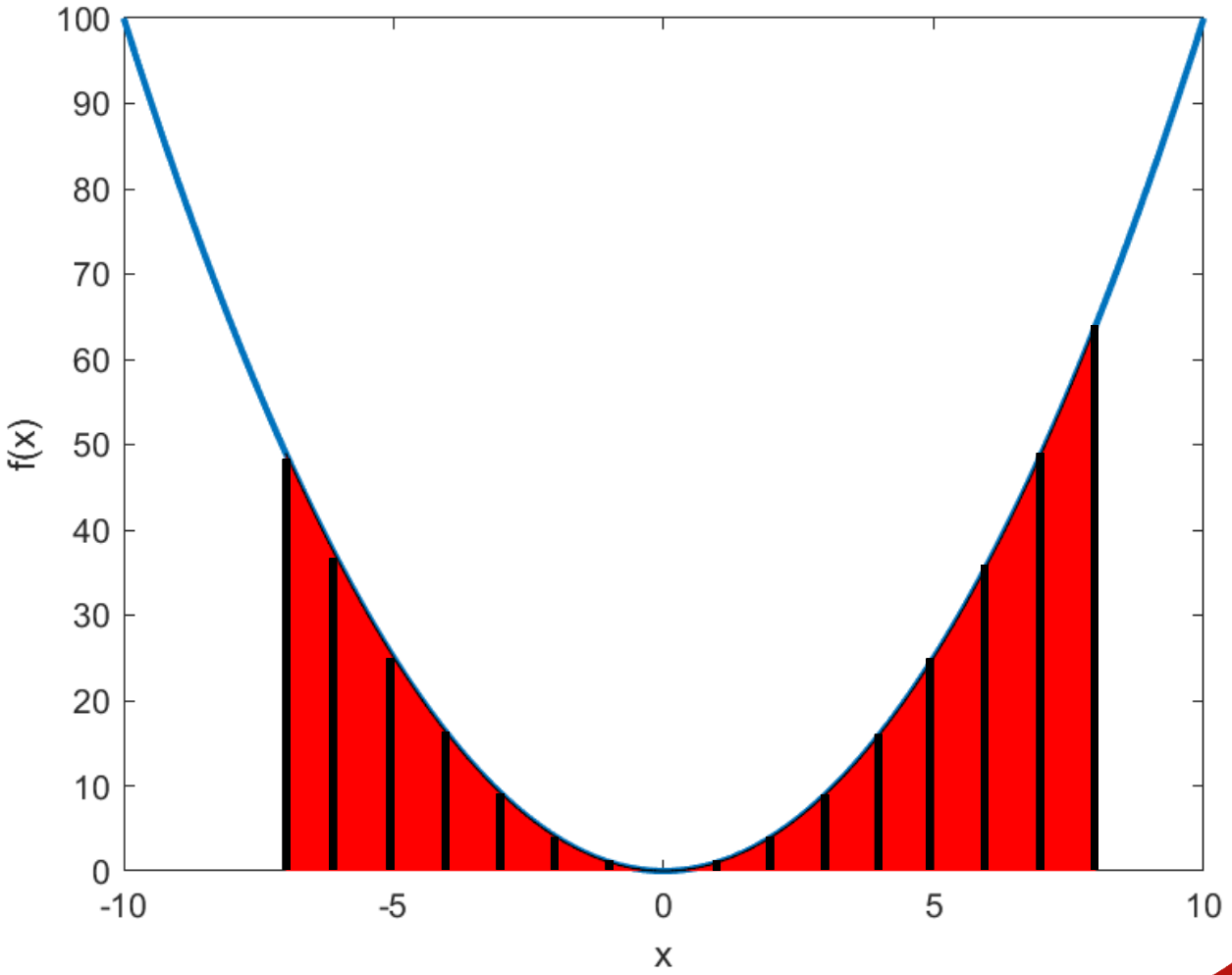
BUT… all we know is $f: \mathbb{Z}^p \to \mathbb{R}$

i.e. we know the values of f but not the equation.

BUT… we are mapping from **integers**, so we can only use this information.

So, our approximation becomes:

$$\int_{-7}^{8} f(x)\,\mathrm{d}x \approx \sum_{x=-7}^{8} f(x)$$



Integration

In general, we have

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{x=a}^{b} f(x)$$

For two dimensions:

$$\int_c^d \int_a^b f(x,y)\,\mathrm{d}x\,\mathrm{d}y = \int_c^d \left(\int_a^b f(x,y)\,\mathrm{d}x \right) \mathrm{d}y \approx \sum_{y=c}^{d} \left(\sum_{x=a}^{b} f(x,y) \right) = \sum_{y=c}^{d} \sum_{x=a}^{b} f(x,y)$$

Integration

In general, we have

$$\int_a^b f(x)\,\mathrm{d}x \approx \sum_{x=a}^{b} \mathbf{f}[x]$$

For two dimensions:

$$\int_c^d \int_a^b f(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \sum_{y=c}^{d} \sum_{x=a}^{b} \mathbf{f}[x,y]$$

Convolution

Real Domain:

$$(k * u)(x_2, y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} k(x_1, y_1)\, u(x_2 - x_1, y_2 - y_1) \mathrm{d}y_1\, \mathrm{d}x_1$$

Discrete Domain:

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{i_1=-\infty}^{\infty} \sum_{j_1=-\infty}^{\infty} \mathbf{k}[x_1, y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

How do we deal with ∞ in practice?

Boundary Conditions

Discrete Domain:

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{i_1=-\infty}^{\infty} \sum_{j_1=-\infty}^{\infty} \mathbf{k}[x_1, y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

How do we deal with ∞ in practice?

Dirichlet boundary condition: Assume that the values outside the image and kernel are 0.

Then, if our kernel matrix is of size $m \times n$, we have

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{i_1=1}^{m} \sum_{j_1=1}^{n} \mathbf{k}[x_1, y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

Transposed Kernel

It can be easier to transpose the kernel. I.e. if our kernel matrix $\mathbf{k}$ is of size $(2m+1) \times (2n+1)$, the transpose $\mathbf{k}^\top$ is defined by

$$\mathbf{k}^\top[i,j] = \mathbf{k}[-i,-j] \text{ for } -m \leq i \leq m \text{ and } -n \leq j \leq n$$

Our convolution equation becomes

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{i_1=-m}^{m} \sum_{j_1=-n}^{n} \mathbf{k}^\top[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

Boundary Conditions

If our kernel matrix is of size $(2m+1) \times (2n+1)$, our convolution equation is

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^{\top}[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

We can think of this as a nested *for* loop:

```
kt = k';
tmp = 0;
for x1 = -m:m
    for y1 = -n:n
        tmp = tmp + kt(-x1,-y1)*u(x2-x1,y2-y1);
    end
end
conv(x2,y2) = tmp;
```

Boundary Conditions

If our kernel matrix is of size $(2m+1) \times (2n+1)$, our convolution equation is

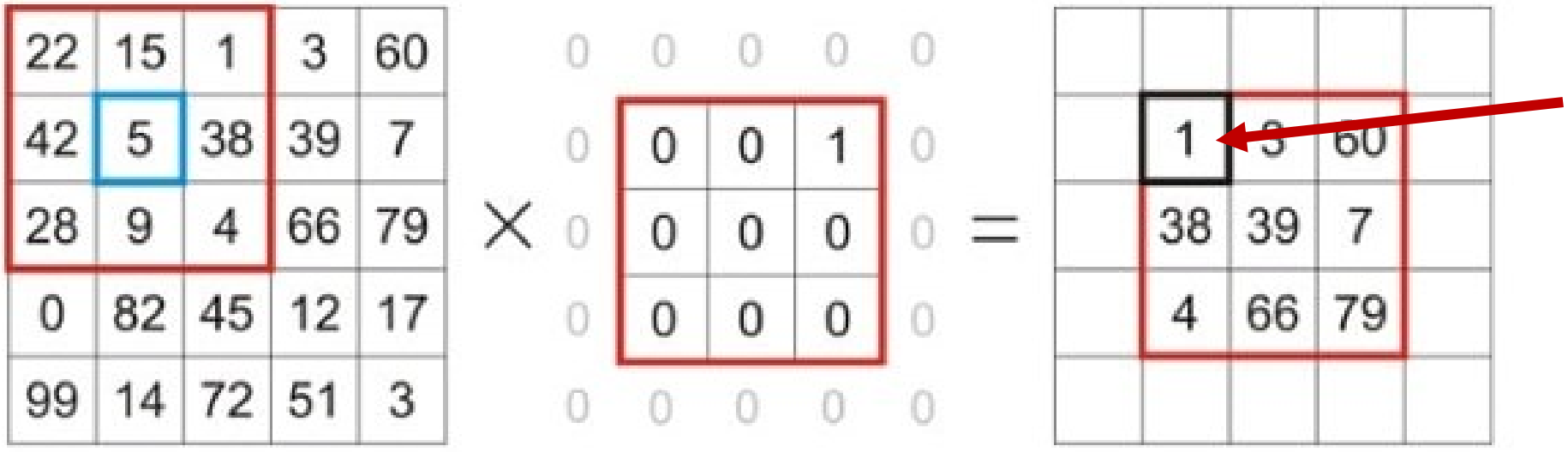
$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^\top[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

Or, even simpler:

```
kt = k';
tmp = 0;
conv(x2,y2) = sum(kt.*u(x2-m:x2+m,y2-n:y2+n),'all');
```

Convolution in Practice

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^{\top}[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$



Convolution in Practice

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^\top[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

$(x_2 - x_1, y_2 - y_1)$
$= (1,1)$

$(-x_1, -y_1)$
$= (-1, -1)$

$$\begin{bmatrix} 22 & 15 & 1 & 3 & 60 \\ 42 & 5 & 38 & 39 & 7 \\ 28 & 9 & 4 & 66 & 79 \\ 0 & 82 & 45 & 12 & 17 \\ 99 & 14 & 72 & 51 & 3 \end{bmatrix} \times \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} & & & & \\ & 1 & 3 & 60 & \\ & 38 & 39 & 7 & \\ & 4 & 66 & 79 & \\ & & & & \end{bmatrix}$$

$(\mathbf{k} * \mathbf{u})(2,2)$

$\mathbf{u}$ $\qquad$ $\mathbf{k}^\top$ $\qquad$ $\mathbf{k} * \mathbf{u}$

Convolution in Practice

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^\top[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

$(x_2 - x_1, y_2 - y_1)$
$= (1,2)$

$(-x_1, -y_1)$
$= (-1,0)$

$$\underbrace{\begin{bmatrix} 22 & 15 & 1 & 3 & 60 \\ 42 & 5 & 38 & 39 & 7 \\ 28 & 9 & 4 & 66 & 79 \\ 0 & 82 & 45 & 12 & 17 \\ 99 & 14 & 72 & 51 & 3 \end{bmatrix}}_{\mathbf{u}} \times \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{k}^\top} = \underbrace{\begin{bmatrix} & & & & \\ & 1 & 3 & 60 & \\ & 38 & 39 & 7 & \\ & 4 & 66 & 79 & \\ & & & & \end{bmatrix}}_{\mathbf{k} * \mathbf{u}}$$

$(\mathbf{k} * \mathbf{u})(2,2)$

Convolution in Practice

$$(\mathbf{k} * \mathbf{u})[x_2, y_2] = \sum_{x_1=-m}^{m} \sum_{y_1=-n}^{n} \mathbf{k}^\top[-x_1, -y_1]\, \mathbf{u}[x_2 - x_1, y_2 - y_1]$$

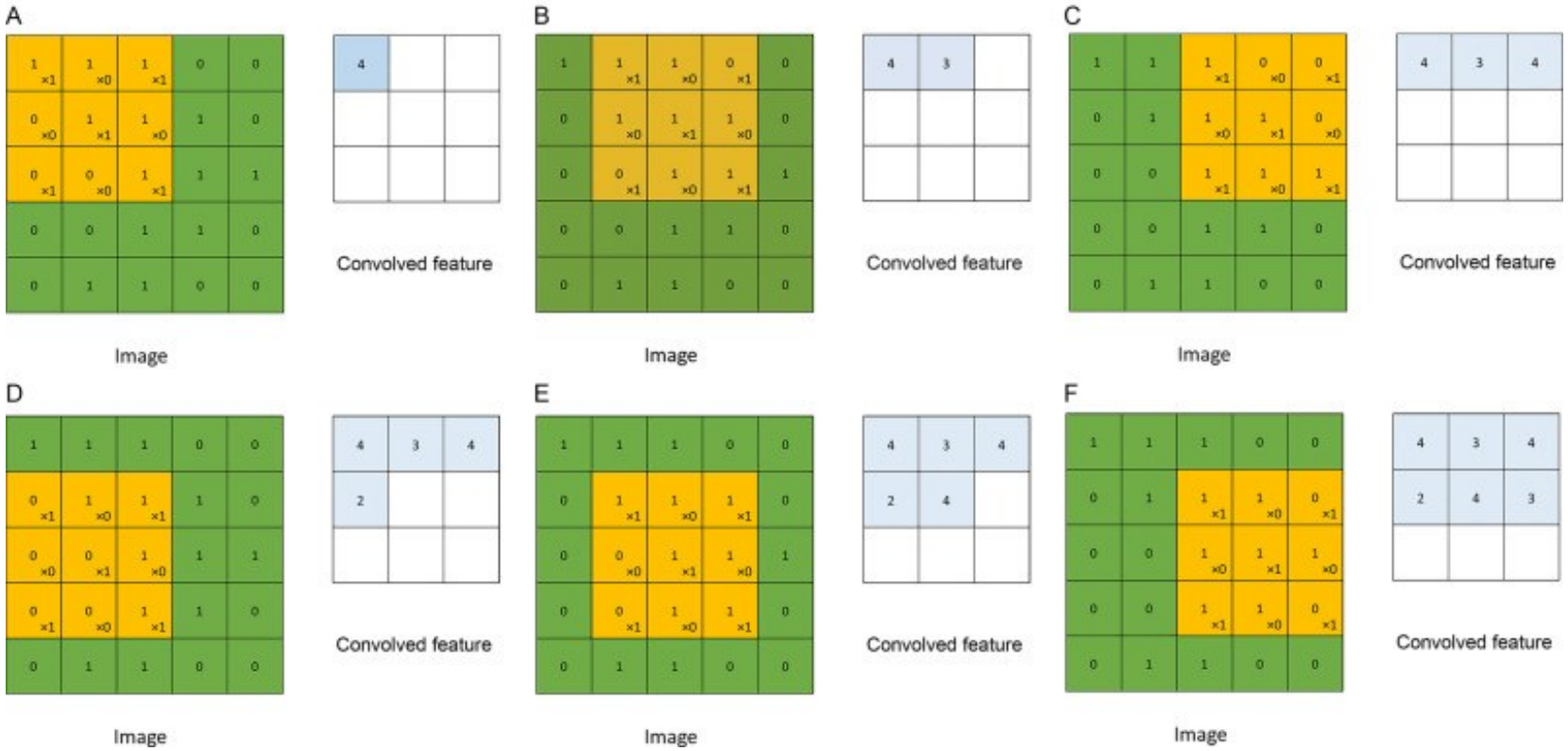
$(x_2 - x_1, y_2 - y_1)$
$= (1,3)$

$(-x_1, -y_1)$
$= (-1,1)$

$$\underbrace{\begin{bmatrix} 22 & 15 & 1 & 3 & 60 \\ 42 & 5 & 38 & 39 & 7 \\ 28 & 9 & 4 & 66 & 79 \\ 0 & 82 & 45 & 12 & 17 \\ 99 & 14 & 72 & 51 & 3 \end{bmatrix}}_{\mathbf{u}} \times \underbrace{\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{k}^\top} = \underbrace{\begin{bmatrix} & & & & \\ & 1 & 3 & 60 & \\ & 38 & 39 & 7 & \\ & 4 & 66 & 79 & \\ & & & & \end{bmatrix}}_{\mathbf{k} * \mathbf{u}}$$

$(\mathbf{k} * \mathbf{u})(2,2)$

Convolution in Practice



Properties of Convolution

Commutativity: $k * u = u * k$ - does not apply with computer functions

Associativity: $k * (h * u) = (k * h) * u$

Distributivity: $u * (k + h) = u * k + u * h$

Associativity with scalar multiplication: $a(k * u) = (ak) * u$ where $a \in \mathbb{C}$ (complex numbers)

Integration:

$$\int_{-\infty}^{\infty} (k * u)(x)\,\mathrm{d}x = \left(\int_{-\infty}^{\infty} k(x)\,\mathrm{d}x\right)\left(\int_{-\infty}^{\infty} u(x)\,\mathrm{d}x\right)$$

Properties of Convolution

Proof of integration relation – for interest only

Let $s = x - y$; then $\frac{\mathrm{d}s}{\mathrm{d}x} = 1$. We have

$$\int_{-\infty}^{\infty} (k * u)(x)\,\mathrm{d}x = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(y)u(x-y)\mathrm{d}y\,\mathrm{d}x$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} k(y)u(s)\mathrm{d}y\,\mathrm{d}s = \int_{-\infty}^{\infty} k(y)\int_{-\infty}^{\infty} u(s)\mathrm{d}y\,\mathrm{d}s$$

$$= \int_{-\infty}^{\infty} k(y)\,\mathrm{d}y\int_{-\infty}^{\infty} u(s)\,\mathrm{d}s = \left(\int_{-\infty}^{\infty} k(x)\,\mathrm{d}x\right)\left(\int_{-\infty}^{\infty} u(x)\,\mathrm{d}x\right)$$

Properties of Convolution

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(k * u)(x,y)\,\mathrm{d}x\,dy = \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}k(x,y)\,\mathrm{d}x\,\mathrm{d}y\right)\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}u(x,y)\,\mathrm{d}x\,\mathrm{d}y\right)$$

$$\approx \sum_{x,y=-\infty}^{\infty}(\mathbf{k} * \mathbf{u})[x,y] = \left(\sum_{x,y=-\infty}^{\infty}\mathbf{k}[x,y]\right)\left(\sum_{x,y=-\infty}^{\infty}\mathbf{u}[x,y]\right)$$

$$\sum k = 0$$

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix} * \mathbf{A}$$

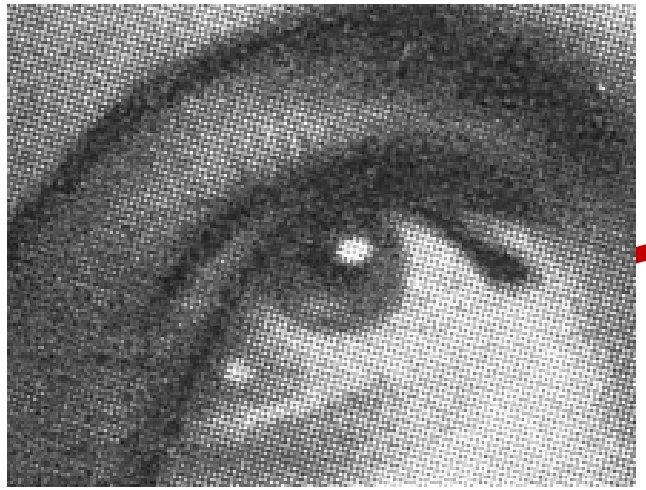
$$\begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * \mathbf{A}$$

Properties of Convolution

$$\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}(k * u)(x,y)\,\mathrm{d}x\,dy = \left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}k(x,y)\,\mathrm{d}x\,\mathrm{d}y\right)\left(\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}u(x,y)\,\mathrm{d}x\,\mathrm{d}y\right)$$

$$\approx \sum_{x,y=-\infty}^{\infty}(\mathbf{k} * \mathbf{u})[x,y] = \left(\sum_{x,y=-\infty}^{\infty}\mathbf{k}[x,y]\right)\left(\sum_{x,y=-\infty}^{\infty}\mathbf{u}[x,y]\right)$$

$$\sum k = 1$$



$$\begin{bmatrix} \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \\ \frac{1}{9} & \frac{1}{9} & \frac{1}{9} \end{bmatrix} * a$$

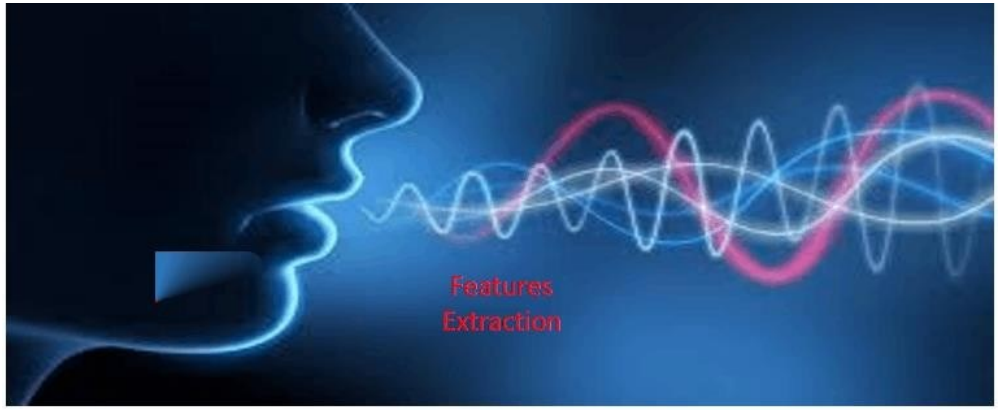
This Week

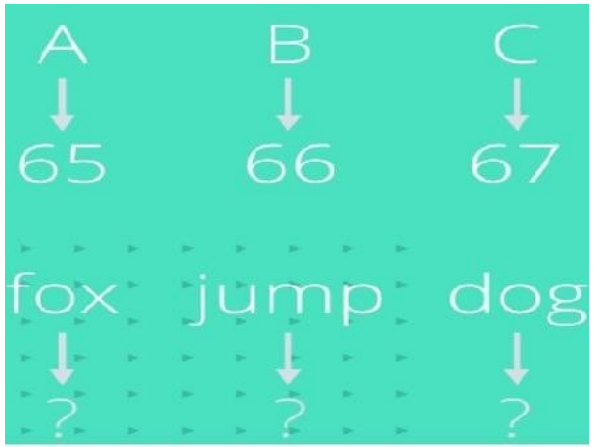
Computer Vision

Natural Language Processing

Feature Extraction from Text

- How do we extract features from text?
- Text data = a sequence of words
- Words are made up of characters
- Characters are represented with ASCII or Unicode
- The machine do not really make any meaning of from that
- **So, how do we encode meaning representations from text?**





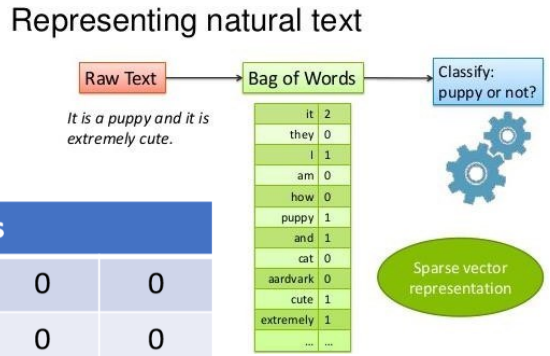
Vector Representation

Common approaches

- Localist approach (1-Hot Vector)
- Bag of Word (BoW) Model
  - TF-IDF, PPMI
- Word Embedding
  - Word2Vec
  - Glove

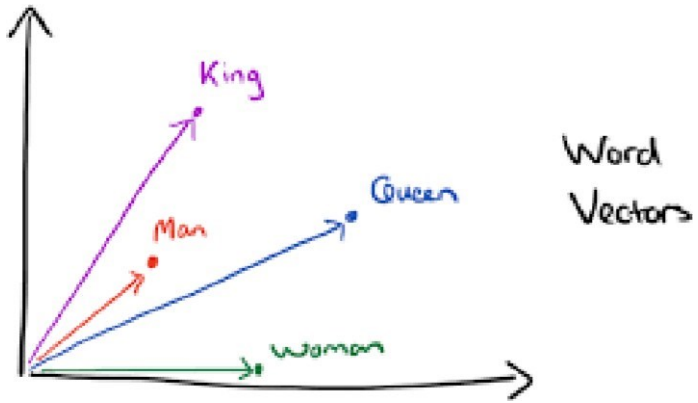
Similarity (distance) measure

- Cosine distance
- Euclidean



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

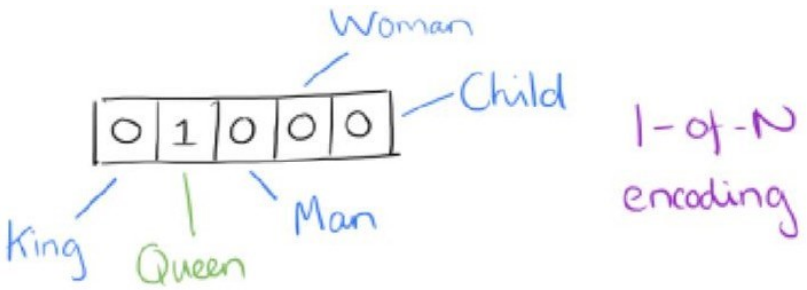
$$\sqrt{\frac{\sum_{i=1}^{n} x_i \cdot y_i}{\|x\| \cdot \|y\|}}$$



1-Hot Encoding

Localist representation

- Vocabulary: [king, queen, man, woman, child]
- Each word is assigned a vector
- 1: in the position of word
- 0: everywhere else



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

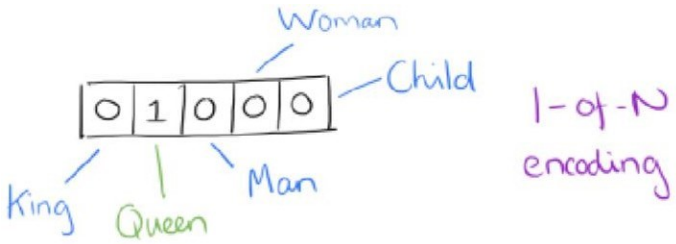
1 – Hot Encoding

Localist representation

- Vocabulary: [king, queen, man, woman, child]
- Each word is assigned a vector
- 1: in the position of word
- 0: everywhere else

Problems

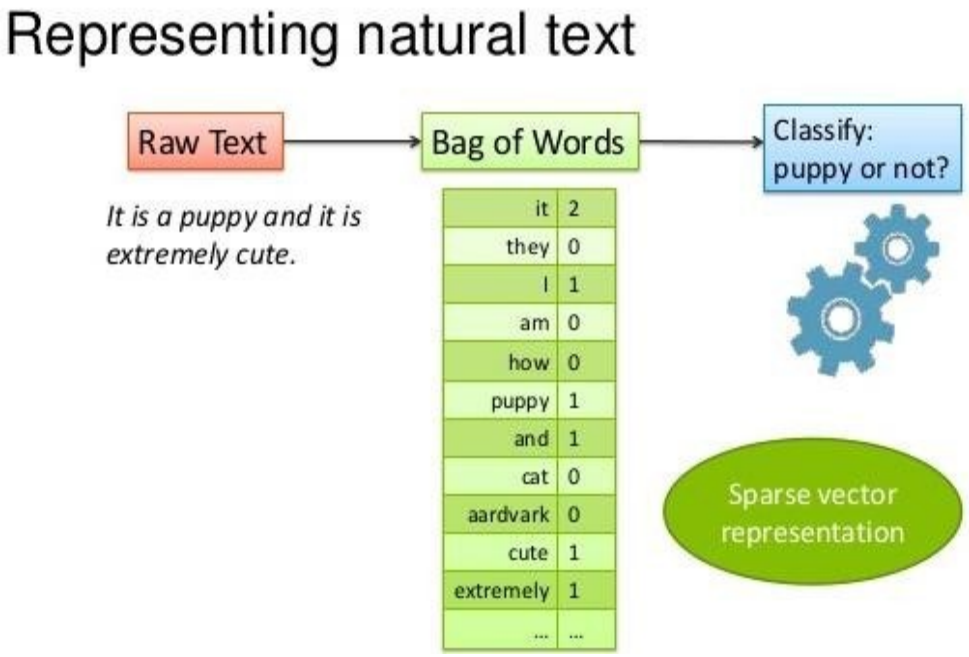
- $|V|$ can be very large 100s of thousands
- Sparse matrix, shape $|V| \times |V|$
  - Memory and compute inefficient
- Orthogonal vectors
  - $similarity(\text{'}king\text{'}, \text{'}queen\text{'}) = 0$



word	features				
king	1	0	0	0	0
queen	0	1	0	0	0
man	0	0	1	0	0
woman	0	0	0	1	0
child	0	0	0	0	1

Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence



Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.
```

Bag of Words (BoW)

- Each document is an unordered collection of words
- Document: a unit of text to be analysed
  - A tweet
  - A book
  - An email
  - A sentence etc
- BoW: The set of words in the document and their frequency of occurrence

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.
```

Bag of Words (BoW)

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.

BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"Mary":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

If we have a union of documents e.g.

```
(3) John likes to watch movies. Mary likes movies too. Mary also likes to watch football games.
```

what will the counts for the BoW model be?

```
BoW3 = {"John":[ ],"likes":[ ],"to":[ ],"watch":[ ],"movies":[ ],"Mary":[ ],"too":[ ],"also":[ ],"football":[ ],"games":[ ]};
```

Bag of Words (BoW)

```
(1) John likes to watch movies. Mary likes movies too.

(2) Mary also likes to watch football games.
```

```
BoW1 = {"John":1,"likes":2,"to":1,"watch":1,"movies":2,"Mary":1,"too":1};
BoW2 = {"Mary":1,"also":1,"likes":1,"to":1,"watch":1,"football":1,"games":1};
```

If we have a union of documents e.g.

```
(3) John likes to watch movies. Mary likes movies too. Mary also likes to watch football games.
```

what will the counts for the BoW model be?

```
BoW3 = {"John":1,"likes":3,"to":2,"watch":2,"movies":2,"Mary":2,"too":1,"also":1,"football":1,"games":1};
```

Bag of Words (BoW)

- Advantage
  - Better similarity representation than 1-hot
  - Can be extended to n-grams
- Disadvantage
  - Most common words dominate the counts
    - the, a, is
  - High dimensional vectors
- Solution
  - TF-IDF
  - PPMI
  - Word embedding

```
[
    "John likes",
    "likes to",
    "to watch",
    "watch movies",
    "Mary likes",
    "likes movies",
    "movies too",
]
```

TF-IDF

- Term Frequency – Inverse Document Frequency
  - For a corpus of documents D:
    - *Term Frequency* (TF)
      - Frequency counts (log transformed)
      - $TF_{t,d} = \begin{cases} \mathbf{1 + \log_{10} c(t,d)} & \text{if } c(t,d)>0 \\ \mathbf{0} & \text{otherwise} \end{cases}$
    - *Inverse Document Frequency* (IDF)
      - $|D|$ = # all documents
      - $|\{d \in D : t \in d\}|$ = # documents with t
      - $IDF_t = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$
      - Words like 'the' or '*of*' have low IDF
    - **TF-IDF**: $TF \times IDF$

	Document 1	Document 2	Document 3	Document 4	Document 5	Document 6	Document 7	Document 8
Term(s) 1	10	0	1	0	0	0	0	2
Term(s) 2	0	2	0	0	0	18	0	2
Term(s) 3	0	0	0	0	0	0	0	2
Term(s) 4	6	0	0	4	6	0	0	0
Term(s) 5	0	0	0	0	0	0	0	2
Term(s) 6	0	0	1	0	0	1	0	0
Term(s) 7	0	1	8	0	0	0	0	0
Term(s) 8	0	0	0	0	0	3	0	0

Word Vector (Passage Vector)

Document Vector

PPMI

PPMI

- Pointwise Mutual Information
  - *Do x and y co-occur more than if they were independent?*
  - PMI = $\log_2 \frac{P(x,y)}{P(x) \cdot P(y)}$
- PMI range: $-\infty$ to $+\infty$
- -ves are problematic
  - Use Positive PMI (PPMI)
  - PPMI $\max(\log_2 \frac{P(x,y)}{P(x) \cdot P(y)}, 0)$

Count(w,context)

	computer	data	pinch	result	sugar
apricot	0	0	1	0	1
pineapple	0	0	1	0	1
digital	2	1	0	1	0
information	1	6	0	4	0

p(w,context) **p(w)**

	computer	data	pinch	result	sugar	
apricot	0.00	0.00	0.05	0.00	0.05	0.11
pineapple	0.00	0.00	0.05	0.00	0.05	0.11
digital	0.11	0.05	0.00	0.05	0.00	0.21
information	0.05	0.32	0.00	0.21	0.00	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

PPMI(w,context)

	computer	data	pinch	result	sugar
apricot	-	-	2.25	-	2.25
pineapple	-	-	2.25	-	2.25
digital	1.66	0.00	-	0.00	-
information	0.00	0.57	-	0.47	-

Extracting More Context

Extracting more context from text

The complex houses married and single soldiers and their families.

The horse raced past the barn fell.

The man the professor the student has studies Rome.

Time flies like an arrow, but fruit flies like a banana.

Ambiguity

Question:

- How many senses does *spring* have?

Answer:

a) 3

b) 4

c) 5

d) 31

Ambiguity

Question:

- How many senses does *spring* have?

Answer:

a) 3

b) 4

c) 5

d) 31

spring (sprɪŋ) *vb* **springs, springing, sprang** *or* **sprung; sprung. 1** to move or cause to move suddenly upwards or forwards in a single motion. **2** to release or be released from a forced position by elastic force: *the bolt sprang back*. **3** (*tr*) to leap or jump over. **4** (*intr*) to come, issue, or arise suddenly. **5** (*intr*) (of a part of a mechanism, etc.) to jump out of place. **6** to make (wood, etc.) warped or split or (of wood, etc.) to become warped or split. **7** to happen or cause to happen unexpectedly: *to spring a surprise; the boat sprung a leak*. **8** (*intr*) to develop or originate: *the idea sprang from a chance meeting*. **9** (*intr;* usually foll. by *from*) to be descended: *he sprang from peasant stock*. **10** (*intr;* often foll. by *up*) to come into being or appear suddenly: *factories springing up*. **11** (*tr*) (of a gun dog) to rouse (game) from cover. **12** (*intr*) (of game or quarry) to start or rise suddenly from cover. **13** (*intr*) to appear to have a strong upward movement: *the beam springs away from the pillar*. **14** to explode (a mine) or (of a mine) to explode. **15** (*tr*) to provide with a spring or springs. **16** (*tr*) *Informal.* to arrange the escape of (someone) from prison. **17** (*intr*) *Archaic or poetic.* (of daylight or dawn) to begin to appear. ◆ *n* **18** the act or an instance of springing. **19** a leap, jump, or bound. **20a** the quality of resilience; elasticity. **20b** (*as modifier*): *spring steel*. **21** the act or an instance of moving rapidly back from a position of tension. **22a** a natural outflow of ground water, as forming the source of a stream. **22b** (*as modifier*): *spring water*. **23a** a device, such as a coil or strip of steel, that stores potential energy when it is compressed, stretched, or bent and releases it when the restraining force is removed. **23b** (*as modifier*): *a spring mattress*. **24** a structural defect such as a warp or bend. **25a** (*sometimes cap.*) the season of the year between winter and summer, astronomically from the March equinox to the June solstice in the N hemisphere and from the September equinox to the December solstice in the S hemisphere. **25b** (*as modifier*): *spring showers*. Related adj: **vernal. 26** the earliest or freshest time of something. **27** a source or origin. **28** one of a set of strips of rubber, steel,

etc., running down the inside of the handle of a cricket bat, hockey stick, etc. **29** Also called: **spring line.** *Nautical.* a mooring line, usually one of a pair that cross amidships. **30** a flock of teal. **31** *Architect.* another name for **springing.** [Old English *springan;* related to Old Norse *springa,* Old High German *springan,* Sanskrit *sprhayati* he desires, Old Slavonic *pragu* grasshopper] ▸ **'springless** *adj* ▸ **'spring,like** *adj*

spring balance *or esp. U.S.* **spring scale** *n* a device in which an object to be weighed is attached to the end of a helical spring, the extension of which indicates the weight of the object on a calibrated scale.

Semantic Analysis and AI

Dictionary definitions might be too fine-grained?
i.e. we might not need 31 definitions for ‘spring’

Bank:

- (n) Financial institution
- (n) Building used by such an institution
- (n) Long raised mass of earth; mound; ridge
- (v) Cause an airplane to tip while turning

Grammatical POS Tagging

Also known as morphosyntactic annotation

- Assign word-class labels
  - Major parts of speech (noun, verb, preposition, etc)
  - Sub-classes (singular vs plural nouns, comparative vs superlative, etc)

Word sequences

the_A

on_I

mat_N

sat_V

cat_N

the_A

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models

The sequence

WORD	The term computer referred to the people who did scientific calculations by hand
POS	A N N V I A N P V J N I N

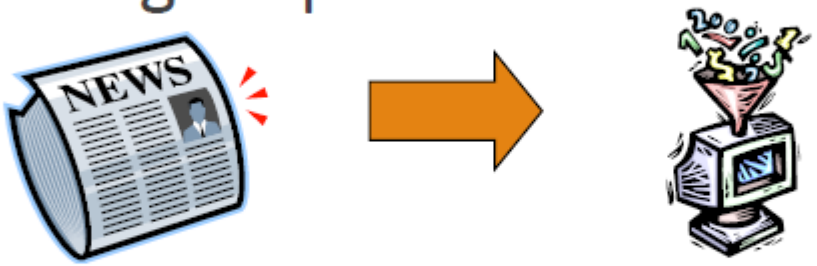
Is much more likely than the sequence

WORD	the by referred who term hand calculations people the computer to did scientific
POS	N P N I A N V A V J N N I

Rule-based POS tagging

STEP 1: Apply most likely tag for each word

Requires training corpus that is manually corrected



e.g. how often does *dogs* occur in a corpus as a noun and how often as a verb? [Unigram probability]

For words not in the training corpus, make a guess based on suffix lists

Rule-based POS tagging

Use immediate context example:

The dogs barked

Noun

He dogs my steps

Verb

Rule-based POS tagging

Examples of patching rule types

Change tag A to tag B when:

- The preceding word is tagged Z
- The following word in tagged Z
- The word two before is tagged Z
- The word two after is tagged Z
- One of the two preceding (following) words is tagged Z
- One of the three preceding (following) words is tagged Z

Rule-based POS tagging

Tagger looks at the application of each rule of this type to the training corpus, and computes the number of errors remedied by its application and the number of new errors introduced

e.g. change tag ‘infinitive marker’ to ‘preposition’ when an ‘article’ follows

- “to the”: *to* can’t really be infinitive

e.g. “If this word is *can* and the previous word is *the*, delete any verb tags”

Rule-based POS tagging

Learning patches iteratively, most successful patches are added to the list in order of success (tags corrected minus tags wrongly changed)

Once training is complete, you can tag a new corpus by applying the unigram tag first then the learnt rules are applied in sequence

Moving on to clustering and classification next week…

Coming up…

Lab: Introduction to Matlab

- Materials on Moodle

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076

Coming up…

Lab: Convolution, Boundary Conditions and Clustering

- Materials on Moodle

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076

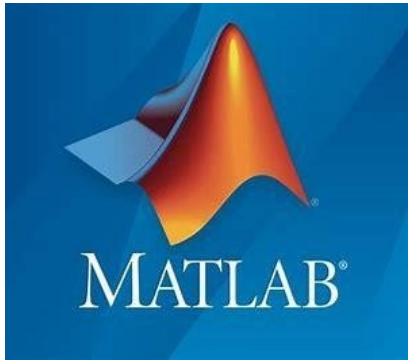
Week 4 Lectures: Clustering and Classification

- Distinguish Classification and Clustering
- Understand implementations and applications

Coursework

Coursework: Task on Moodle

- 20% of Overall Mark
- Feature Extraction and Machine Learning
- Due Friday 18th November 2022 @ 4pm (Week 6)
- Submission on Moodle
- Marks after 3 weeks (Week 9)



Coursework

Computer Vision

Speed sign classification

OR

Natural Language Processing

Tweet classification

sentiment	Content
relief	finally made it to phoenix!
happiness	finally made it to the QC...h
happiness	Finally making some headw
happiness	Finally off work!!!
happiness	Finally on my way home
relief	Finally out of school for tod
relief	finally over my sickness.
happiness	finally packed...thinking its
relief	Finally picked up some han
relief	finally seventeen
surprise	finally shifted from Twhirl t
relief	Finally sleep time
relief	finally some effing internet
happiness	Finally started Twittering....
relief	finally synced iPod

Coursework

Classification Task

1. Data preparation
2. Extracting features
3. Model training and evaluation

Report (2 pages)

1. Introduction
2. Data and Preparation
3. Methodology
4. Results
5. Conclusion

Deadline

4pm Friday, week 6

18 November 2022

Submission on Moodle

20 Marks Total

- Code [10 Marks]
  - Correct Implementation
  - Appropriate Structure
  - Suitable Annotation
- Report
  - 2 Marks per section

SCC361: Artificial Intelligence

Week 4: Clustering and Classification

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

Attendance Check-in

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the “You are checked in” confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Lecture Plan

Weeks 1-5

1. Introduction to Artificial Intelligence and Machine Learning
2. Features in Machine Learning and Feature Extraction
3. Computer Vision and Natural Language Processing
4. **Clustering and Classification**
5. Artificial Neural Networks

Weeks 6-10:

6. Genetic Algorithms
7. Naïve Bayesian Classifier
8. Decision Tree Classifier
9. Introduction to Deep Neural Networks
10. Introduction to Convolutional Neural Networks

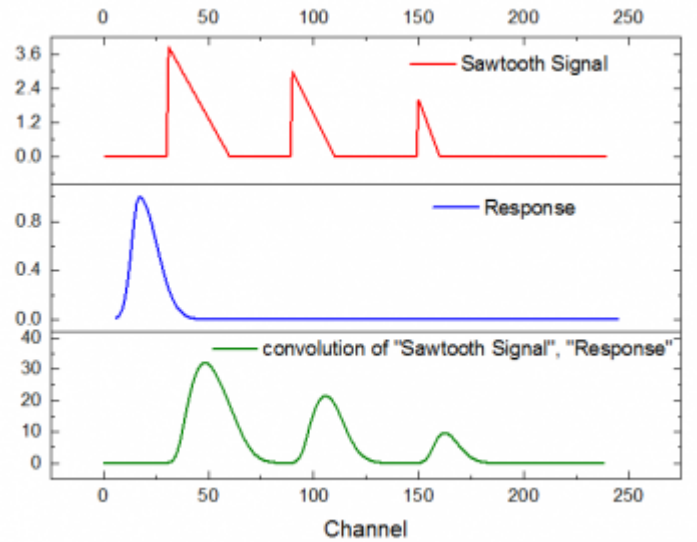
Expected Learning Outcomes

By the end of this week:

- Distinguish between classification and clustering as supervised and unsupervised learning methods
- Demonstrate how clustering algorithms work
- Demonstrate how classification algorithms work
- Understand their typical application scenarios

Last Week: Convolution

Convolution is a **mathematical operation** on two functions (f and g) that produces a third function ($f * g$) that expresses how the shape of one is modified by the other.

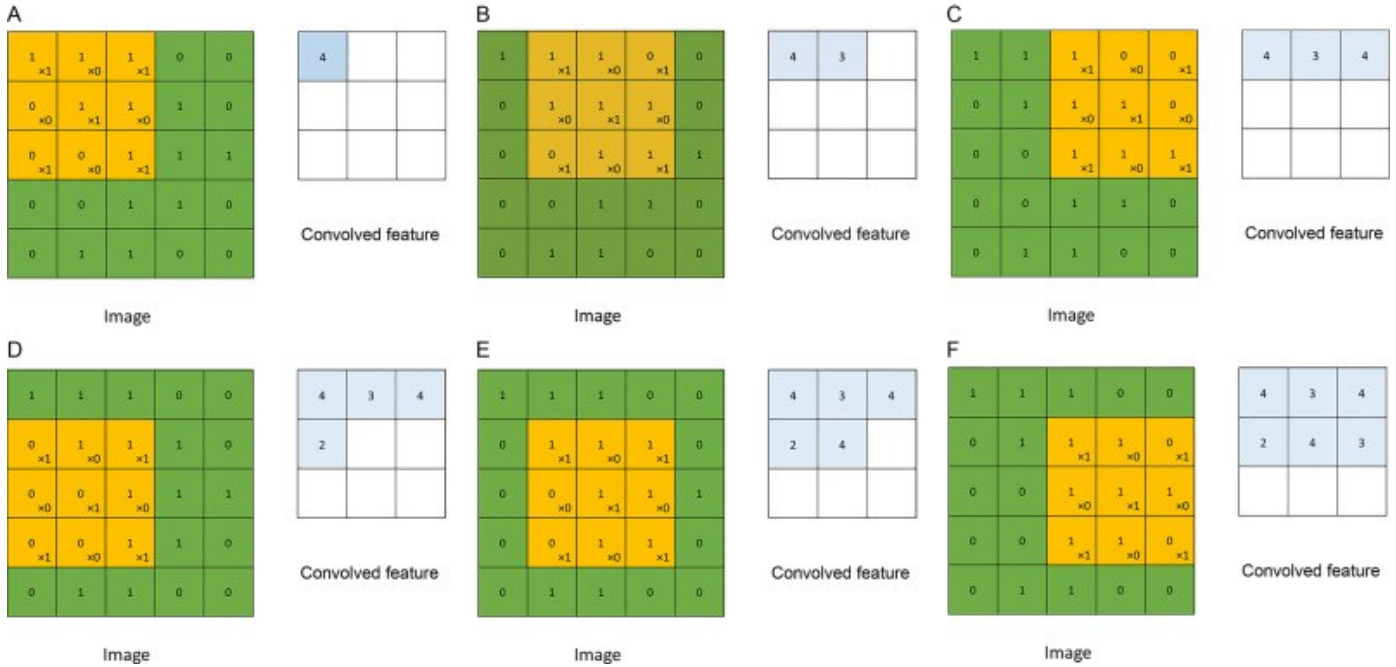


Last Week: Convolution

$$(f * g)(x_2, y_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x_1, y_1)\, g(x_2 - x_1, y_2 - y_1) \mathrm{d}y_1\, \mathrm{d}x_1$$

$$(\mathbf{f} * \mathbf{g})[i_2, j_2] = \sum_{i_1=-\infty}^{\infty} \sum_{j_1=-\infty}^{\infty} \mathbf{f}[i_1, j_1]\, \mathbf{g}[i_2 - i_1, j_2 - j_1]$$

Don't forget to transpose the kernel



Last Week: Properties of Convolution

Commutativity: $f * g = g * f$

Associativity: $f * (g * h) = (f * g) * h$

Distributivity: $f * (g + h) = f * g + f * h$

Associativity with scalar multiplication: $a(f * g) = (af) * g$ where $a \in \mathbb{C}$ (complex numbers)

Integration:

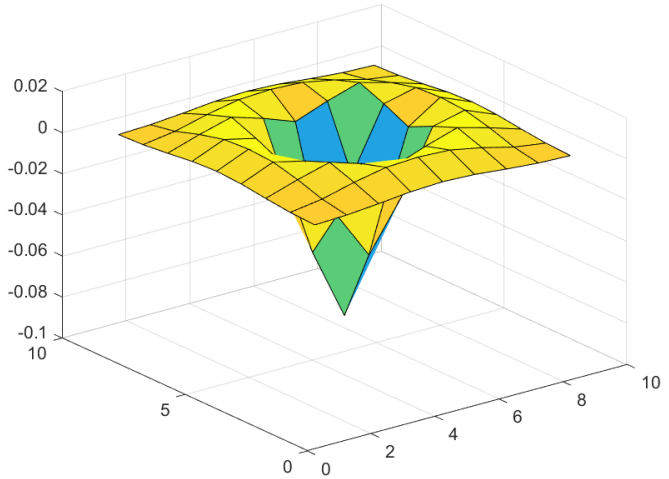
$$\int_{-\infty}^{\infty} (f * g)(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \int_{-\infty}^{\infty} g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

Last Week: Boundary Conditions

$$\int_{-\infty}^{\infty} (f * g)(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \int_{-\infty}^{\infty} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \int_{-\infty}^{\infty} g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}$$

$f(\boldsymbol{x})$:

$g(\boldsymbol{x})$:



$\int_{-\infty}^{\infty} f(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = 21217$

$\int_{-\infty}^{\infty} g(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = \mathbf{0}$

$$\int_{-\infty}^{\infty} (f * g)(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} = -103$$

Image

Full Image

Dirichlet Boundary Conditions

Periodic Boundary Conditions

Neumann Boundary Conditions

Last Week: Boundary Conditions

$$\int_\Omega (f * g)(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = \left(\int_\Omega f(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)\left(\int_\Omega g(\boldsymbol{x})\mathrm{d}\boldsymbol{x}\right)$$

Dirichlet

$\int_\Omega (f * g)(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = -102.5$

Neumann

$\int_\Omega (f * g)(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = -1.4$

Periodic

$\int_\Omega (f * g)(\boldsymbol{x})\mathrm{d}\boldsymbol{x} = 7.5 \times 10^{-15}$

Grammatical POS Tagging

Also known as morphosyntactic annotation

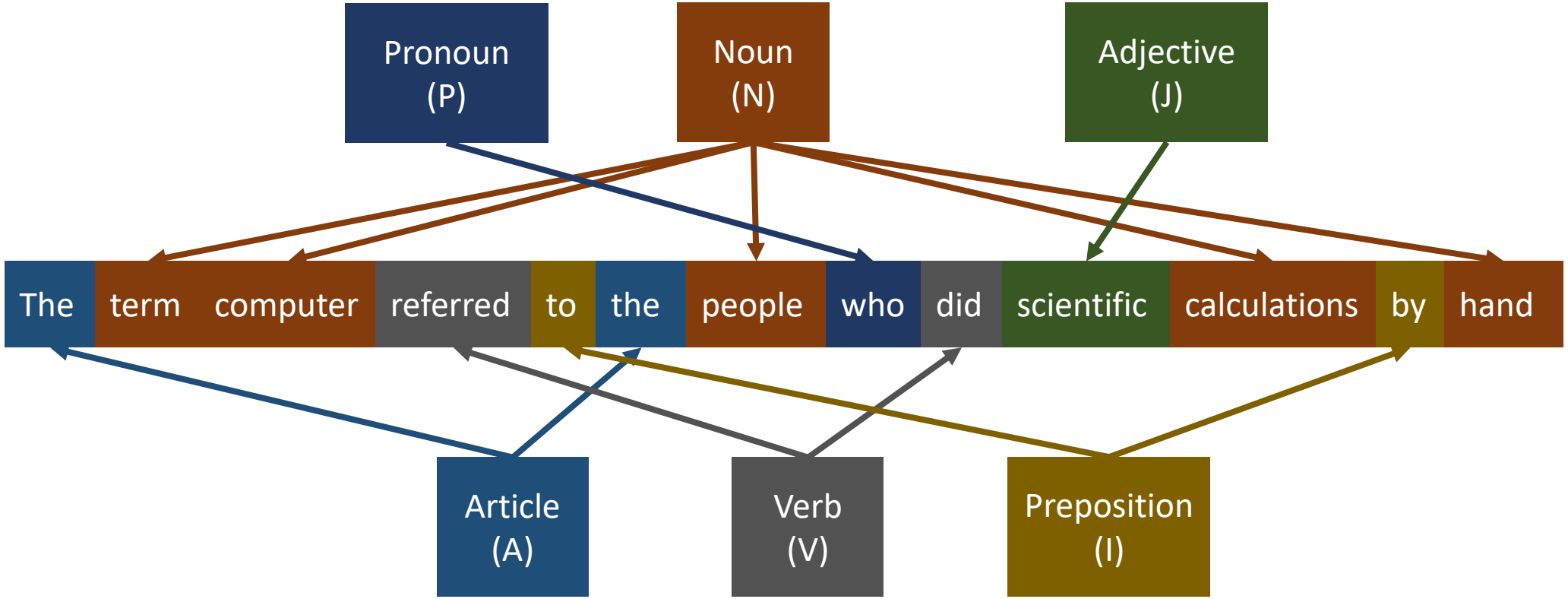
- Assign word-class labels
  - Major parts of speech (noun, verb, preposition, etc)
  - Sub-classes (singular vs plural nouns, comparative vs superlative, etc)

Language Models

Consider the sentence:

The term computer referred to the people who did scientific calculations by hand

Language Models



A N N V I A N P V J N I N

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Rule-based POS Tagging

Lancaster University

Initialisation

- Apply the most likely tag for each word

Define Rules

- Use immediate context
- Change Tag A to Tag B when
  - Preceding word is tagged Z
  - Following word is tagged Z

Evaluate

- Computer number of errors rectified
- Computer number of new errors introduced

Refine

- Learn iteratively
- Build list of rules

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand

Initial Options

A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J

Initial Estimate

A	V	N	V	R	A	V	P	V	J	N	R	N

Ground Truth

A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 69.2%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand
Initial Options												
A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J
Iteration 1												
A	V	N	V	R	A	V	P	V	J	N	R	N
Ground Truth												
A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 69.2%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand

Initial Options

A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J

Iteration 2

A	N	N	V	R	A	V	P	V	J	N	R	N

Ground Truth

A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 76.9%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand

Initial Options

A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J

Iteration 3

A	N	N	V	R	A	N	P	V	J	N	R	N

Ground Truth

A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 84.6%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand

Initial Options

A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J

Iteration 4

A	N	N	V	R	A	N	P	V	J	N	I	N

Ground Truth

A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 84.6%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand

Initial Options

A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J

Iteration 5

A	N	N	V	R	A	N	P	V	J	N	I	N

Ground Truth

A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 84.6%

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

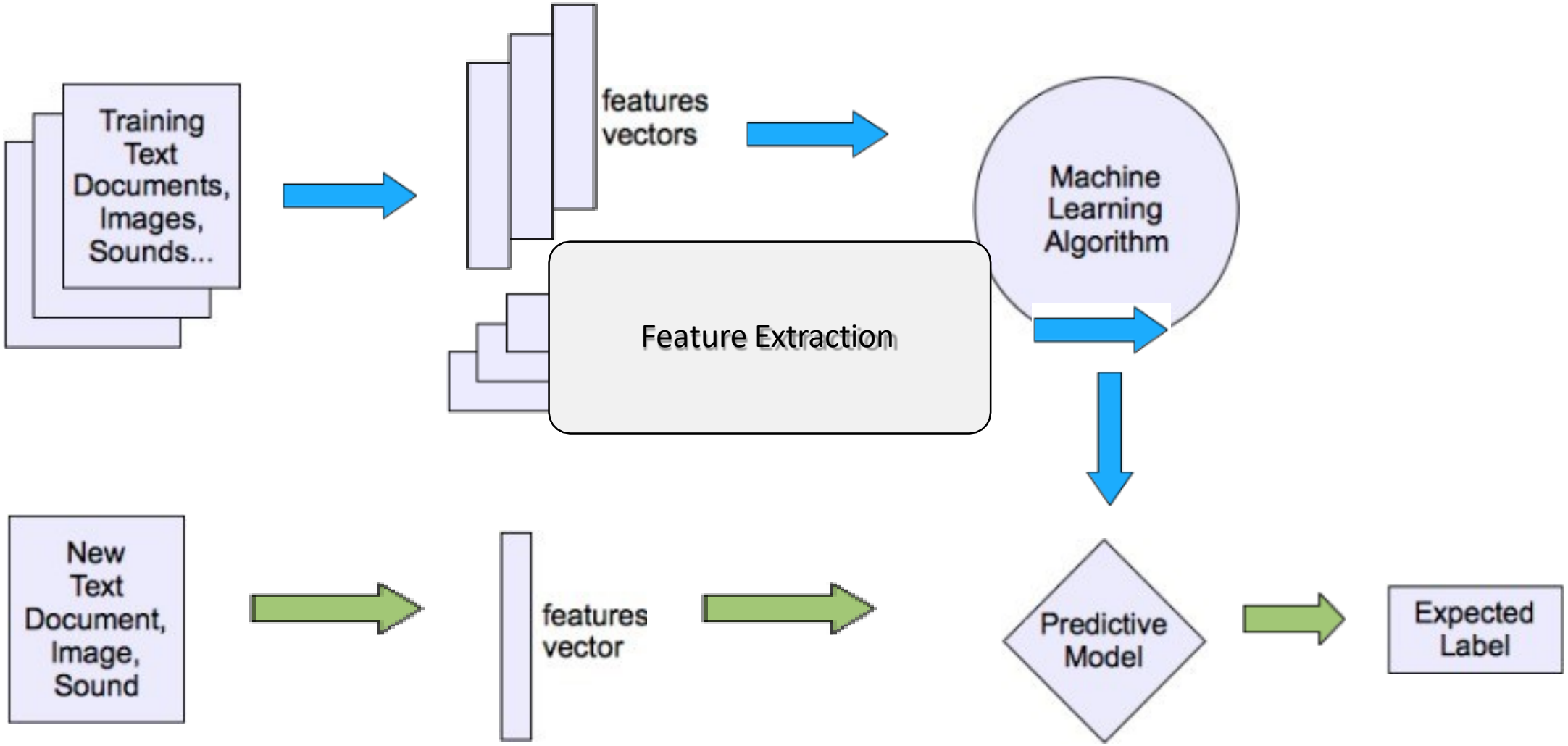
Language Models - Example

The	term	computer	referred	to	the	people	who	did	scientific	calculations	by	hand
Initial Options												
A	N V	N	V	I R	A	N V	P	V	J	N	I R J	N V J
Final Result												
A	N	N	V	R	A	N	P	V	J	N	I	N
Ground Truth												
A	N	N	V	I	A	N	P	V	J	N	I	N

Accuracy: 84.6%, low confidence

POS Tagset	
A	Article
C	Conjunction
D	Determiner
I	Preposition
J	Adjective
M	Number
N	Noun
P	Pronoun
R	Adverb
V	Verb

Machine Learning Paradigm



This Week

Lancaster University

Week 1: Intro to AI and ML

- Definitions
- Goals of AI
- AI in Society
- Machine Learning
- Supervised vs Unsupervised

Week 2: Features and Extraction

- Features and Feature Vectors
- Feature Labels
- Filters
- 1-Hot Encoding
- Bag of Words

Week 3: CV and NLP

- Computer Vision
- Convolution and Properties
- Natural Language Processing
- Parts of Speech Tagging

Week 4: Clustering and Classification

- **Hierarchical Clustering**
- **K-Means Clustering**
- **Classification Algorithms**
- **Nearest Neighbour**

Week 5: Artificial Neural Networks

- ANNs and Components
- Training and Solution Techniques
- Multilayer Networks and Components
- Recurrent Neural Networks

This Week

Lancaster University

Week 1: Intro to AI and ML

- Definitions
- Goals of AI
- AI in Society
- Machine Learning
- Supervised vs Unsupervised

Week 2: Features and Extraction

- Features and Feature Vectors
- Feature Labels
- Filters
- 1-Hot Encoding
- Bag of Words

Week 3: CV and NLP

- Computer Vision
- Convolution and Properties
- Natural Language Processing
- Parts of Speech Tagging

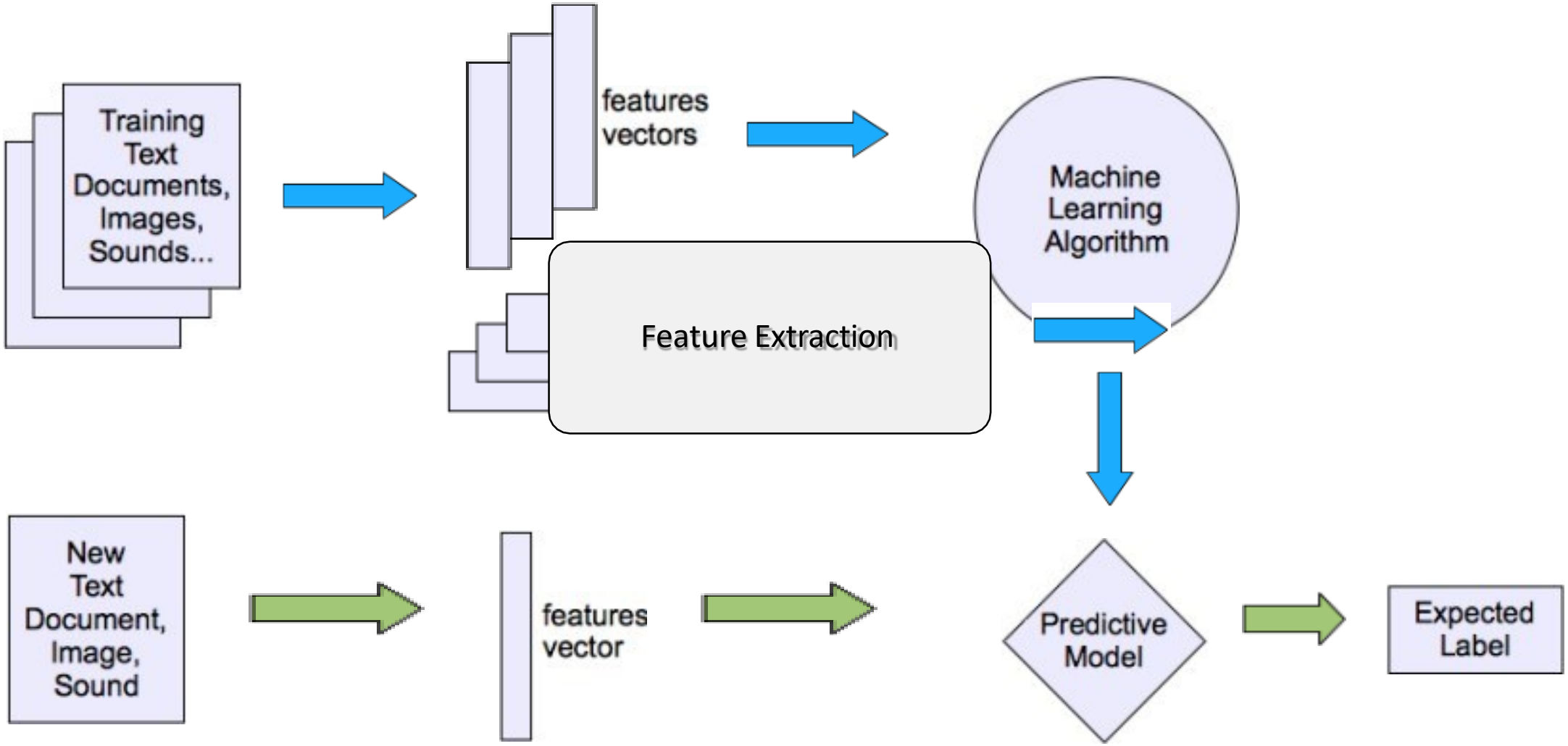
Week 4: Clustering and Classification

- **Hierarchical Clustering**
- **K-Means Clustering**
- **Classification Algorithms**
- **Nearest Neighbour**

Week 5: Artificial Neural Networks

- ANNs and Components
- Training and Solution Techniques
- Multilayer Networks and Components
- Recurrent Neural Networks

Machine Learning Paradigm



Machine Learning Paradigm

Training Data
The observed set of example data

Test Data
Previously unseen data generated by the same process as the training data

Training process

Inferring or learning something about the process that generated the training data

Prediction

Using the inference to make predictions about the test data

Evaluation

Comparing the prediction outputs with the correct test data outputs

Machine Learning Paradigm

Training Data
The observed set of example data

Validation Data
Data tested during the training process

Test Data
Previously unseen data generated by the same process as the training data

Training process

Inferring or learning something about the process that generated the training data

Prediction

Using the inference to make predictions about the test data

Evaluation

Comparing the prediction outputs with the correct test data outputs

Supervised vs Unsupervised

Supervised learning

- Given a set of feature-label pairs, find a rule that predicts the label associated with a previously unseen input
  - Classification
  - Regression

Unsupervised learning

- Given a set of feature vectors (without labels) group them into some “natural clusters”
  - Clustering
  - Association

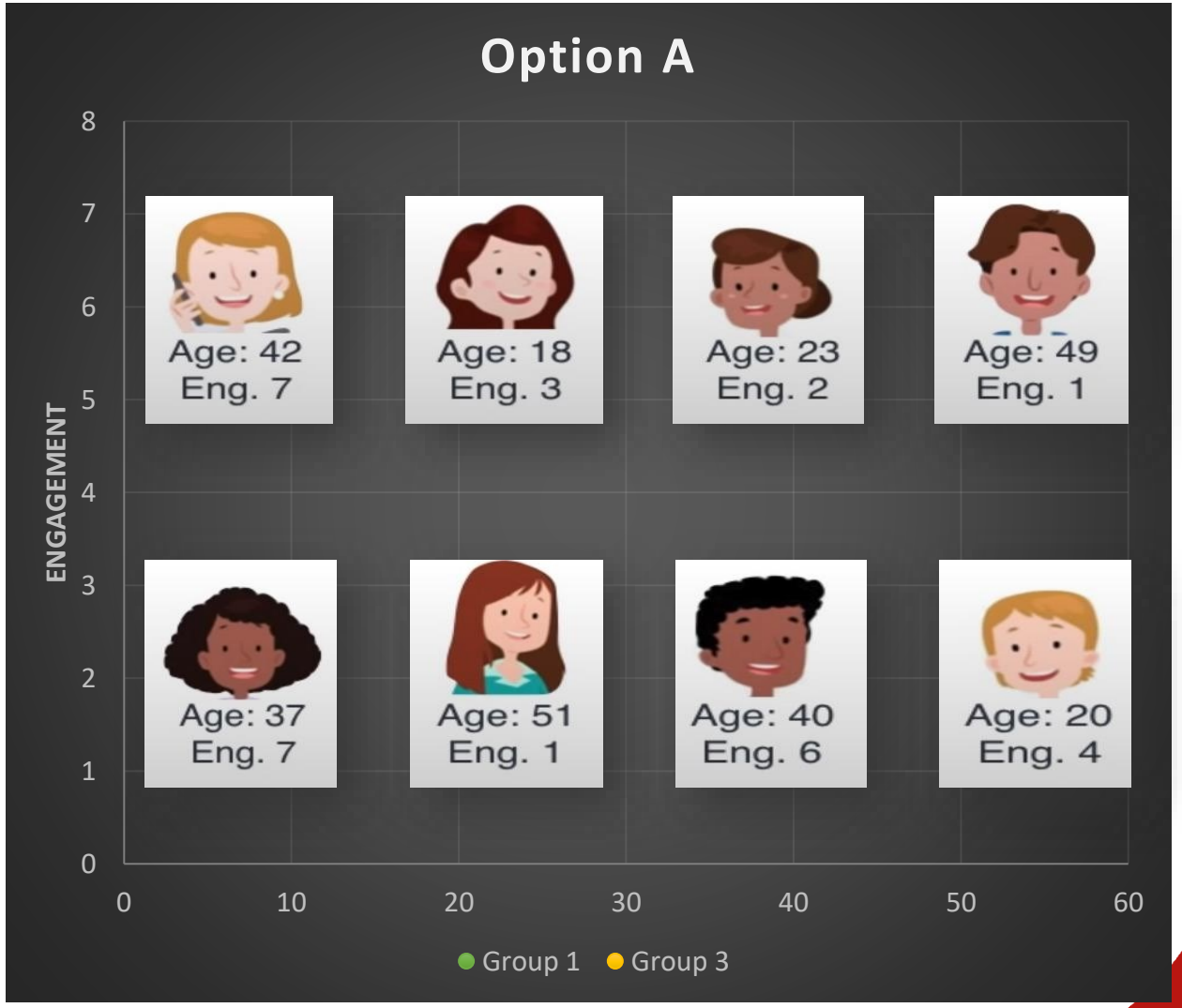
Clustering

Customer Engagement Data

Here are the **ages** (in years) and **engagements** (in days/weeks) of our customers that use our app.

If we have to put them into three groups to effectively serve them, how should we do that?

A	{1,5,6} {4,8} {2,3,7}
B	{1,8,3} {4,7,2} {5,6}
C	{2,8,3} {5,7,1} {4,6}
D	{3,7,8} {4,7} {2,5,6}

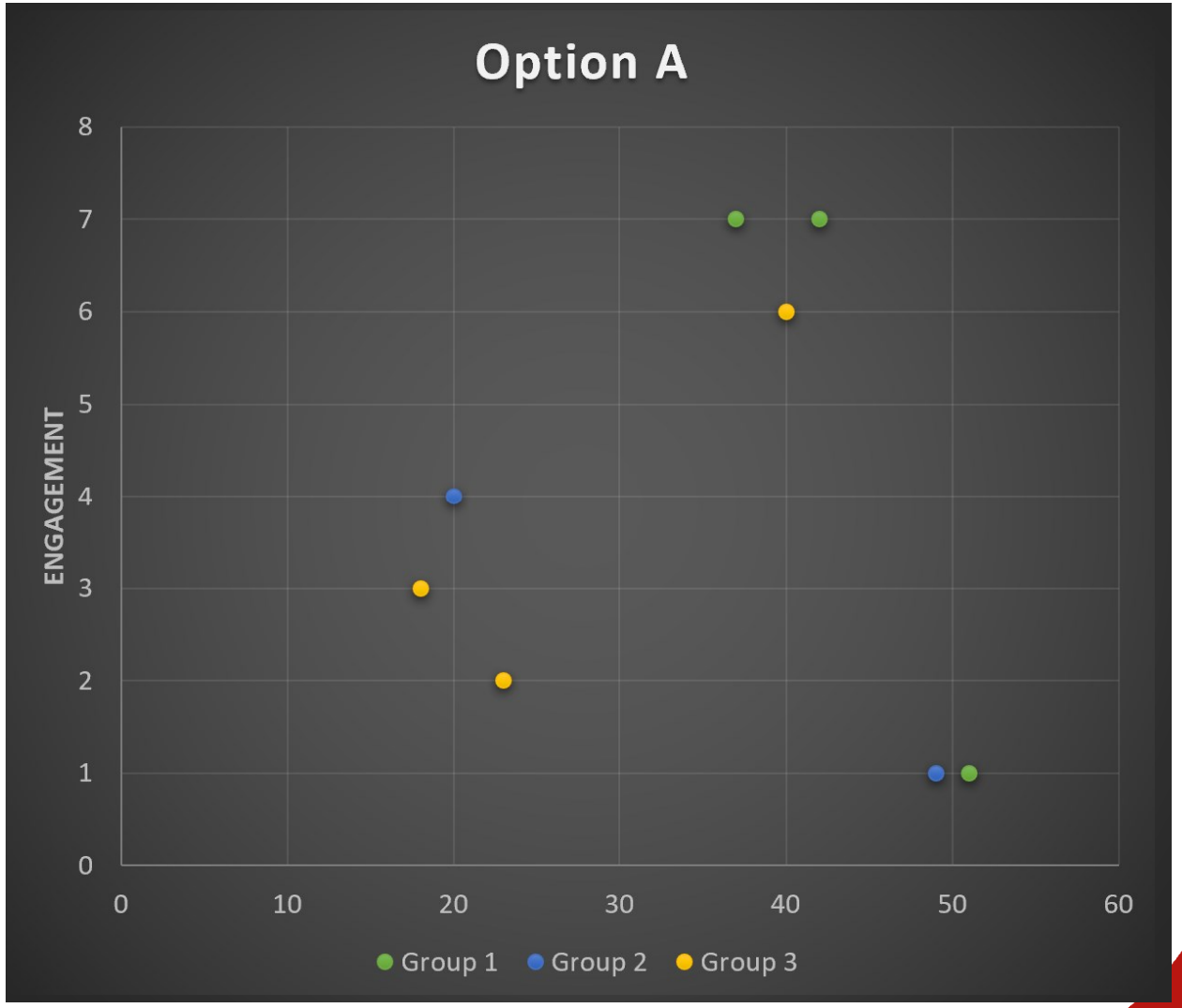


Customer Engagement Data

Here are the **ages** (in years) and **engagements** (in days/weeks) of our customers that use our app.

If we have to put them into three groups to effectively serve them, how should we do that?

A	{1,5,6} {4,8} {2,3,7}
B	{1,8,3} {4,7,2} {5,6}
C	{2,8,3} {5,7,1} {4,6}
D	{3,7,8} {4,1} {2,5,6}

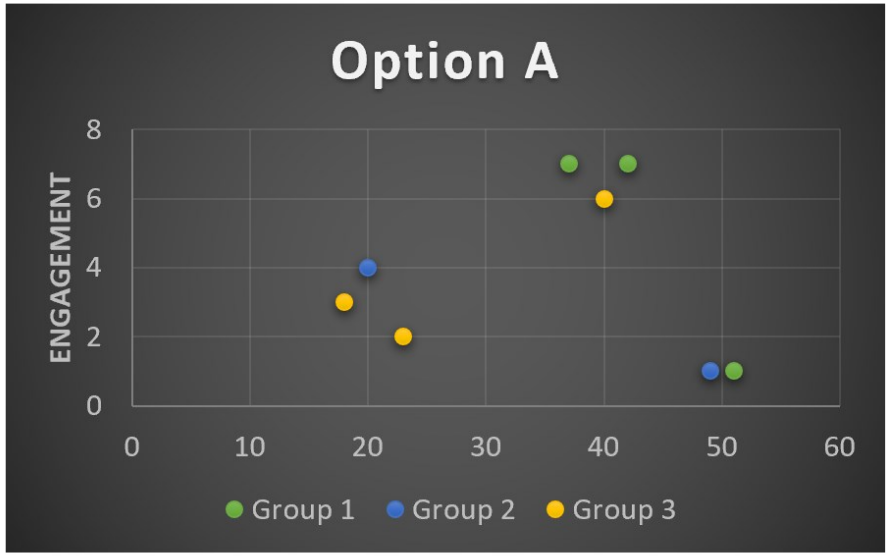


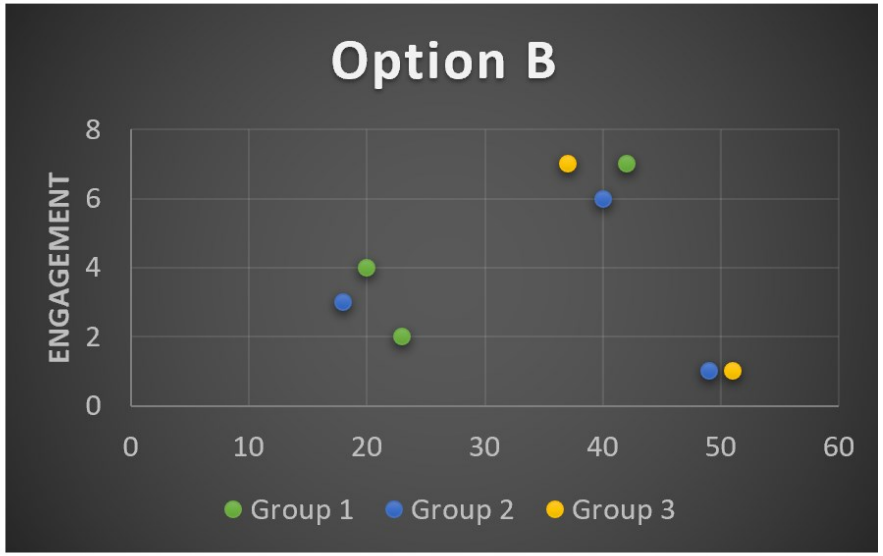
Customer Engagement Data

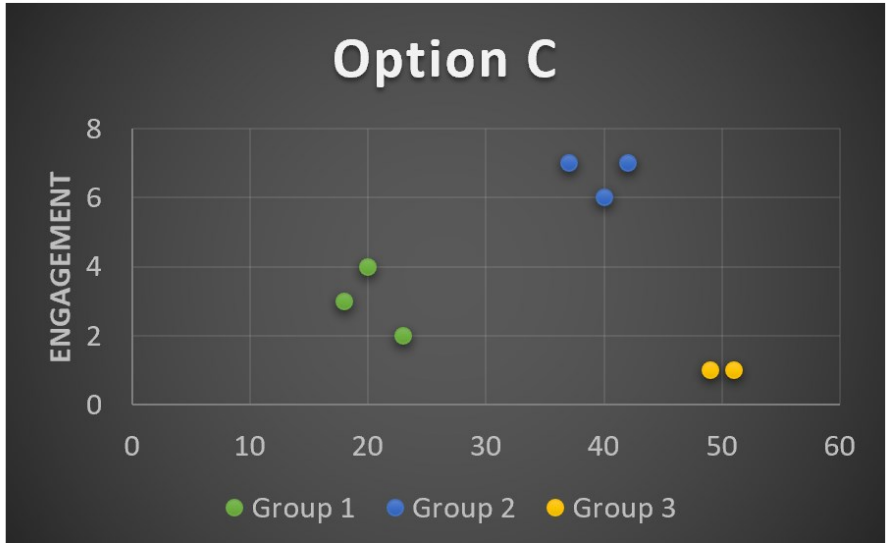
Here are the **ages** (in years) and **engagements** (in days/weeks) of our customers that use our app.

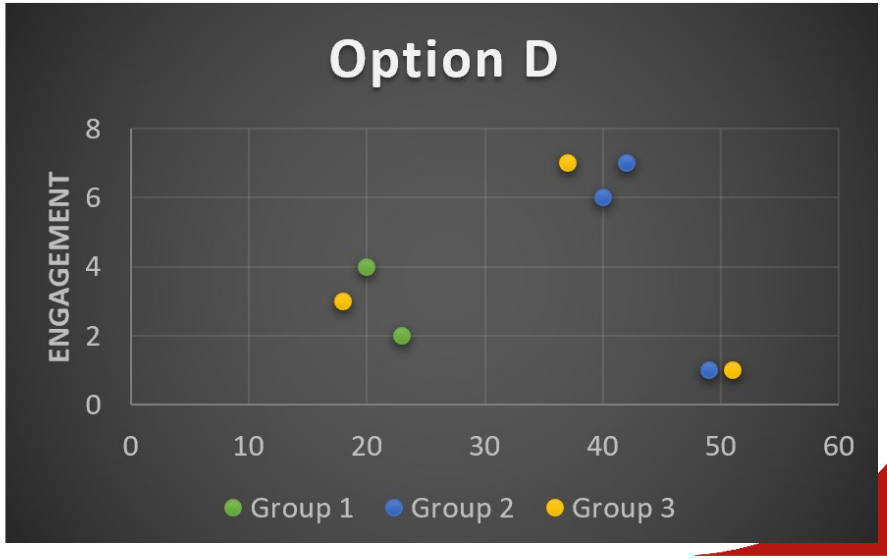
If we have to put them into three groups to effectively serve them, how should we do that?

A	{1,5,6} {4,8} {2,3,7}
B	{1,8,3} {4,7,2} {5,6}
C	{2,8,3} {5,7,1} {4,6}
D	{3,7,8} {4,1} {2,5,6}









Learning from Data

ML Goal

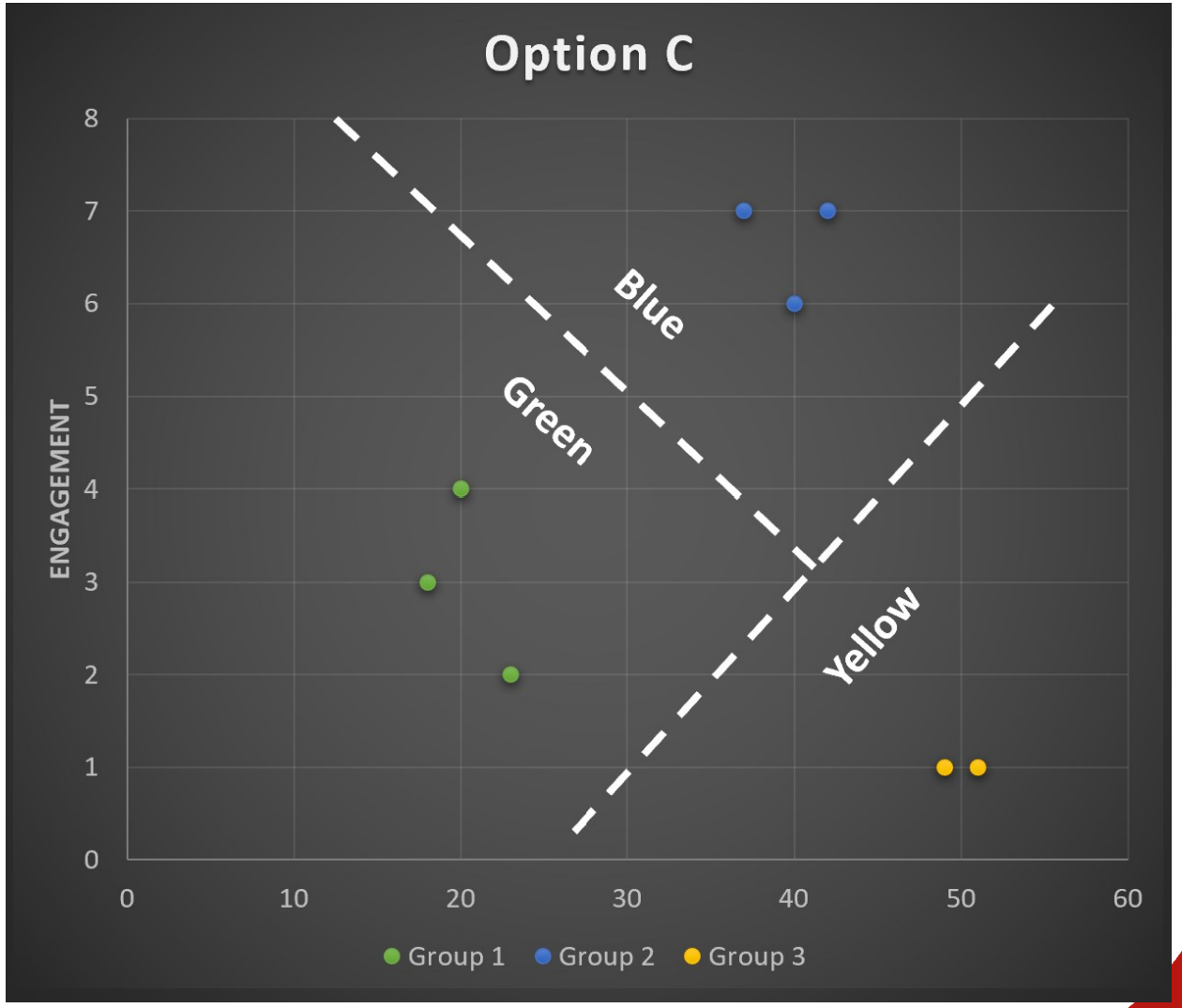
Learn from observed data in two ways

1: Clustering

- Identify meaningful patterns, clusters or groups in observed data points

2: Classification

- Classify or categorise new data points into one of the identified groups



What is Clustering?

How do we group these characters?

What is Clustering?

Grouping data into “clusters”

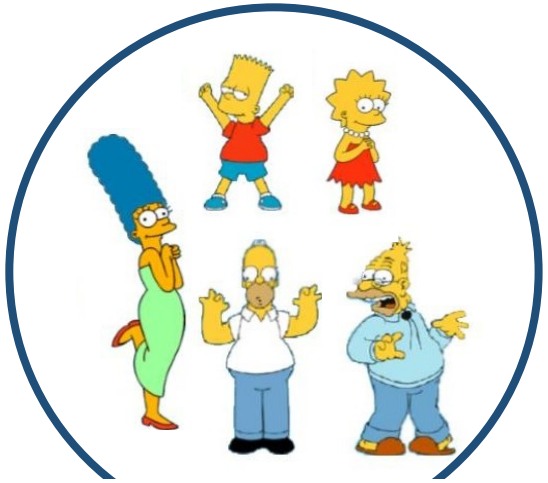
Optimisation with constraints:

- Number of clusters
- Minimum distance between clusters

Reduce dissimilarity between members in a cluster

Two common methods:

Hierarchical | K-Means



Simpson Family

VS

Non-Simpson Family

Male

VS

Female

Hierarchical Clustering

E.g. Family Tree

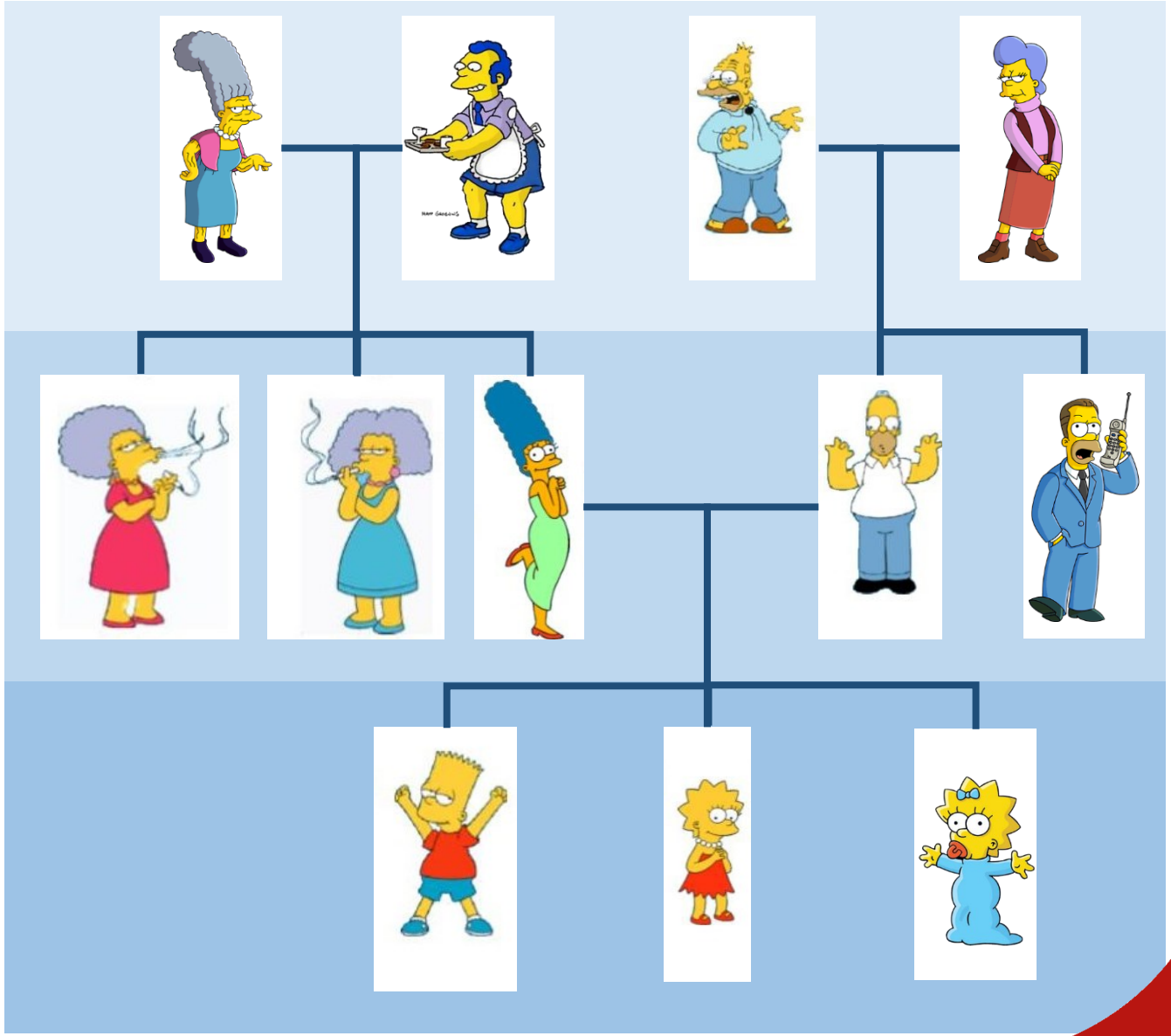
Two common approaches:

Agglomerative (bottom-up)

- Each observation starts in own cluster.
- Pairs of clusters are merged as you move up the hierarchy.

Divisive (top-down)

- All observations start in one cluster.
- Splits are performed recursively as you move down the hierarchy.



Hierarchical Clustering

Aim: Build a hierarchy of clusters

1: Initialisation

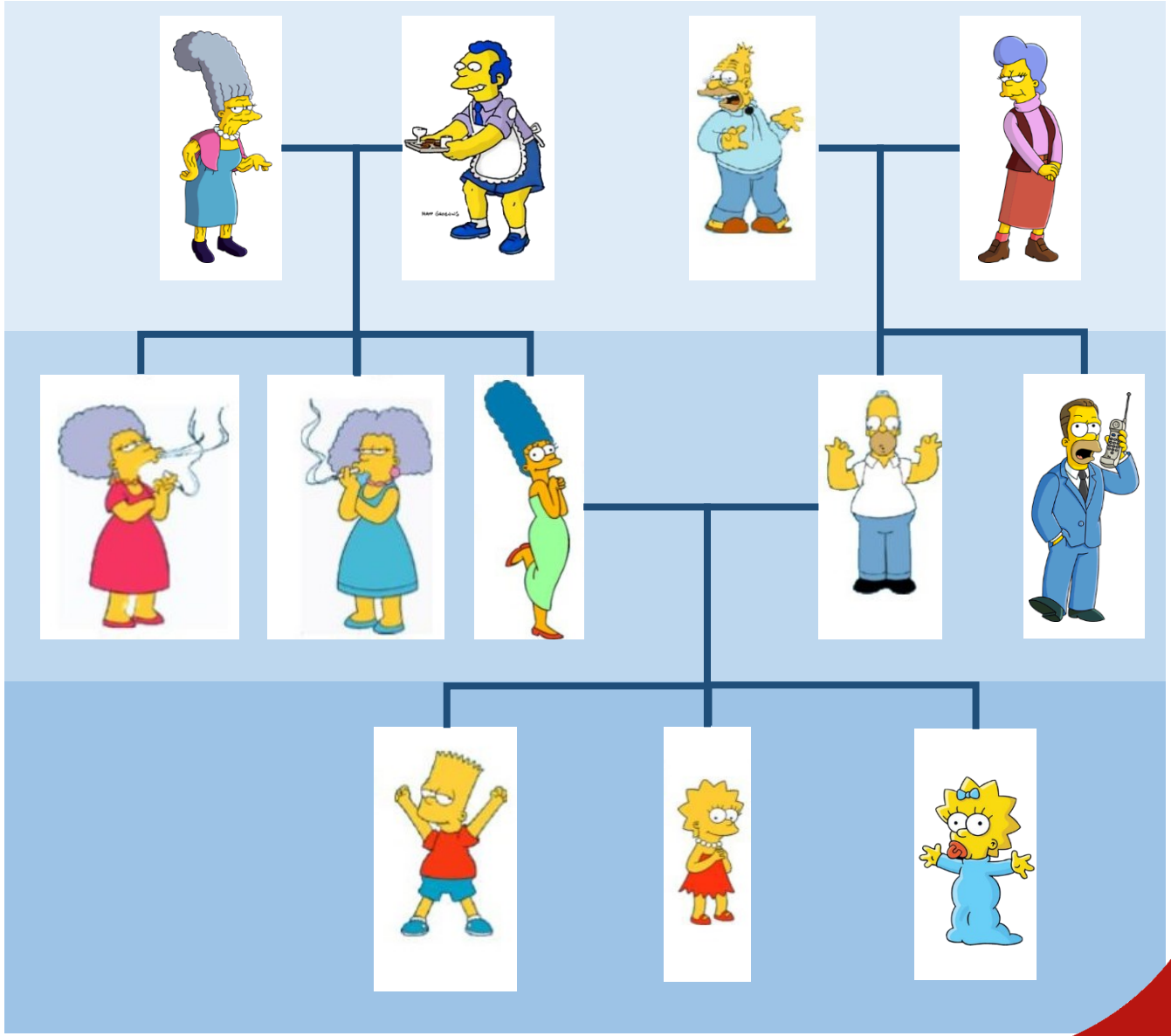
- Assign each item to a cluster
- n items results in n clusters containing 1 item each
- Here, $n = 12$

2: Merge iteratively

- Find and merge the **closest** pair of clusters into a single cluster within a range
- Continue until you have a single cluster of n items

Result

- Hierarchical decomposition of the dataset



Hierarchical Clustering – Example 1

Distance Measure

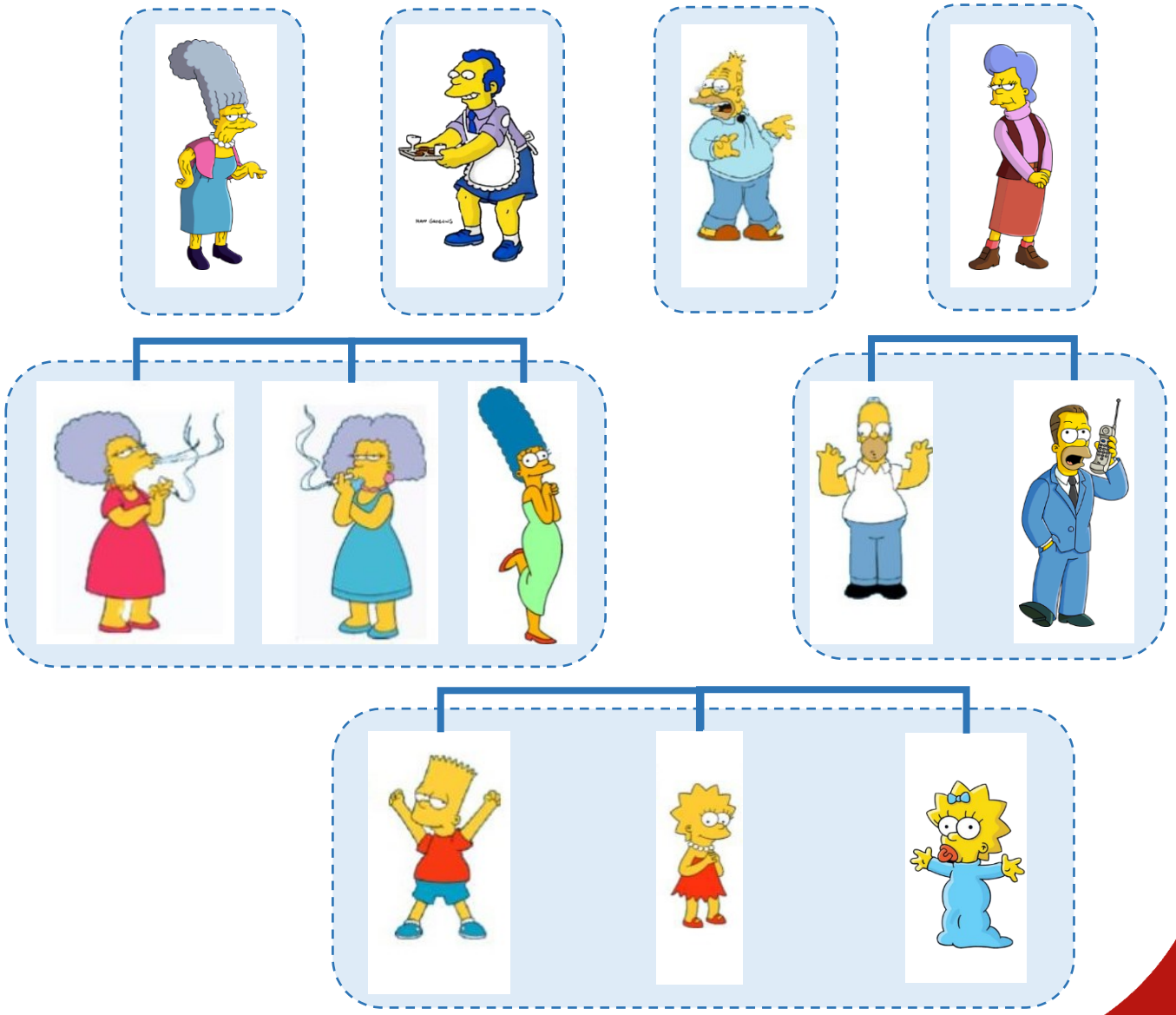
Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling



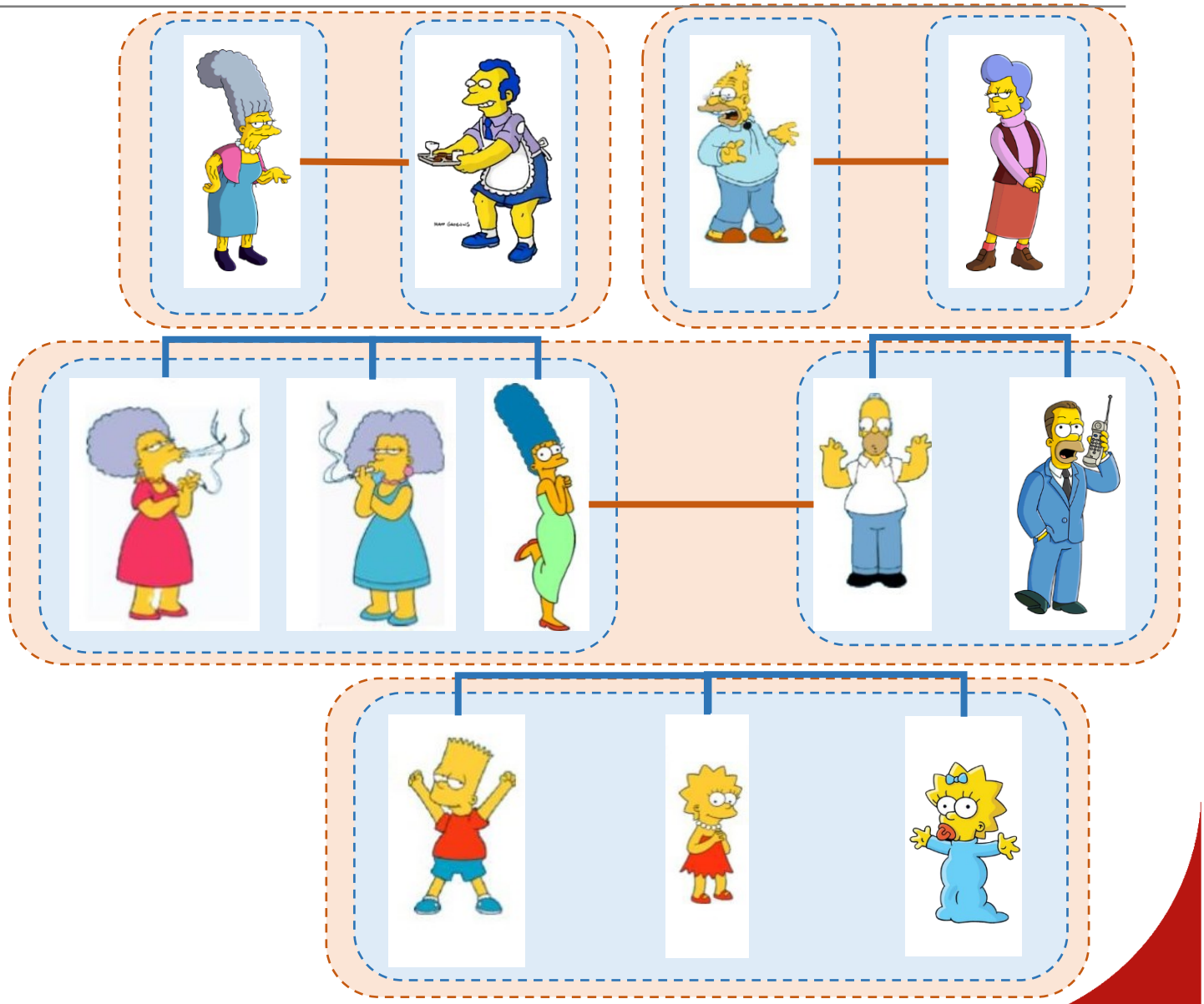
Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling

Level 2: Spouse

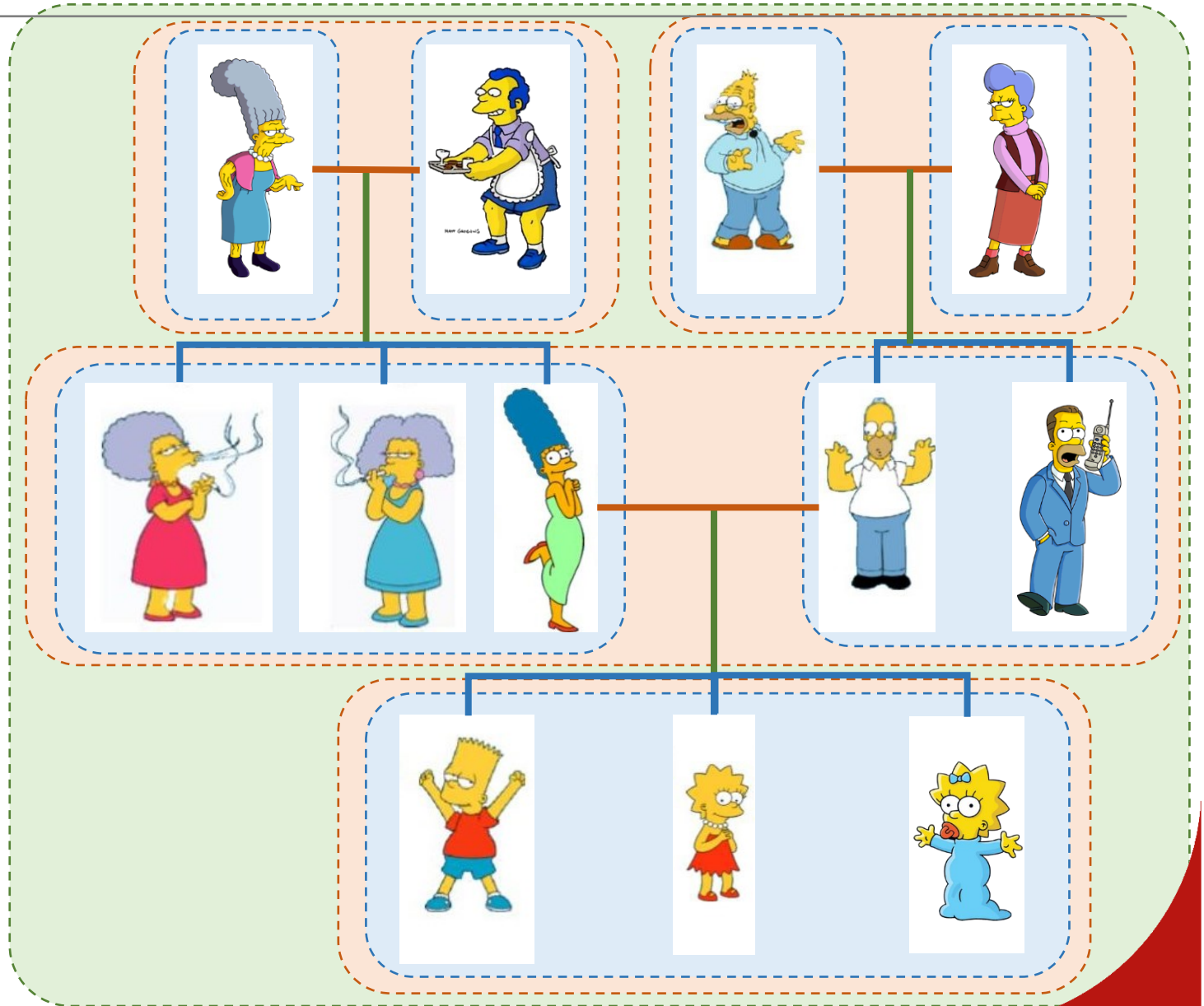


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling
Level 2: Spouse
Level 3: Parent/ Child

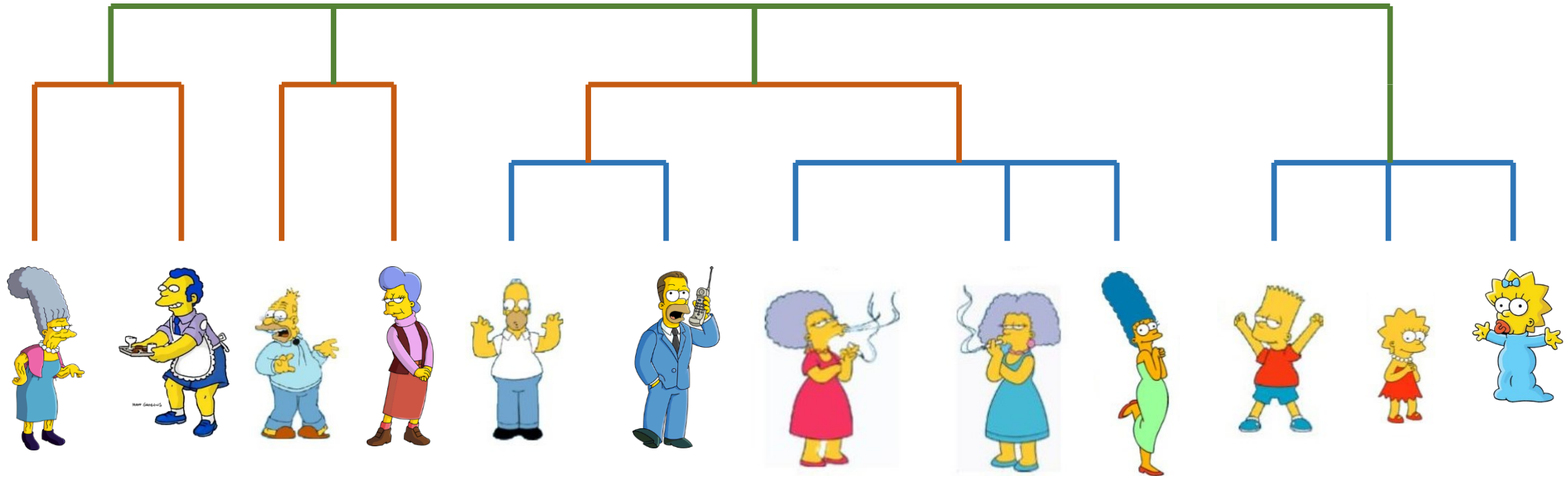


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Sibling	1
Spouse	2
Parent/ Child	3

Level 1: Sibling
Level 2: Spouse
Level 3: Parent/ Child

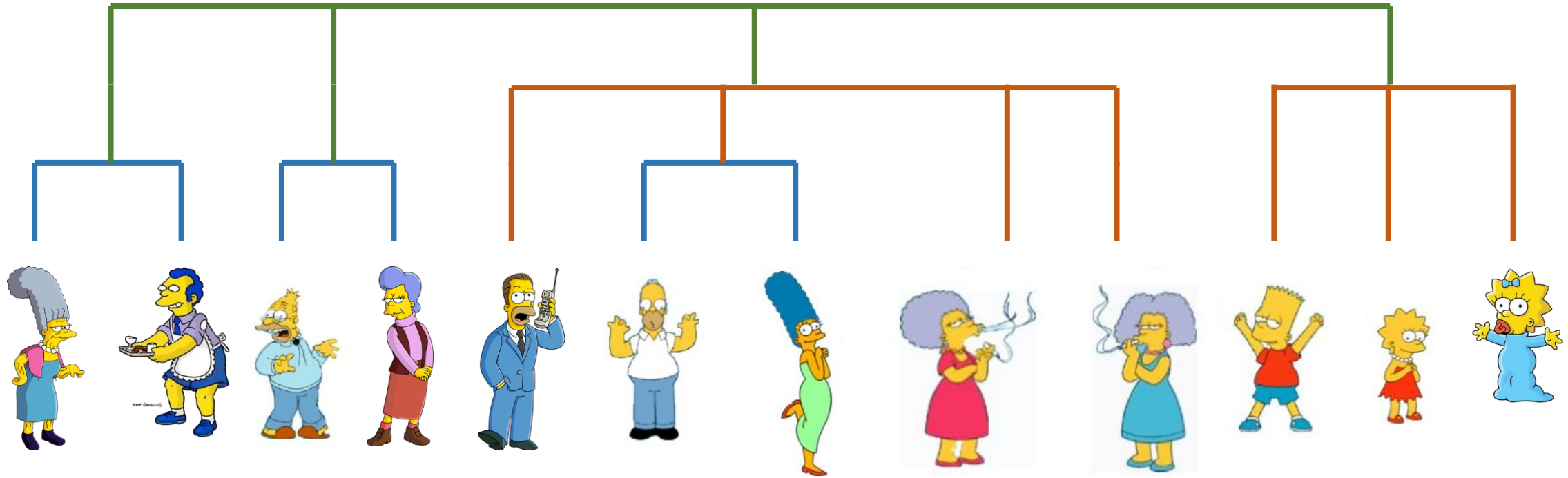


Hierarchical Clustering – Example 1

Distance Measure

Relationship	Value
Spouse	1
Sibling	2
Parent/ Child	3

Level 1: Spouse
Level 2: Sibling
Level 3: Parent/ Child



Hierarchical Clustering – Example 2

ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN

IATA Codes for 12 Random UK Airports

ABZ: Aberdeen

BRS: Bristol

EDI: Edinburgh

GLA: Glasgow

INV: Inverness

LBA: Leeds Bradford

LHR: London Heathrow

LPL: Liverpool John Lennon

LTN: London Luton

MAN: Manchester

NCL: Newcastle

STN: London Stansted

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

Distance to nearest mile between airports

Source: https://airport.globefeed.com/UK_Distance_Between_Airports.asp

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

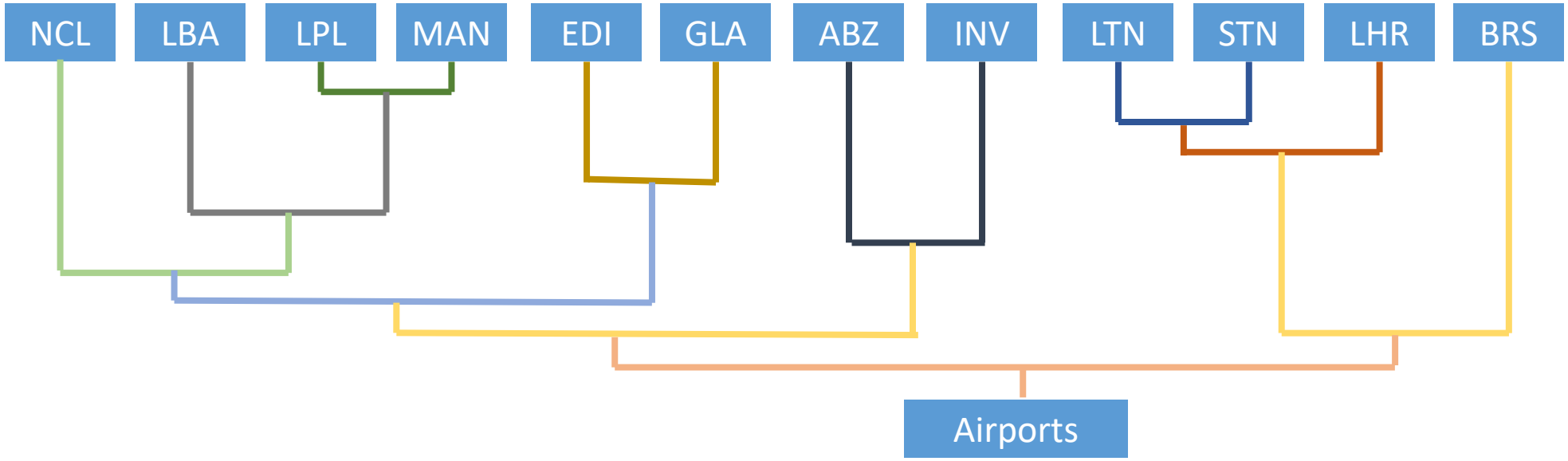
Distance to nearest mile between airports

ABZ BRS EDI GLA INV LBA LHR LPL LTN MAN NCL STN

Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

Distance to nearest mile between airports

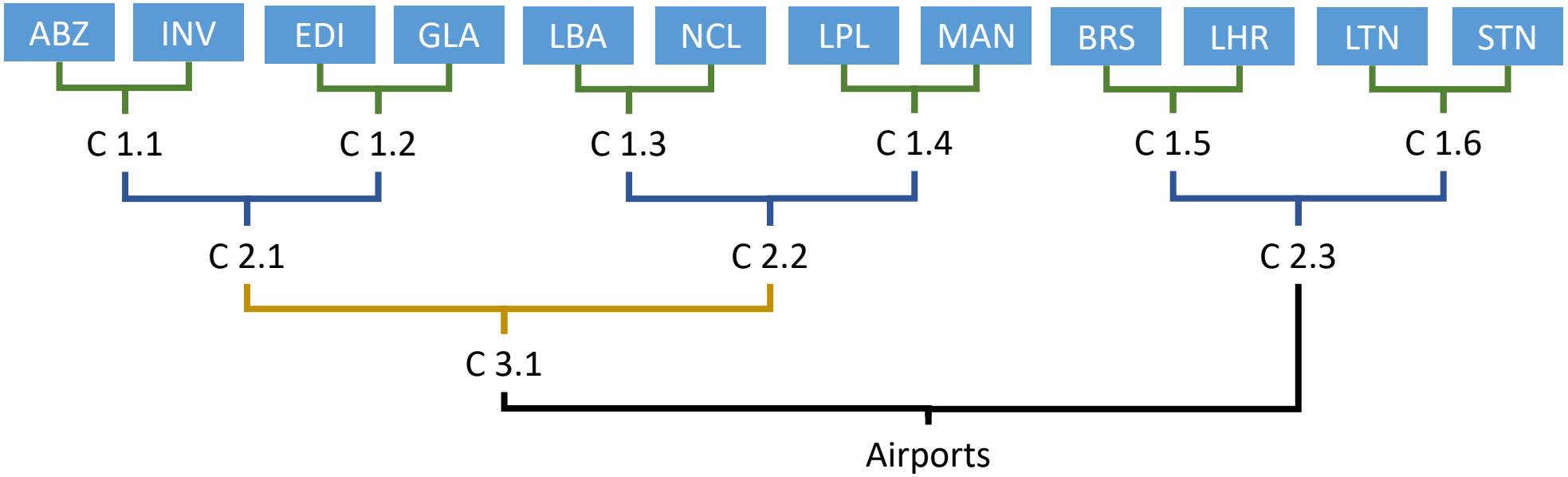


Hierarchical Clustering – Example 2

	ABZ	BRS	EDI	GLA	INV	LBA	LHR	LPL	LTN	MAN	NCL	STN
ABZ	0											
BRS	403	0										
EDI	97	317	0									
GLA	125	318	41	0								
INV	73	429	112	116	0							
LBA	231	177	159	177	270	0						
LHR	402	97	332	345	443	173	0					
LPL	268	134	182	186	294	61	163	0				
LTN	375	106	307	322	418	148	28	145	0			
MAN	266	137	185	194	297	43	151	24	113	0		
NCL	242	256	91	122	195	81	252	127	225	119	0	
STN	380	131	317	335	426	158	41	164	26	146	232	0

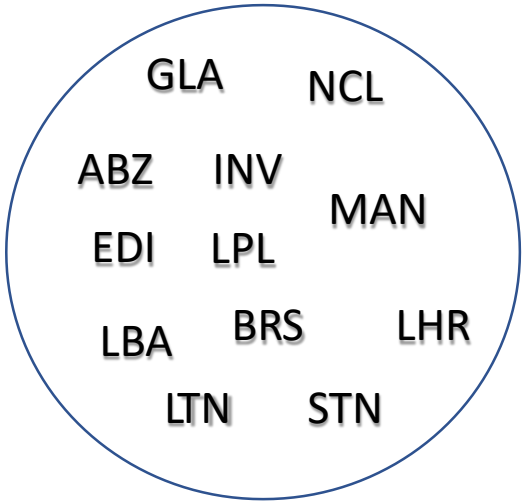
Distance to nearest mile between airports

1. Find each pair of clusters
2. Merge the closest pair



Linkage Metrics

A linkage criterion is used to determine whether cluster pairs can be merged



One big cluster containing all items

Different cluster containing all items

Common Distance Metrics

Name	Formula
Euclidean Distance	$\|a-b\|_2 = \sqrt{\sum_i (a_i - b_i)^2}$
Square Euclidean Distance	$\|a-b\|_2^2 = \sum_i (a_i - b_i)^2$
Manhattan Distance	$\|a-b\|_1 = \sum_i \|a_i - b_i\|$
Maximum Distance	$\|a-b\|_\infty = \max_i \|a_i - b_i\|$
Mahalanobis Distance	$\sqrt{(a-b)^\top S^{-1}(a-b)}$ where S is the Covariance matrix

Hamming distance or Levenshtein distance are often used for text or other non-numeric data

Source: Hierarchical Clustering on Wikipedia

Linkage Types

Single-linkage:

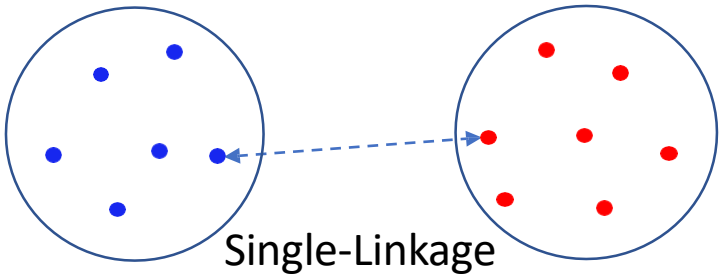
- shortest distance from any member of one cluster to any member of the other cluster

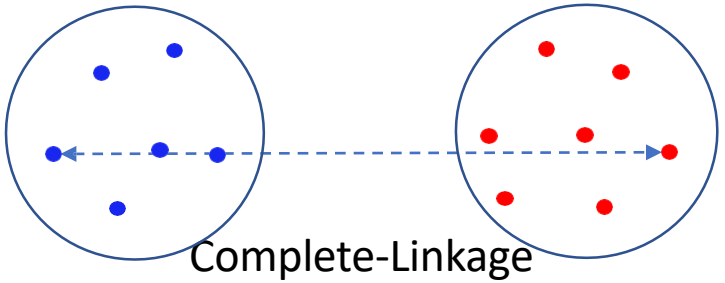
Complete-linkage:

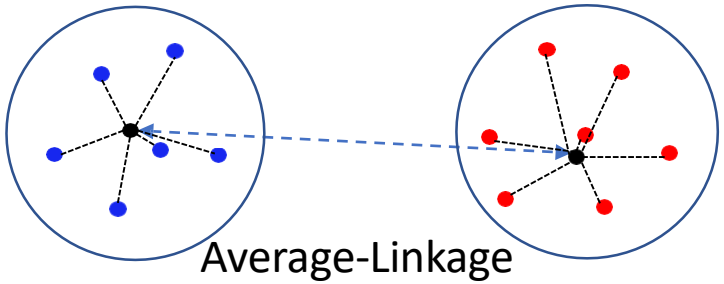
- greatest distance from any member of one cluster to any member of the other cluster

Average-linkage:

- average distance from any member of one cluster to any member of the other cluster

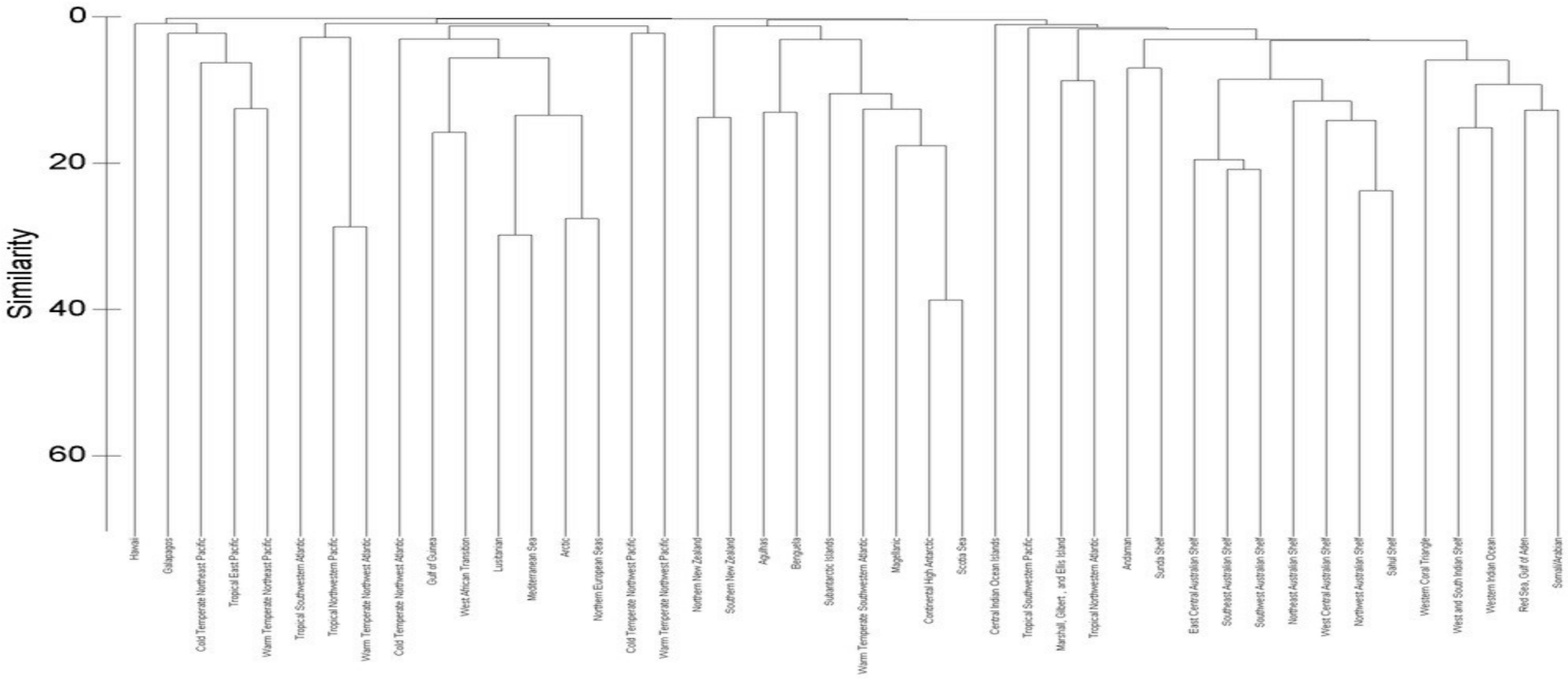






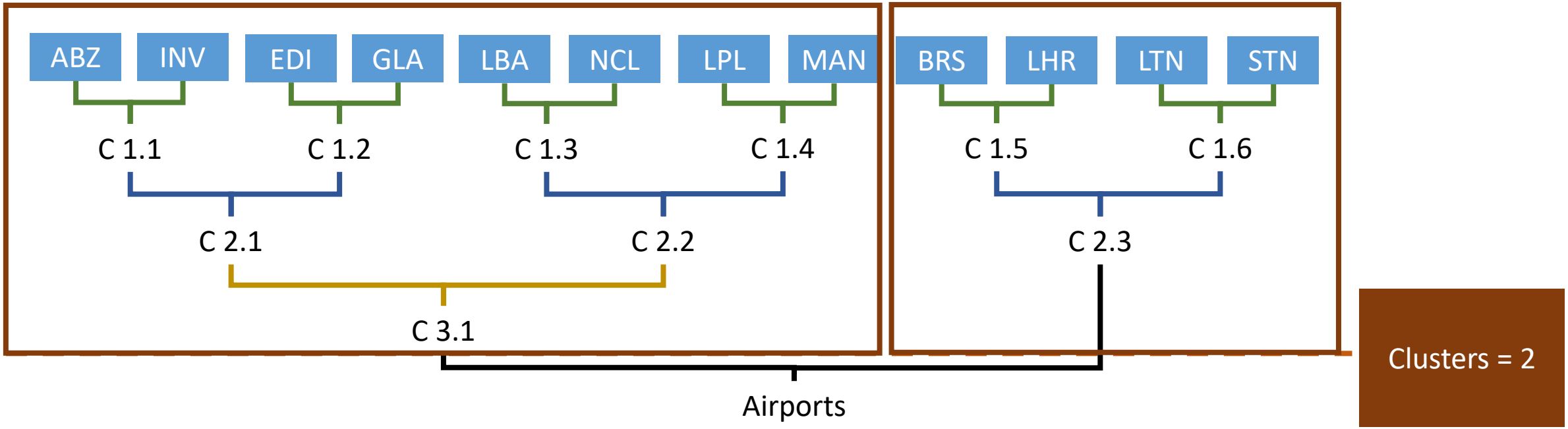
Dendrogram

A **dendrogram** is a diagrammatic representation
It is often used in hierarchical clustering



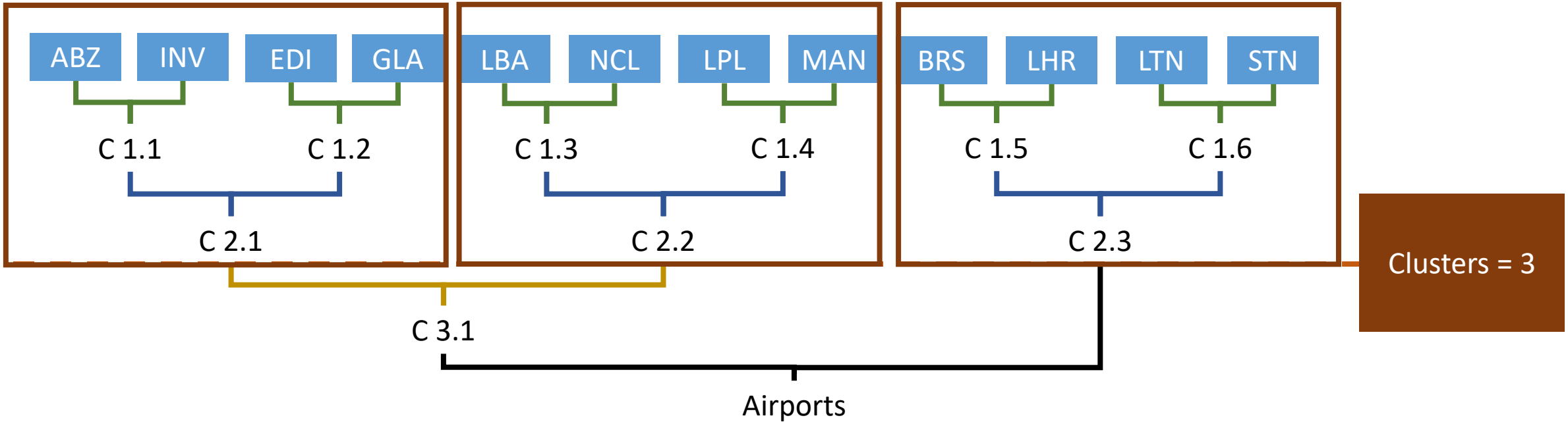
Dendrogram

We can obtain different clusterings of the data by splitting on different levels



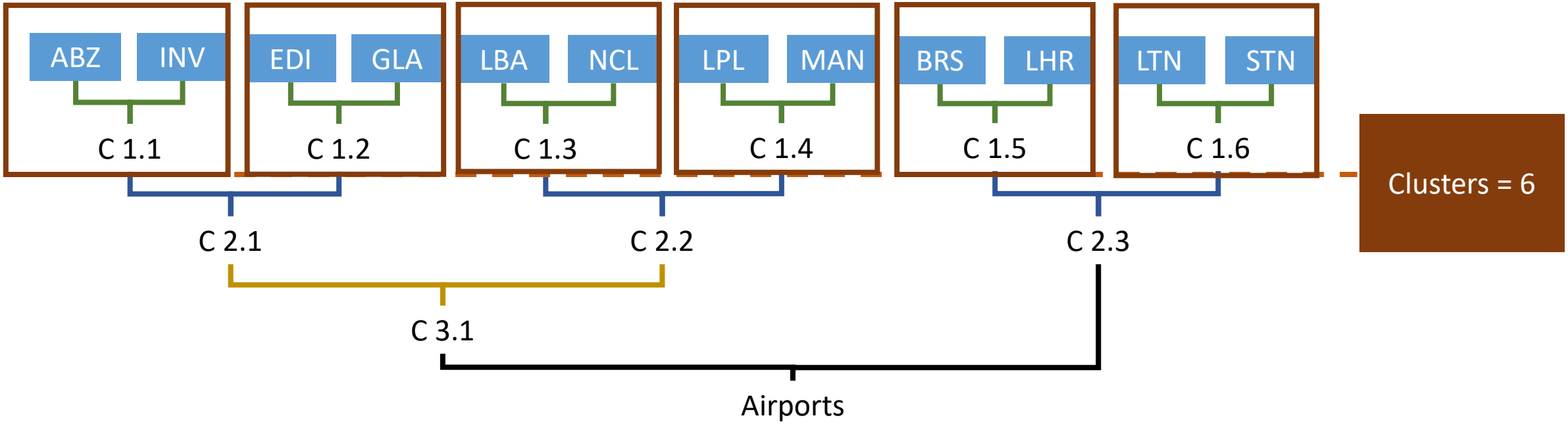
Dendrogram

We can obtain different clusterings of the data by splitting on different levels



Dendrogram

We can obtain different clusterings of the data by splitting on different levels



Hierarchical Clustering Summary

Deterministic

- You can always get the same result

Inefficient

- Too many distance computations to perform
- Depends on the task

Multilevel Representation

- No need to select number of clusters
- Can be visualised with dendrogram to select k

Flexible

- with respect to linkage criteria

K-Means Clustering

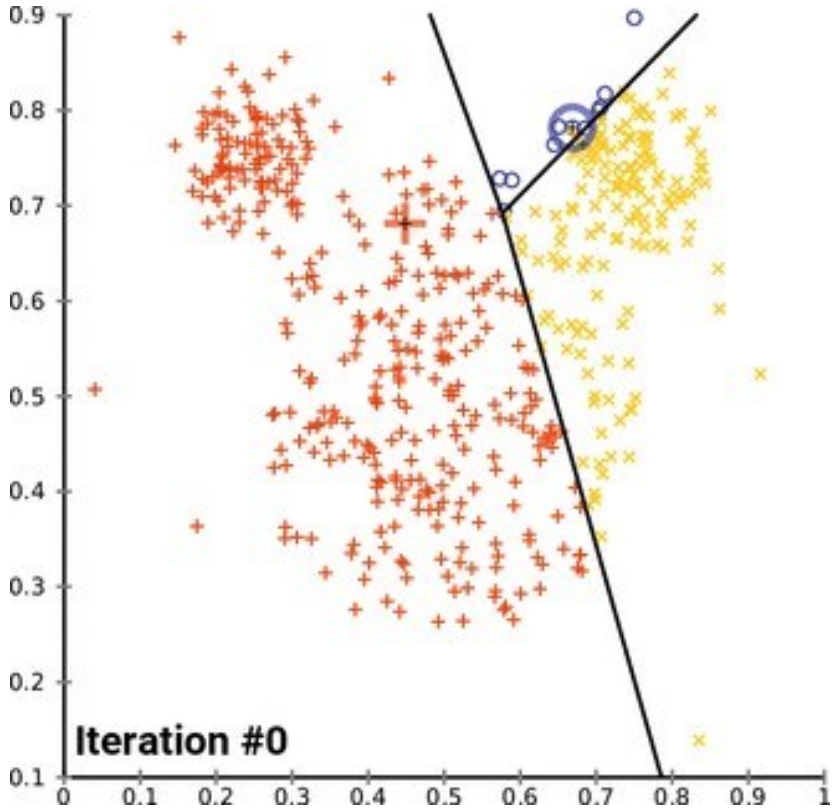
K-Means clustering is an **optimisation problem**

Aim:
Minimize the within-cluster sum of squares (WCSS) i.e. variance

Given n observations $(x_1, x_2, \dots, x_n)$, where each is a d-dimensional real vector, K-Means partitions the n observations into k sets S where $k \leq n$:

$$S = \{S_1, S_2, \dots, S_k\}$$

Such that the variance of each subset S_i is minimised.



Source: K-Means Clustering on Wikipedia

K-Means Clustering

K-Means clustering is an **optimisation problem**

Aim:

Mi

(W

$$\arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2 = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} |S_i| \operatorname{Var} S_i$$

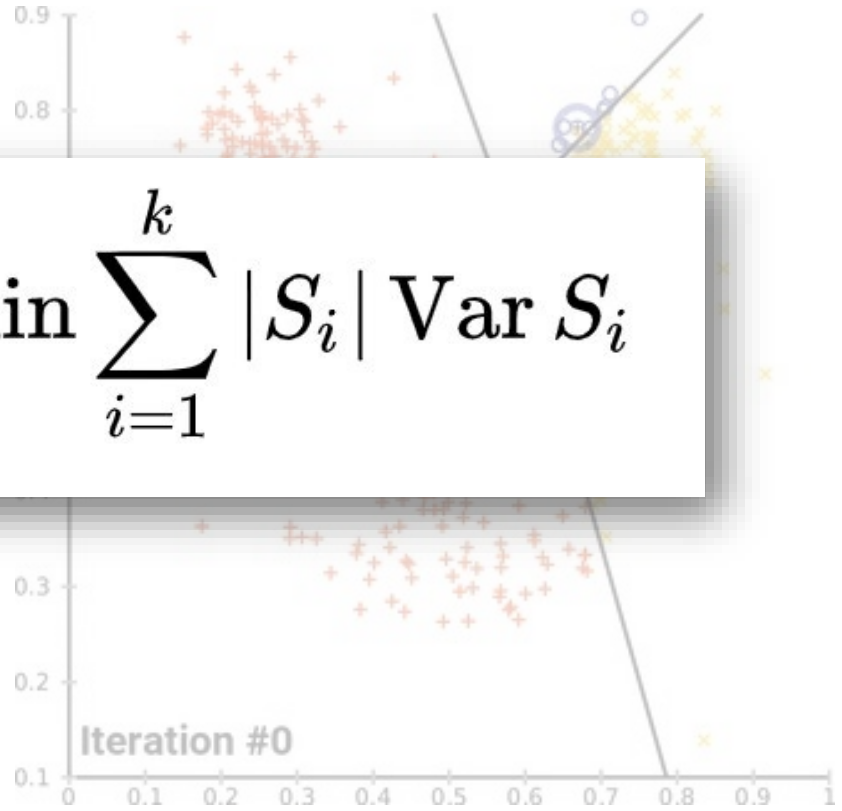
Giv

a d

n observations into k sets S where $k \leq n$:

$$S = \{S_1, S_2, \dots, S_k\}$$

Such that the variance of each subset S_i is minimised.



Source: K-Means Clustering on Wikipedia

How K-Means Works

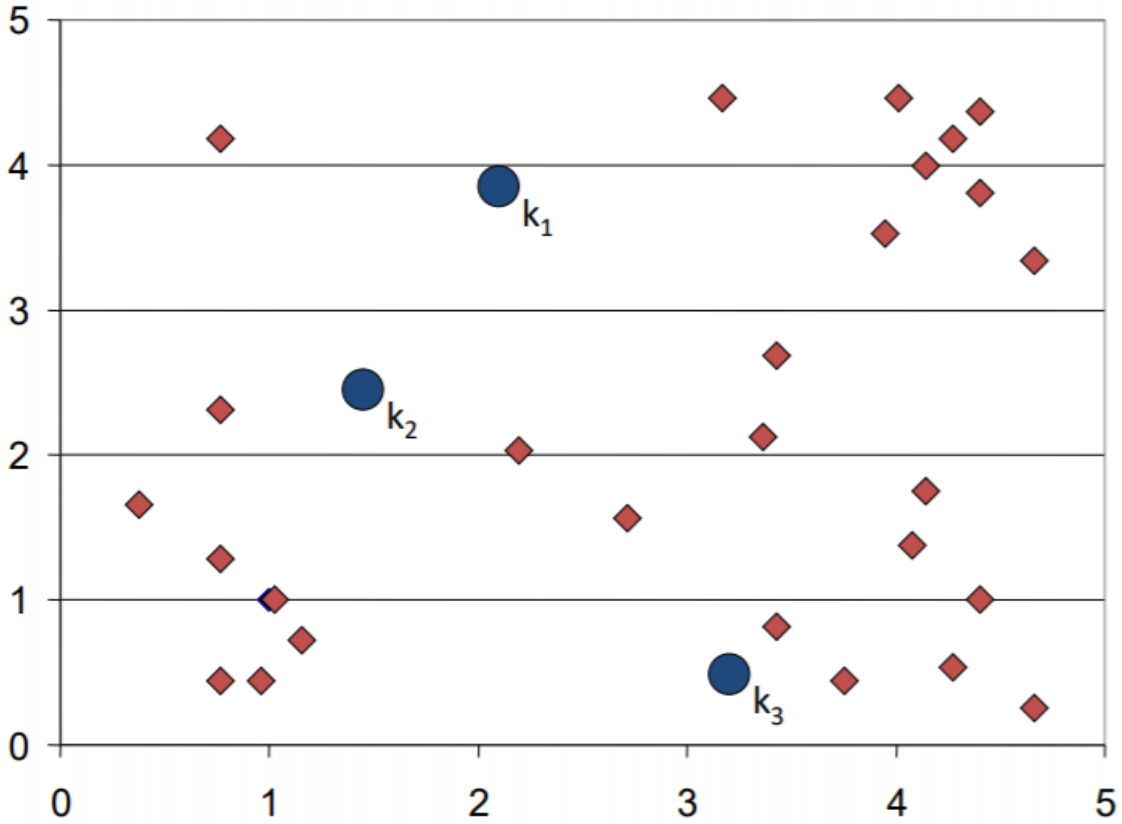
Algorithm:

- Choose k, i.e. the number of clusters
- while true:
  - create k clusters by assigning each point to closest centroid
  - compute k new centroids by averaging points in each cluster
  - if centroids don't change:
    - stop

K-Means Example

Step 1:
Let's choose $k = 3$.

Place the k points at random locations in the data space to form initial centroids

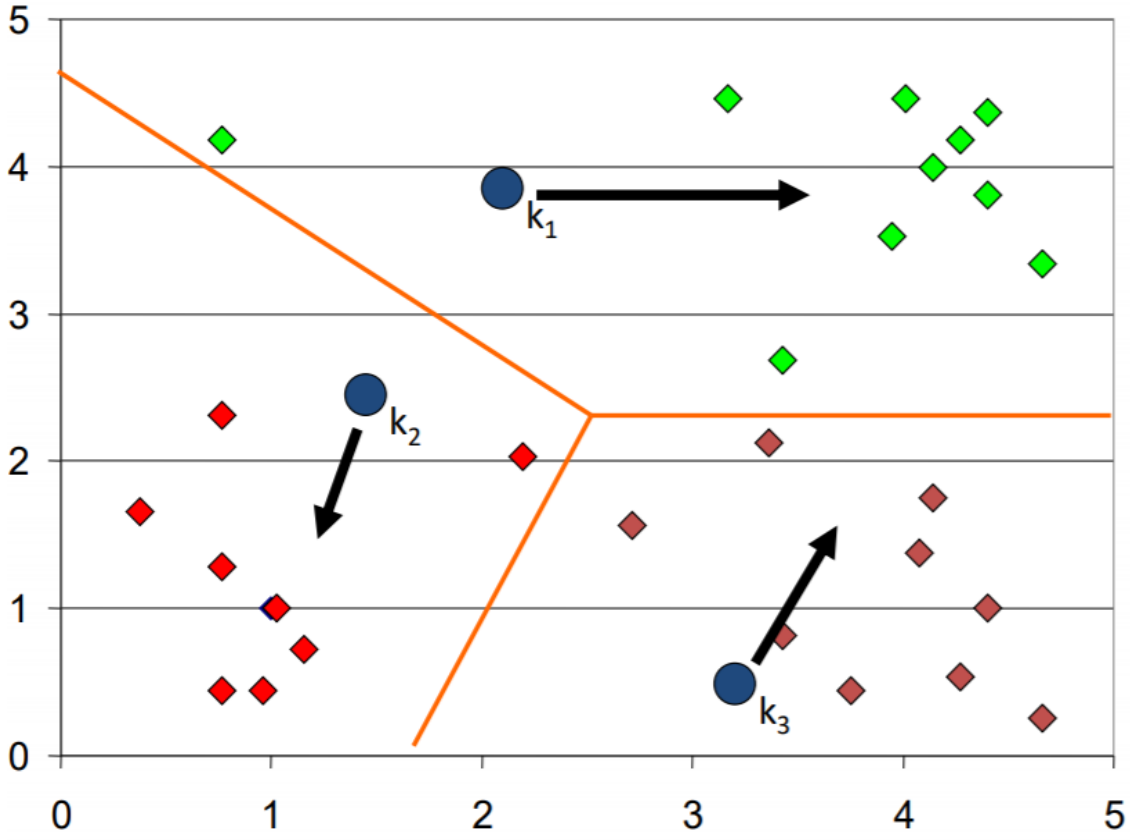


K-Means Example

Step 1:
Let's choose $k = 3$.

Place the k points at random locations in the data space to form initial centroids

Step 2:
Create k clusters by assigning each point to the closest centroid



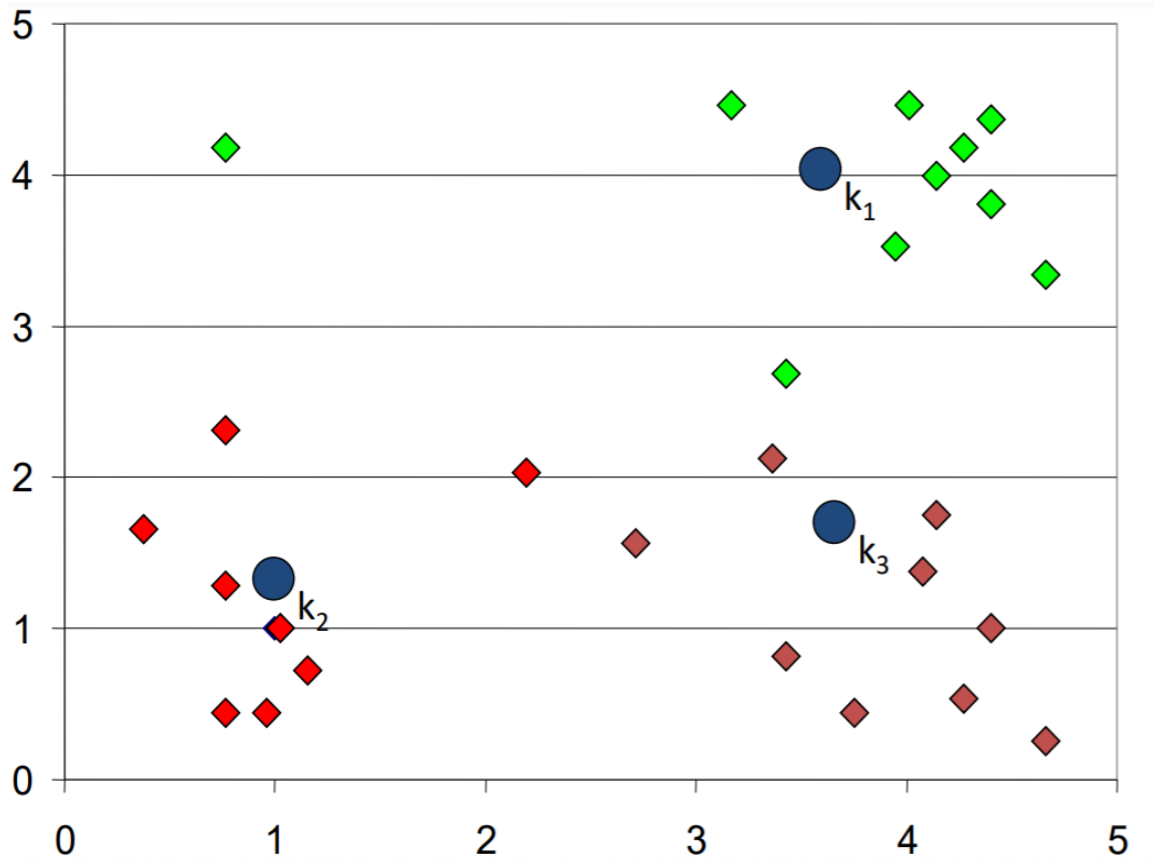
K-Means Example

Step 1:
Let's choose $k = 3$.

Place the k points at random locations in the data space to form initial centroids

Step 2:
Create k clusters by assigning each point to the closest centroid

Step 3:
Compute new centroids by averaging points in each cluster



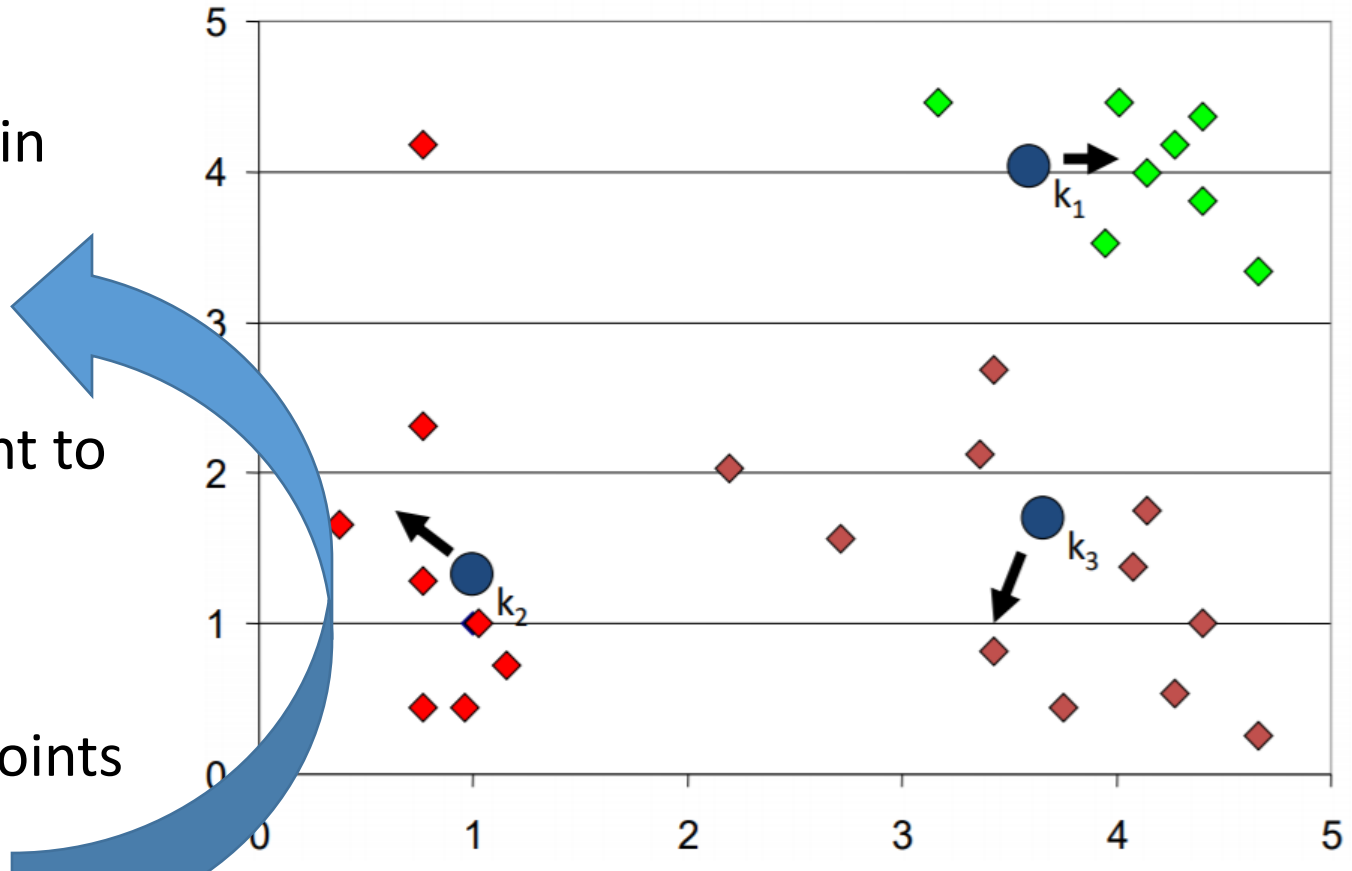
K-Means Example

Step 1:
Let's choose $k = 3$.

Place the k points at random locations in the data space to form initial centroids

Step 2:
Create k clusters by assigning each point to the closest centroid

Step 3:
Compute new centroids by averaging points in each cluster



Repeat

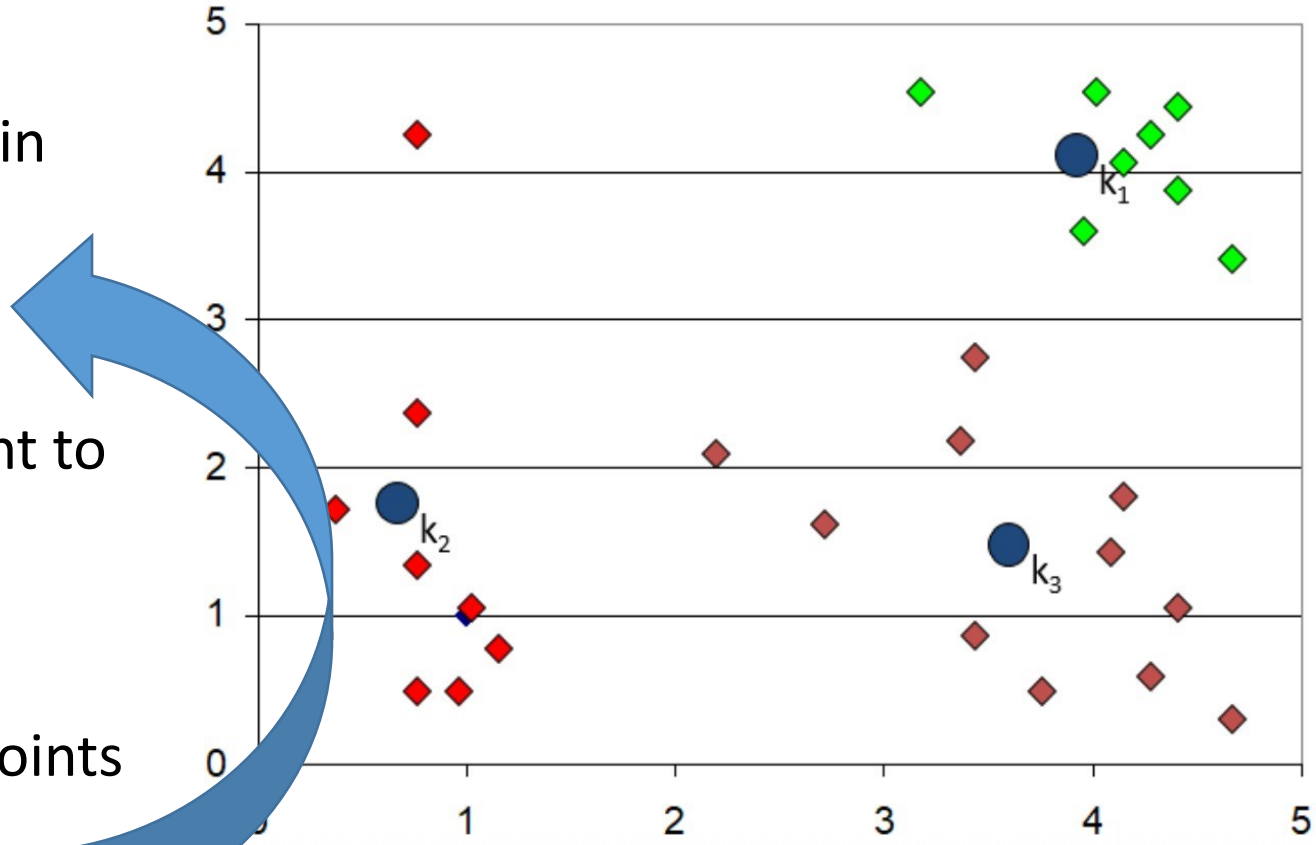
K-Means Example

Step 1:
Let's choose $k = 3$.

Place the k points at random locations in the data space to form initial centroids

Step 2:
Create k clusters by assigning each point to the closest centroid

Step 3:
Compute new centroids by averaging points in each cluster



Repeat

K-Means Summary

Efficiency

- K-Mean is more efficient but has some weaknesses

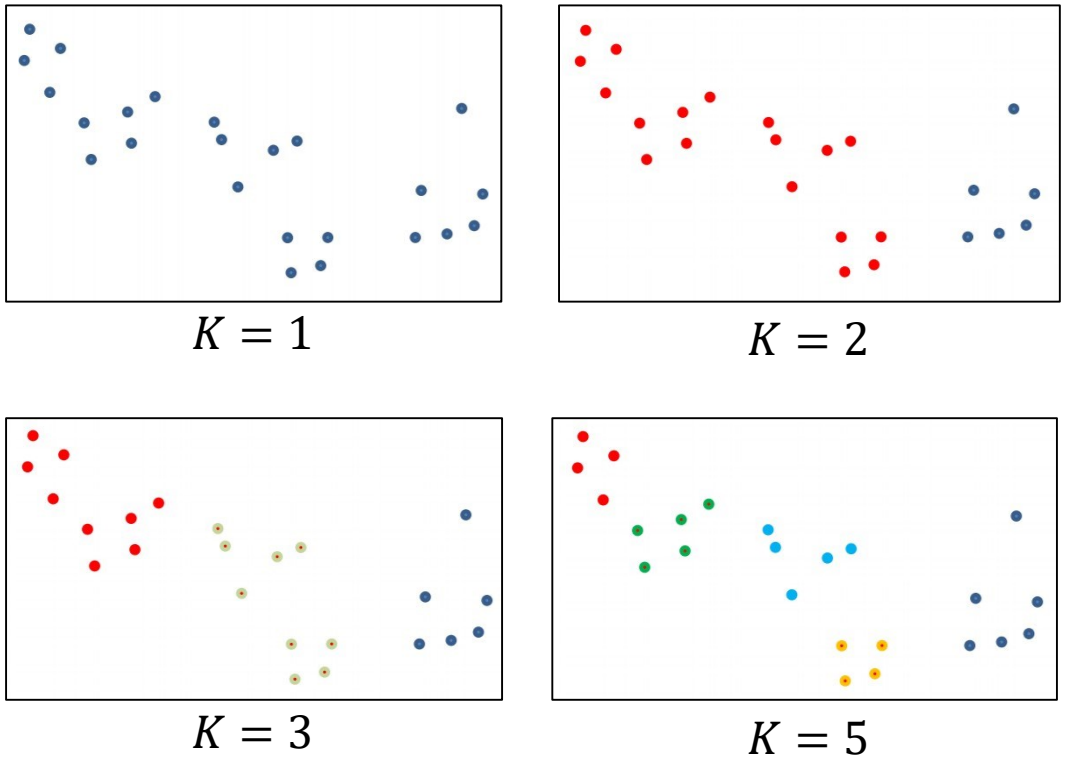
K-Means Summary

Efficiency

- K-Mean is more efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.



How do we choose the right k?

K-Means Summary

Efficiency

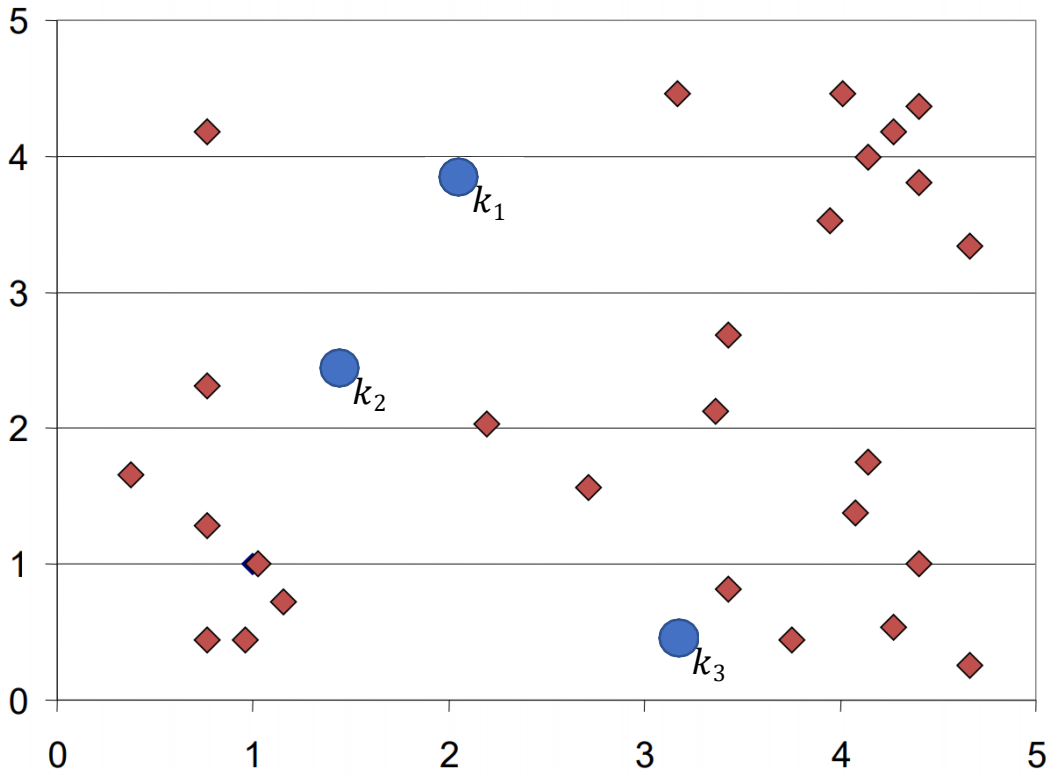
- K-Mean is more efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.

It is non-deterministic

- Initial centroids are chosen at random



K-Means Summary

Efficiency

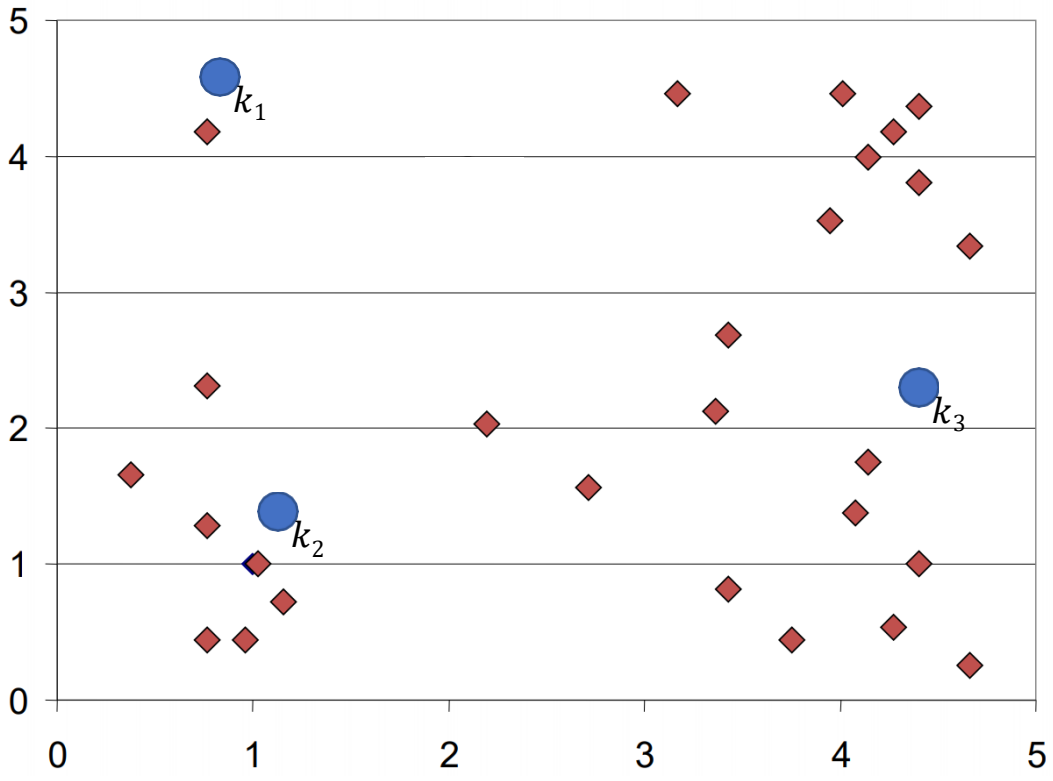
- K-Mean is more efficient but has some weaknesses

Number of Clusters

- You don't necessarily know $\boldsymbol{k}$, i.e. number of clusters
- You can choose "wrong" k and get strange results.

It is non-deterministic

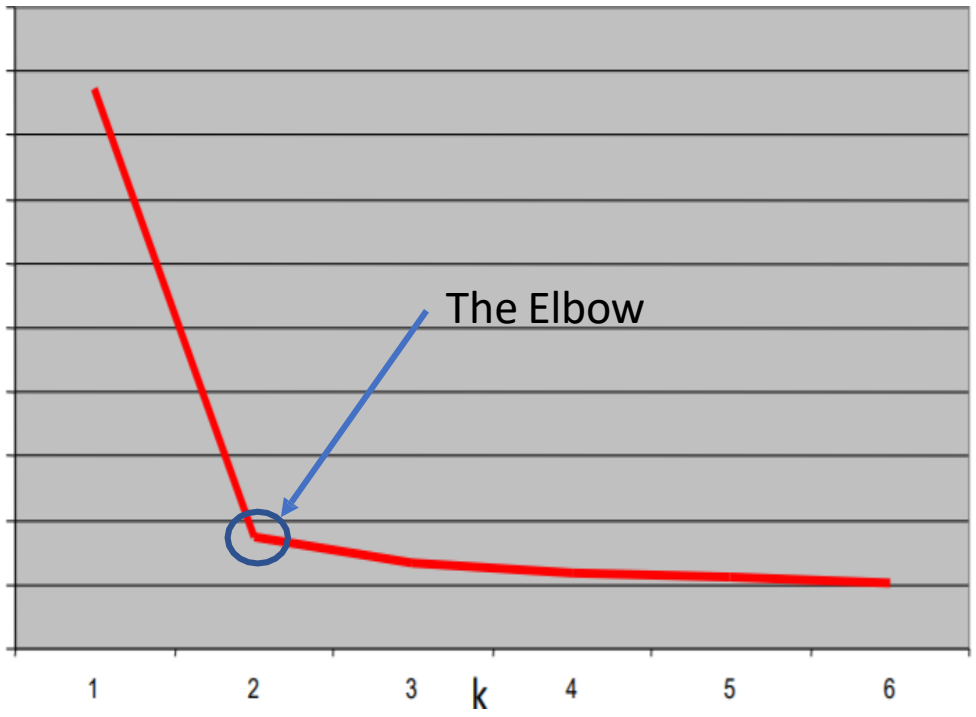
- Initial centroids are chosen at random
- Centroids can be too close



K-Means Summary

How can we choose k ?

- A prior knowledge of the data space can help
  - Three classes of flowers in the Iris dataset
  - Two types of emails: good and spam
- Use the Elbow method
  - Try different values and look for abrupt change in result
- Run hierarchical clustering on subset of data



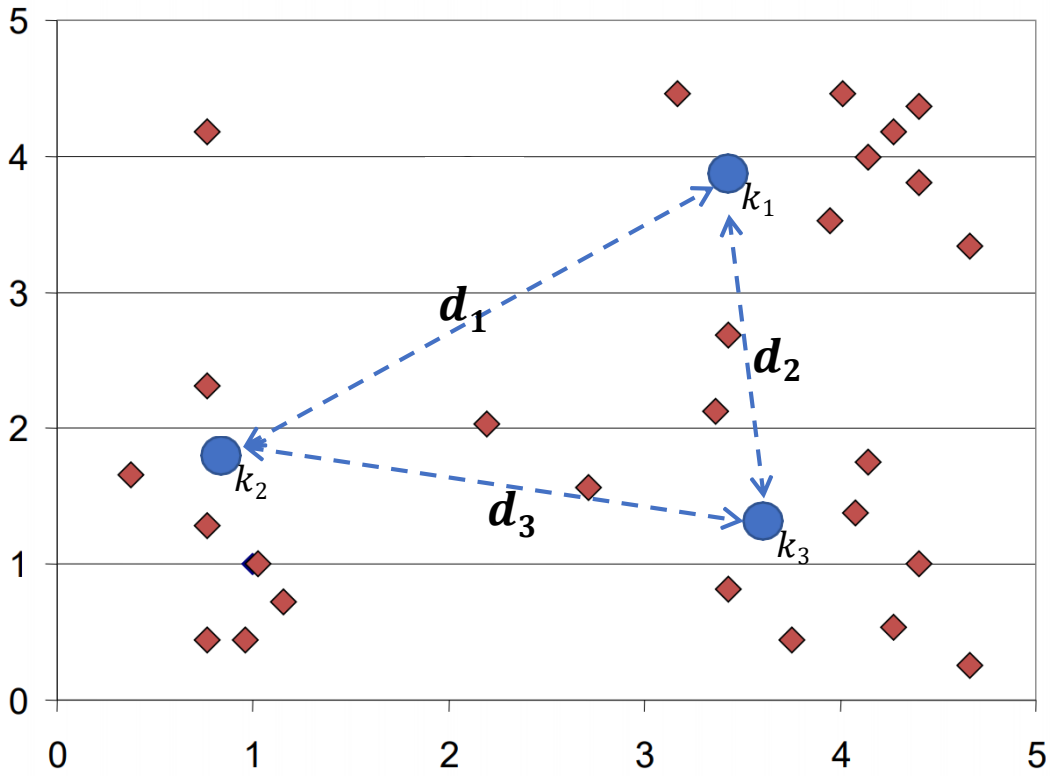
K-Means Summary

Mitigating Initial Centroids Dependency

- Use a random number seed
- Define a minimum distance $\min(d)$ between clusters:

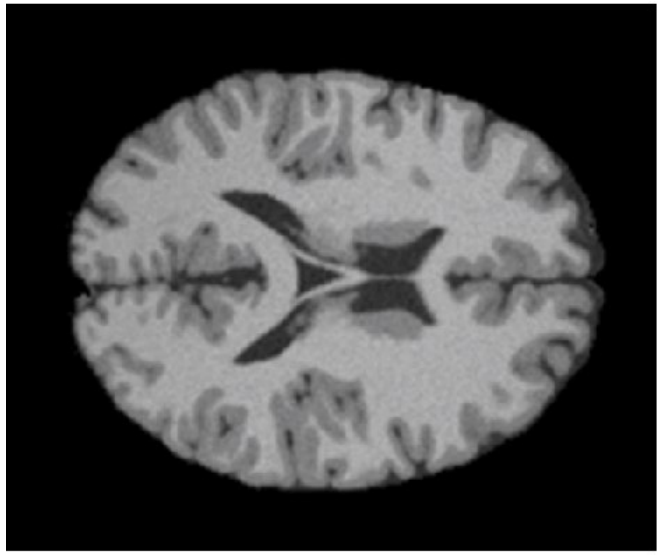
$$d_1, d_2, d_3 \geq \min(d)$$

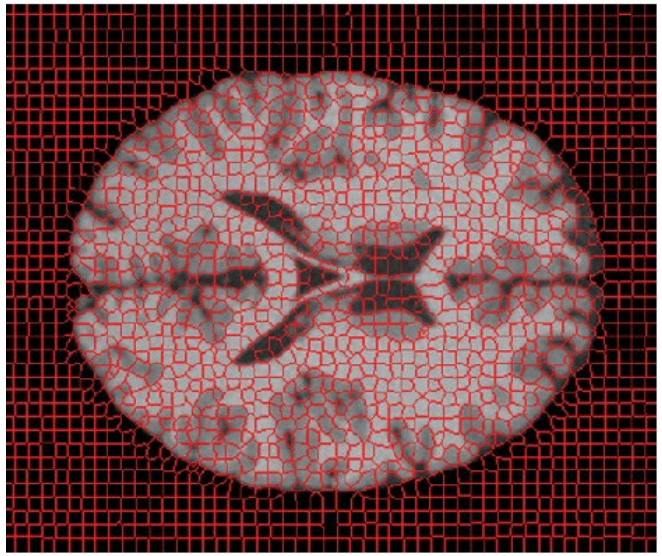
- Define the minimum data points in a cluster.



Applications of Clustering

- Market segmentation
- Social network analysis
- Search results grouping
- Medical imaging
- Genetics and evolutionary biology
- Anomaly detection
- Recommender systems





Classification

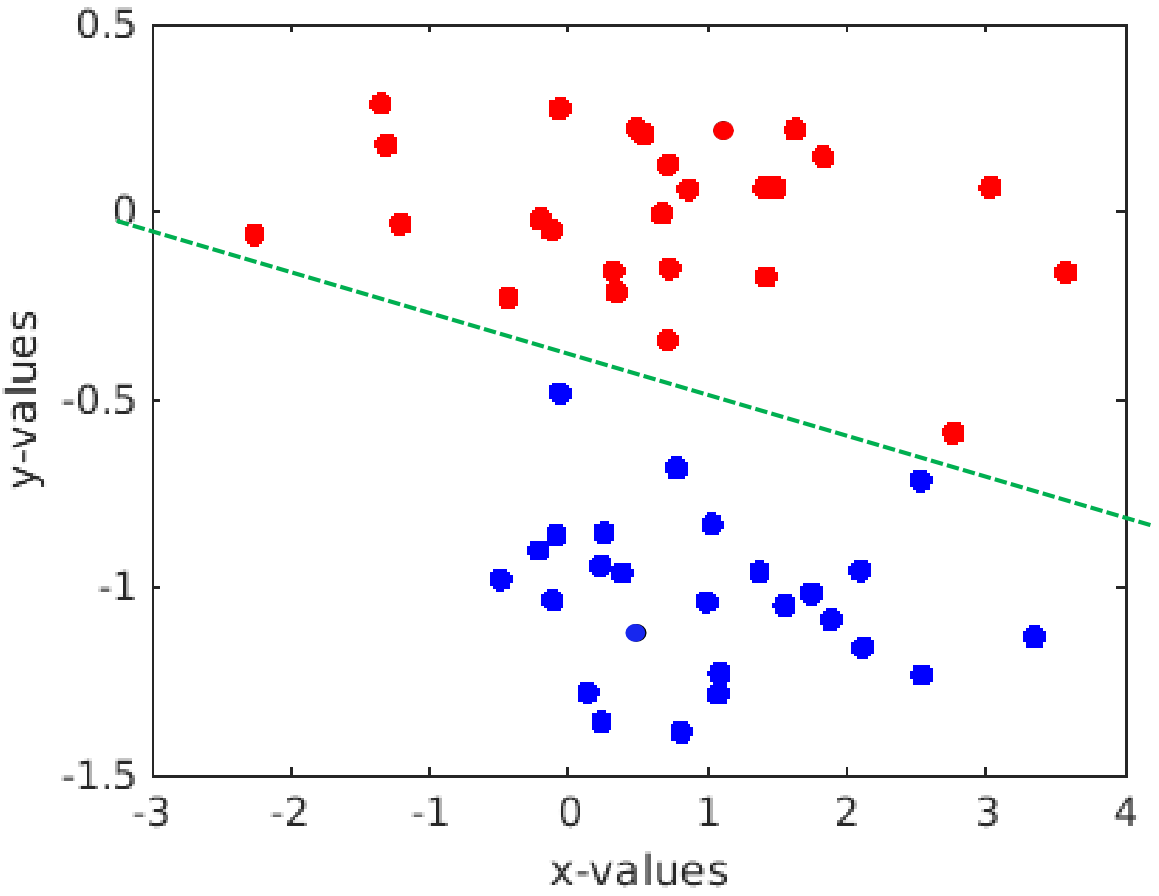
Classification

A supervised learning concept used in building machine learning models that categorise data items into classes

Classifier is trained to specify which of k categories some input belongs to

A classifier is a function:

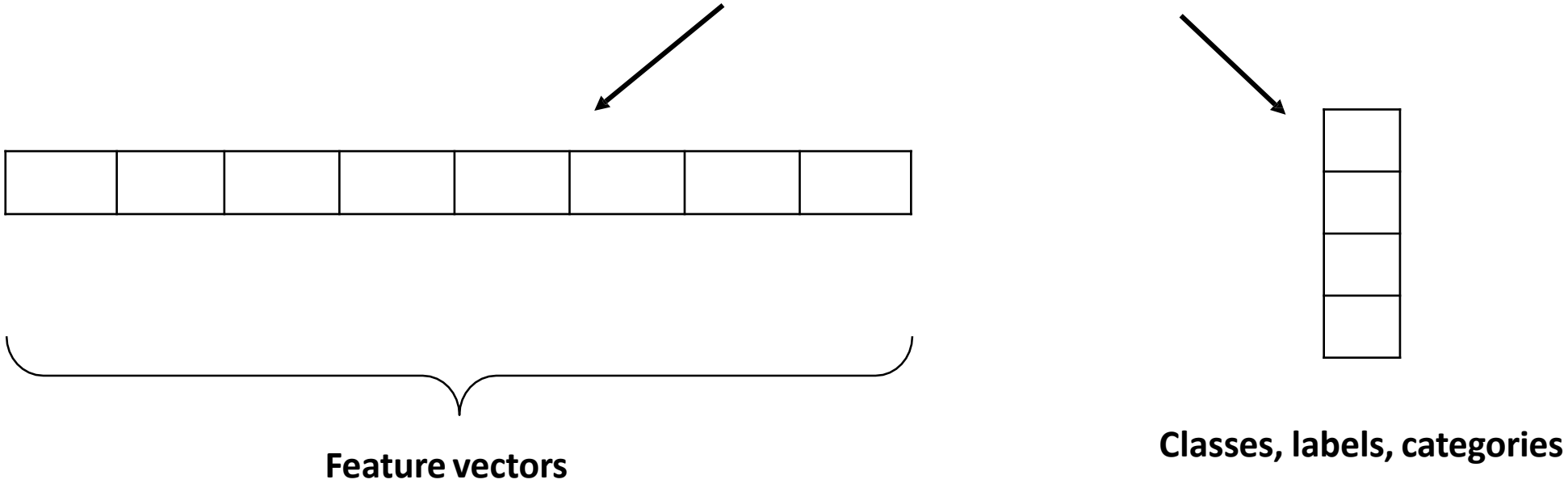
$$f: \mathbb{R}^n \rightarrow \{1, \dots, k\}$$



Classification

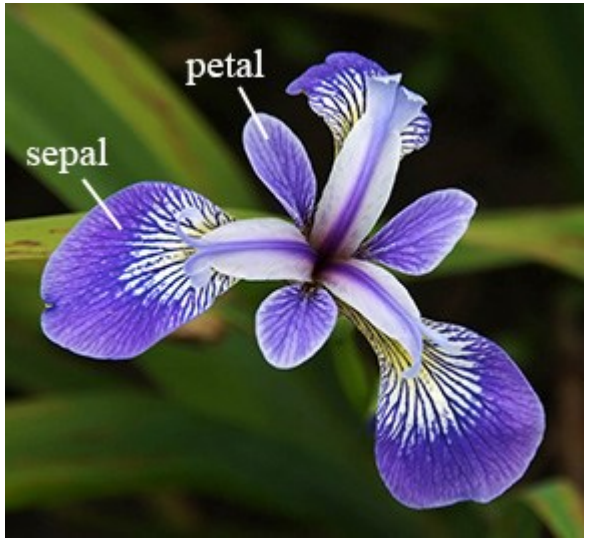
Aim: find f such that $f(x) = y \in \{1, \dots, k\}$ where
x = feature vector and
y = class label

$$f: \mathbb{R}^n \to \{1, \dots, k\}$$



Iris Dataset Example

Aim: find f such that $f(x) = y \in \{1, \dots, k\}$ where
x = feature vector: values of measurements
y = class label: names of species



$$f: \mathbb{R}^n \to \{1, \dots, k\}$$

$$k = 3$$

$$n = 150 \times 4$$

Sepal Length	Sepal Width	Petal Length	Petal Width

Feature vectors

setosa
versicolor
virginica

Classes, labels, categories

Classification Algorithms

Common classification algorithms and their MATLAB functions

Algorithm	MATLAB function
K-Nearest Neighbour	`fitcknn()`
SVM Binary classification	`fitcecoc()`
SVM Multiclass	`fitcecoc()`
Decision Tree	`fitctree()`
Naïve Bayes	`fitcnb()`
Discriminant Analysis	`fitcdiscr()`
Ensembles	`fitcensemble()`

K-Nearest Neighbour

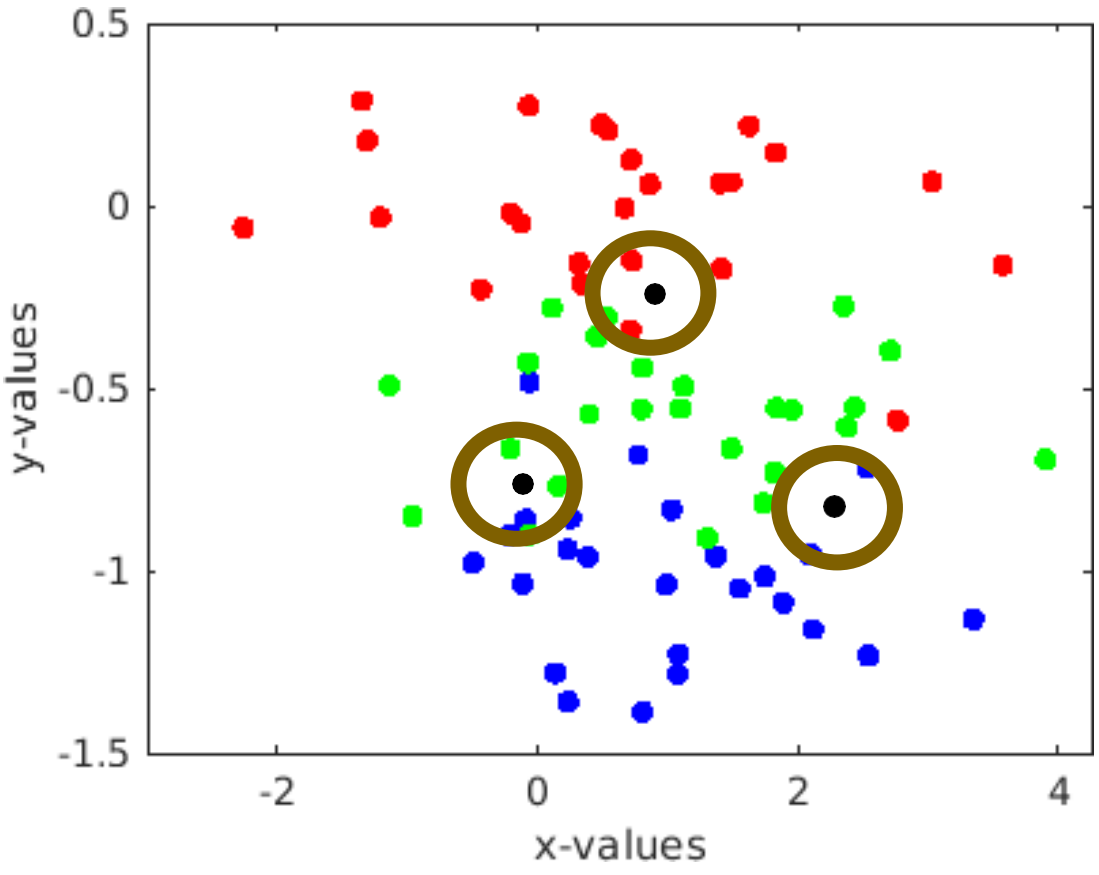
Given **training data** and **a test point**

Look at the k most similar examples

Assign the majority class label k = number of nearest neighbours to search for

Special case: $k = 1$

- 1 nearest neighbour

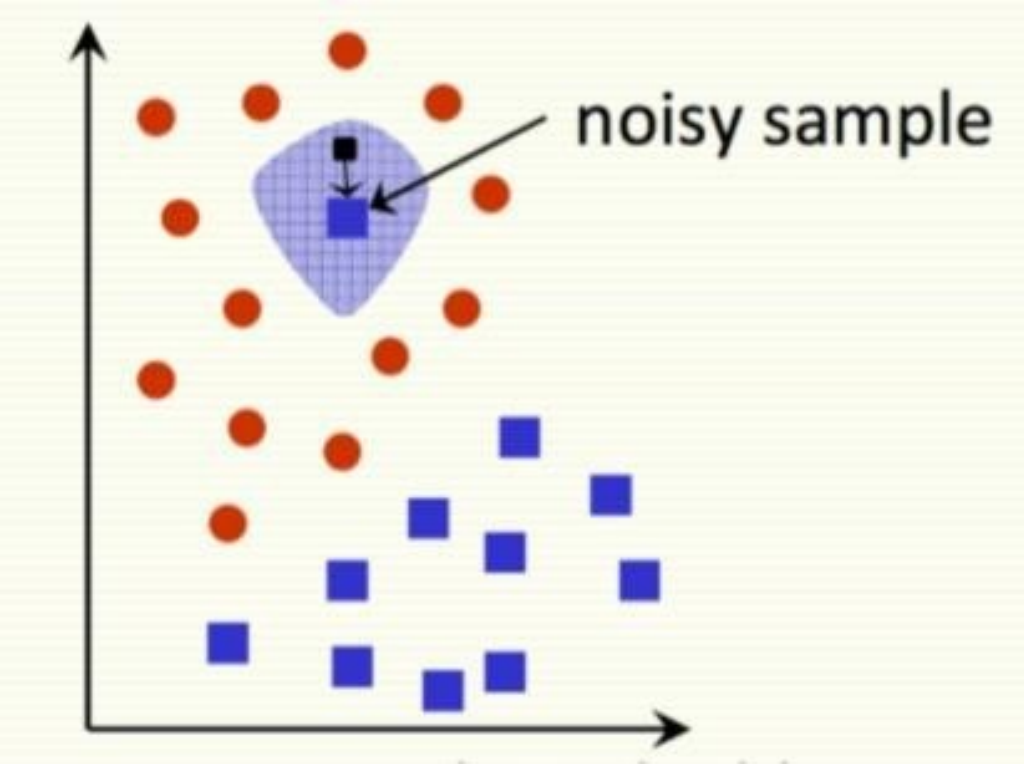


1-Nearest Neighbour

1-NN is sensitive to mis-labelled data

Every example in the blue shaded area will be misclassified as the blue class

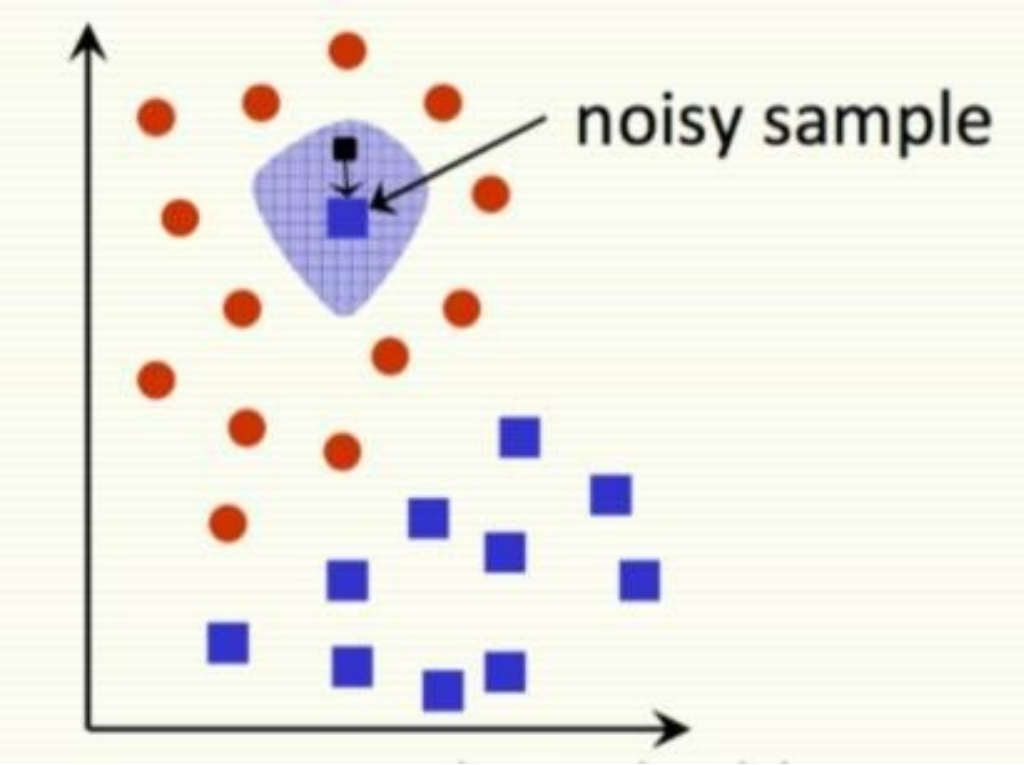
Solution? Increase k



3-Nearest Neighbour

3-NN reduces the classification error

Every example in the blue shades will now be classified correctly as the red class



1-Nearest Neighbour

Iris Example

Using 1-Nearest Neighbour and all of the measures, can we determine the species of the 4th row?

Distance: Euclidean

Sepal Length (m_1)	Sepal Width (m_2)	Petal Length (m_3)	Petal Width (m_4)	Species
2.3	4.3	1.2	2.4	setosa
3.4	6.4	4.5	2.3	Versicolor
6.4	4.2	2.1	1.2	virginica
2.5	3.5	2.7	2.1	???

1-Nearest Neighbour

Iris Example

Using 1-Nearest Neighbour and all of the measures, can we determine the species of the 4th row?

Distance: Euclidean

Distance to 4th Row	Sepal Length (m_1)	Sepal Width (m_2)	Petal Length (m_3)	Petal Width (m_4)	Species
1.73	2.3	4.3	1.2	2.4	setosa
3.54	3.4	6.4	4.5	2.3	Versicolor
4.11	6.4	4.2	2.1	1.2	virginica
0	2.5	3.5	2.7	2.1	???

1-Nearest Neighbour

Iris Example

Using 1-Nearest Neighbour and all of the measures, can we determine the species of the 4th row?

Distance: Euclidean

What if we only use measures m_3 and m_4

Distance to 4th Row	Sepal Length (m_1)	Sepal Width (m_2)	Petal Length (m_3)	Petal Width (m_4)	Species
1.53	2.3	4.3	1.2	2.4	setosa
1.81	3.4	6.4	4.5	2.3	Versicolor
1.08	6.4	4.2	2.1	1.2	virginica
0	2.5	3.5	2.7	2.1	???

Application of Classification

Image recognition

- Face detection, optical character recognition

Sentiment analysis

- aka opinion mining e.g. politics, product reviews

Text classification

- e.g. classifying news to topics (technology, sports, entertainment)

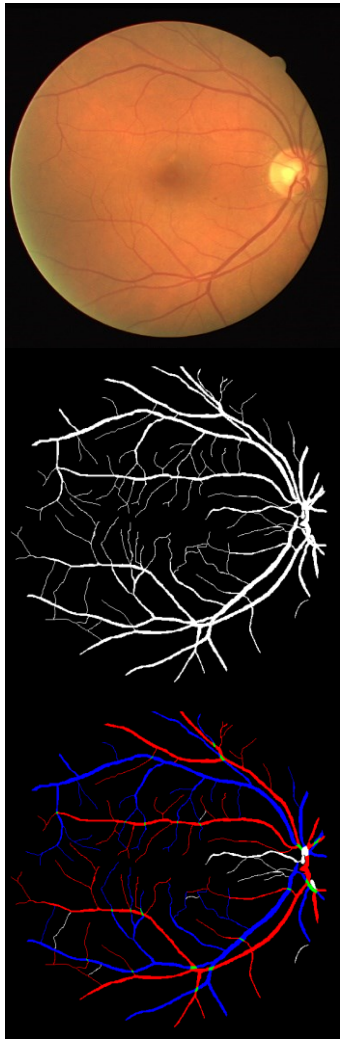
Email Classification and Spam filtering

- sorting emails into appropriate folders and removing spams

Authorship attribution

- identifying writing styles of different authors.

sentiment	tweets
MIXED	168
NEGATIVE	1956
NEUTRAL	5472
POSITIVE	854



SCC361: Artificial Intelligence

Week 4: Clustering and Classification

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

Measures of Success and Error Measures

Machine Learning Paradigm

Training Data
The observed set of example data

Test Data
Previously unseen data generated by the same process as the training data

Training process

Inferring or learning something about the process that generated the training data

Prediction

Using the inference to make predictions about the test data

Evaluation

Comparing the prediction outputs with the correct test data outputs

Evaluation

Suppose we have a method f and a test set $S_x = \{X_1, \dots, X_n\}$ where $X_i = (\boxed{x_i}, \boxed{l_i})$

data

label

Our method estimates the label as

$$\widehat{l_i} = f(x_i)$$

If the model is suitable, we should have

$$\widehat{l_i} = l_i \text{ for all } i = 1, \dots, n$$

$$Accuracy = \frac{Number\ of\ test\ elements\ where\ the\ estimated\ label\ is\ correct}{Number\ of\ test\ elements}$$

Accuracy

Accuracy is a measure of how close your estimate is to the actual value

For classification with categorical labels:

Accuracy = ratio of **correct estimates** to **all estimates**

$$Accuracy = \frac{Number\ of\ test\ elements\ where\ the\ estimated\ label\ is\ correct}{Number\ of\ test\ elements}$$

$$Accuracy = \frac{\sum_i b(l_i, \widehat{l_i})}{|S_x|} \text{ where } b(l_i, \widehat{l_i}) = \begin{cases} 1 \text{ if } l_i = \widehat{l_i} \\ 0 \ if \ l_i \neq \widehat{l_i} \end{cases}$$

Accuracy

Example

x_1: $l_1 = \text{"dog"}$ $\widehat{l_1} = f(x_1) = \text{"dog"}$ ✓ $b(l_1, \widehat{l_1}) = 1$

x_2: $l_2 = \text{"dog"}$ $\widehat{l_2} = f(x_2) = \text{"dog"}$ ✓ $b(l_2, \widehat{l_2}) = 1$

x_3: $l_3 = \text{"cat"}$ $\widehat{l_3} = f(x_3) = \text{"cat"}$ ✓ $b(l_3, \widehat{l_3}) = 1$

x_4: $l_4 = \text{"cat"}$ $\widehat{l_4} = f(x_4) = \text{"dog"}$ ✗ $b(l_4, \widehat{l_4}) = 0$

x_5: $l_5 = \text{"cat"}$ $\widehat{l_5} = f(x_5) = \text{"cat"}$ ✓ $b(l_5, \widehat{l_5}) = 1$

$$Accuracy = \frac{\sum_i b(l_i, \widehat{l_i})}{|S_x|}$$

$$|S_x| = 5$$

$$\sum_i b(l_i, \widehat{l_i}) = 1 + 1 + 1 + 0 + 1 = 4$$

$$Accuracy = \frac{4}{5} = 0.8 = 80\%$$

Accuracy

Non-categorical

What if model output is not categorical?

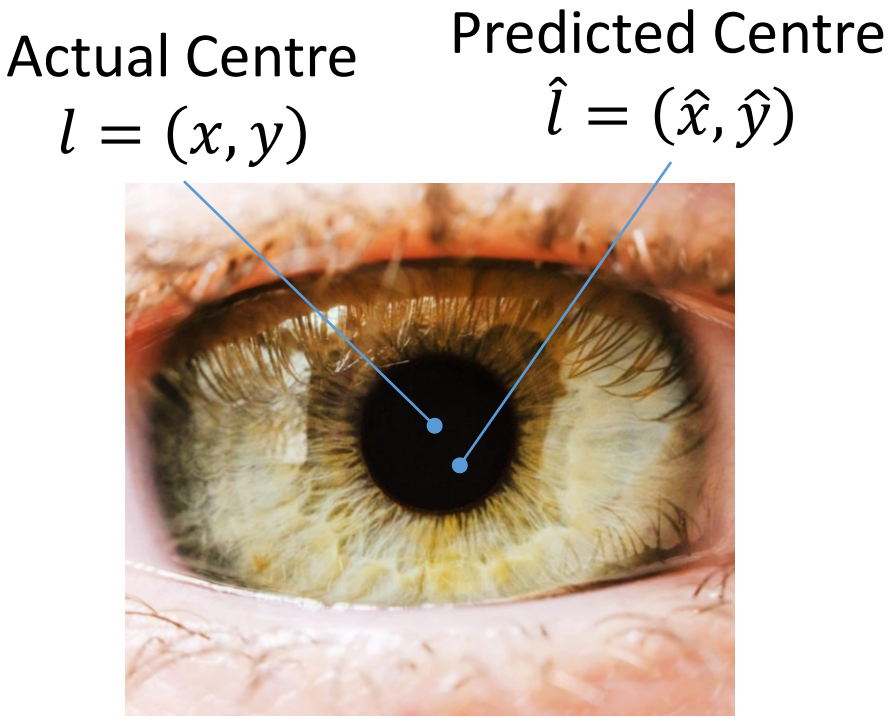
Example: determine location of centre of pupil

More meaningful: **distance** such as L^2-norm:

$$Accuracy = \left\| l - \hat{l} \right\|_2 = \sqrt{(x - \hat{x})^2 + (y - \hat{y})}$$

E.g. $l = (95,101)$, $\hat{l} = (100,103)$

$$Accuracy = \left\| l - \hat{l} \right\|_2 = \sqrt{(95 - 100)^2 + (101 - 103)}$$
$$= \sqrt{(-5)^2 + (-2)^2} = \sqrt{25 + 4} = \sqrt{29} \approx 5.385$$



Confusion Matrices

- Accuracy is important as an **overall** view.

Example:

- Suppose we are developing a new automated technique for Glaucoma diagnosis
- Our technique is 95% accurate on a test set of 10,000 people
- It is wrong for 5%
  - But which ones???

Healthy eyes

Periphial vision loss due to glaucoma

Confusion Matrices

- Confusion Matrices give us more insight.
  - **TP**: We predicted **positive** and we were **right**
  - **TN**: We predicted **negative** and we were **right**
  - **FP**: We predicted **positive** and we were **wrong**
  - **FN**: We predicted **negative** and we were **wrong**

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Examples:

- Positive = Glaucoma, Negative = Healthy
- Positive = Cats, Negative = Dogs

Confusion Matrices

Glaucoma (GL) Diagnosis Example:

- Our technique is 95% accurate on a test set of 10,000 people
- Our test set includes
  - 500 GL Patients
  - 9500 Non-GL Patients
- Scenario 1: Our results:
  - All GL Patients diagnosed correctly
  - 9000 Healthy patients diagnosed correctly
  - 500 Healthy patients mis-diagnosed

		True Class	
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.95$$

Confusion Matrices

Glaucoma (GL) Diagnosis Example:

- Our technique is 95% accurate on a test set of 10,000 people
- Our test set includes
  - 500 GL Patients
  - 9500 Non-GL Patients
- Scenario 2: Our results:
  - All Healthy Patients diagnosed correctly
  - All DL patients mis-diagnosed

		True Class	
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.95$$

Confusion Matrices

What other measures can we consider?

- **Sensitivity**, recall, hit rate, true positive rate
- **Specificity**, selectivity, true negative rate
- **Precision**, positive predicted value
- **Negative predictive value**
- Miss rate, false negative rate
- Fall out, false positive rate
- False discovery rate
- False omission rate
- **F1 score**, dice coefficient

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Sensitivity and Specificity

Sensitivity: Probability of *positive* outcome if truly positive

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{|True\ Positives|}$$

Specificity: Probability of *negative* outcome if truly *negative*

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{|True\ Negatives|}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Sensitivity and Specificity

Example

Scenario 1	True Class		
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

$$Sensitivity = \frac{TP}{TP + FN} = \frac{500}{500 + 0} = 1$$

$$Specificity = \frac{TN}{TN + FP} = \frac{9000}{9000 + 500} \approx 0.95$$

Scenario 2	True Class		
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

$$Sensitivity = \frac{TP}{TP + FN} = \frac{0}{500 + 0} = 0$$

$$Specificity = \frac{TN}{TN + FP} = \frac{9500}{9500 + 0} = 1$$

Sensitivity and Specificity

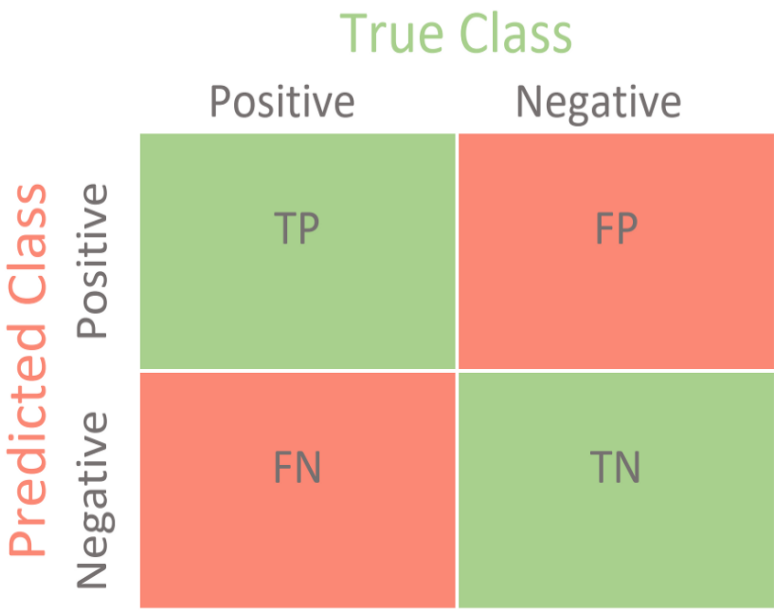
Sensitivity: Probability of *positive* outcome if truly positive

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{|True\ Positives|}$$

Specificity: Probability of *negative* outcome if truly *negative*

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{|True\ Negatives|}$$

- Usually consider sensitivity and specificity together.
- Aim for high sensitivity and high specificity.
- Can use trade-off thresholds.



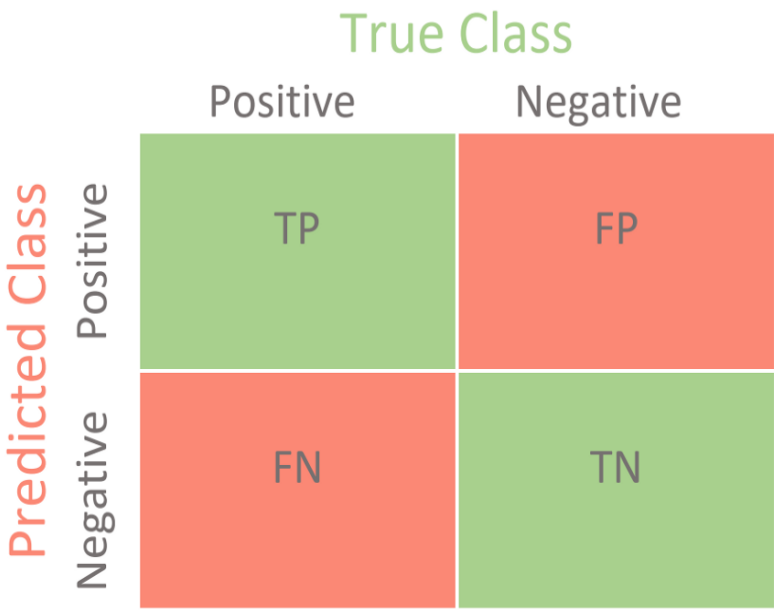
Precision and Negative Predicted Value (NPV)

Precision: Proportion of *truly positive* outcomes to *positive predictions*

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{|Predicted\ Positives|}$$

NPV: Proportion of *truly negative* outcomes to *predicted negative*

$$NPV = \frac{TN}{TN + FN} = \frac{TN}{|Predicted\ Negatives|}$$



Precision and Negative Predicted Value (NPV)

Example

Scenario 1	True Class		
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

$$Precision = \frac{TP}{TP + FP} = \frac{500}{500 + 500} = 0.5$$

$$NPV = \frac{TN}{TN + FN} = \frac{9000}{9000 + 0} = 1$$

Scenario 2	True Class		
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

$$Precision = \frac{TP}{TP + FP} = \frac{0}{0 + 0} = undefined$$

$$NPV = \frac{TN}{TN + FN} = \frac{9500}{9500 + 500} = 0.95$$

Precision and Negative Predicted Value (NPV)

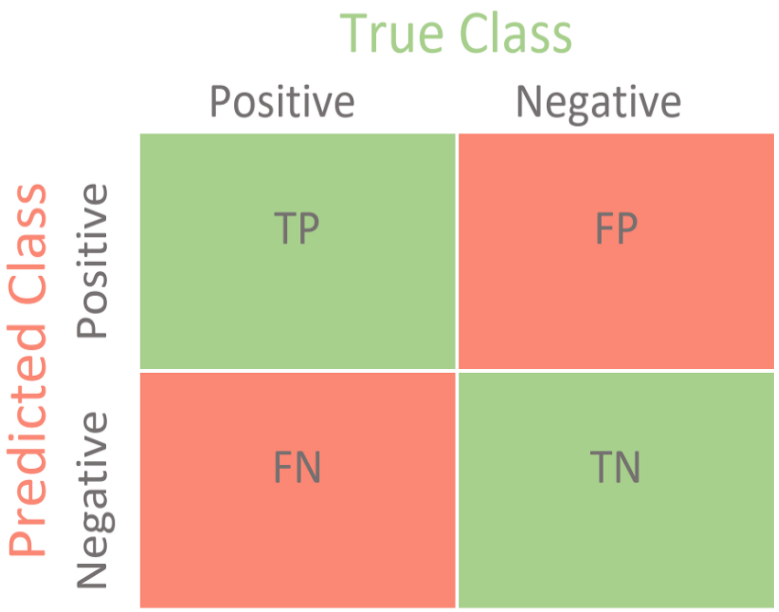
Precision: Proportion of *truly positive* outcomes to *positive predictions*

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{|Predicted\ Positives|}$$

NPV: Proportion of *truly negative* outcomes to *predicted negative*

$$NPV = \frac{TN}{TN + FN} = \frac{TN}{|Predicted\ Negatives|}$$

- Should consider precision and NPV together.
- Aim for high precision and NPV.
- Often consider **precision and recall** (sensitivity) together



F_1 Score

It can be difficult to compare a pair of values

Aim to reduce to a single value

We use harmonic mean when interested in average *rate*

F_1 score is the harmonic mean of precision and recall

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2\frac{precision \cdot recall}{precision + recall}$$

$$= \frac{2TP}{2TP + FP + FN}$$

For our examples:

Example 1: $F_1 = \frac{2 \cdot 500}{2 \cdot 500 + 500 + 0} = \frac{1000}{1500} \approx 0.66$

Example 2: $F_1 = \frac{2 \cdot 0}{2 \cdot 0 + 0 + 500} = 0$

Confusion Matrices

Back to our example

Measure	Scenario 1	Scenario 2
TP	500	0
TN	9000	9500
FP	500	0
FN	0	500
Accuracy	0.95	0.95
Sensitivity	1	0
Specificity	0.947	1
Precision	0.5	Undefined
NPV	1	0.95
F1 Score	0.667	0

Scenario 1	**True Class**		
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

Scenario 2	**True Class**		
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

Non-Binary Confusion Matrices

Compute accuracy in the same way.

There exist generalised versions of F1 Score etc but this is not covered here.

Discriminant Analysis Acc=50.29

True Class \ Predicted Class	1	2	3	4
1	3	129	9	16
2	7	806	33	39
3	1	267	19	9
4	2	340	8	42

Example: Clinical Test

Cup to disc ratio

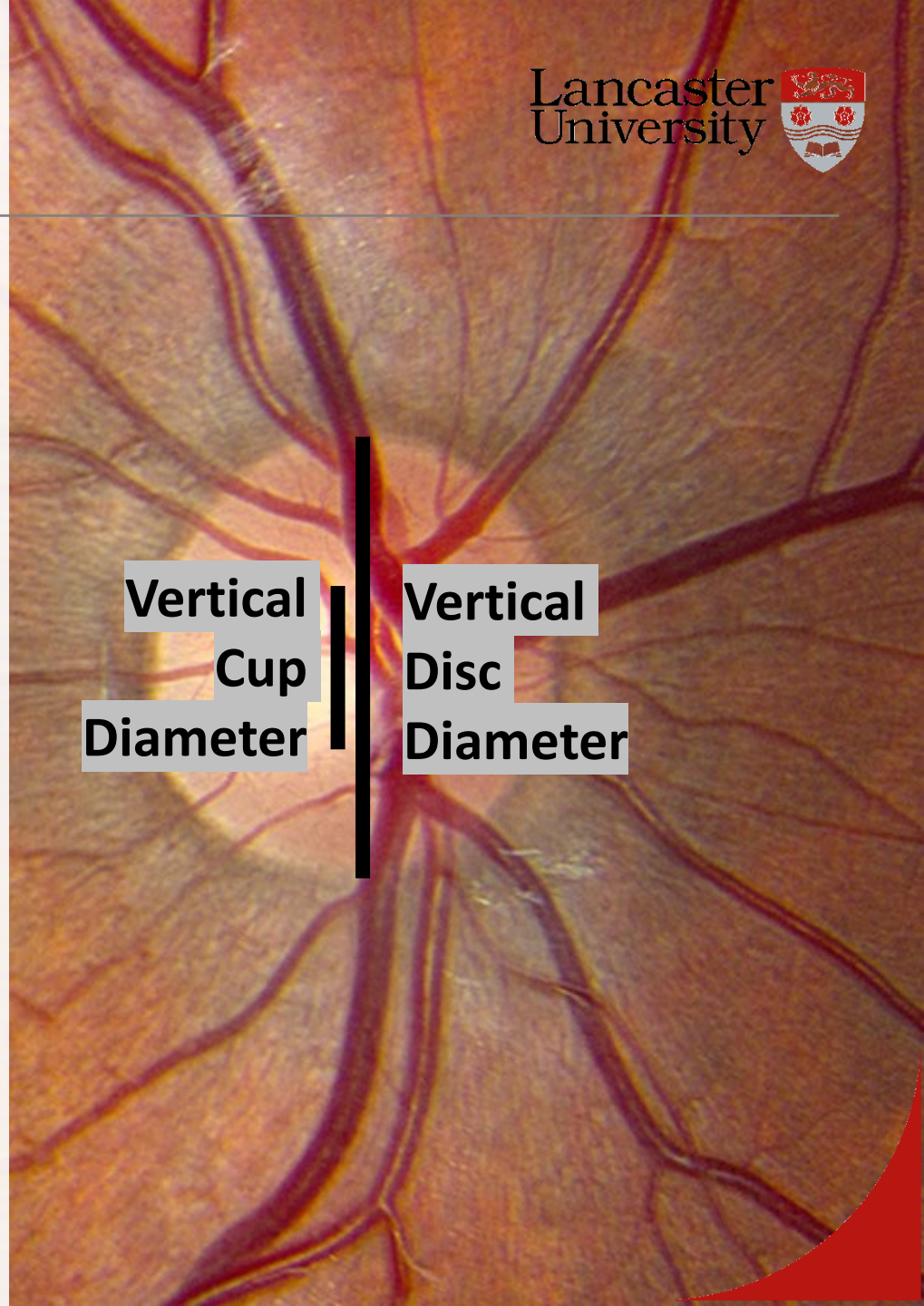
$$cdr = \frac{vertical\ cup\ diameter}{vertical\ disc\ diameter}$$

Rule:

$$cdr \geq 0.7 \Rightarrow Glaucoma$$

$$cdr < 0.7 \Rightarrow Healthy$$

Is this any good?



Example: Clinical Test

Test: 650 Cases (168 GL, 482 Healthy)

Threshold of 0.7:

Accuracy: 0.78

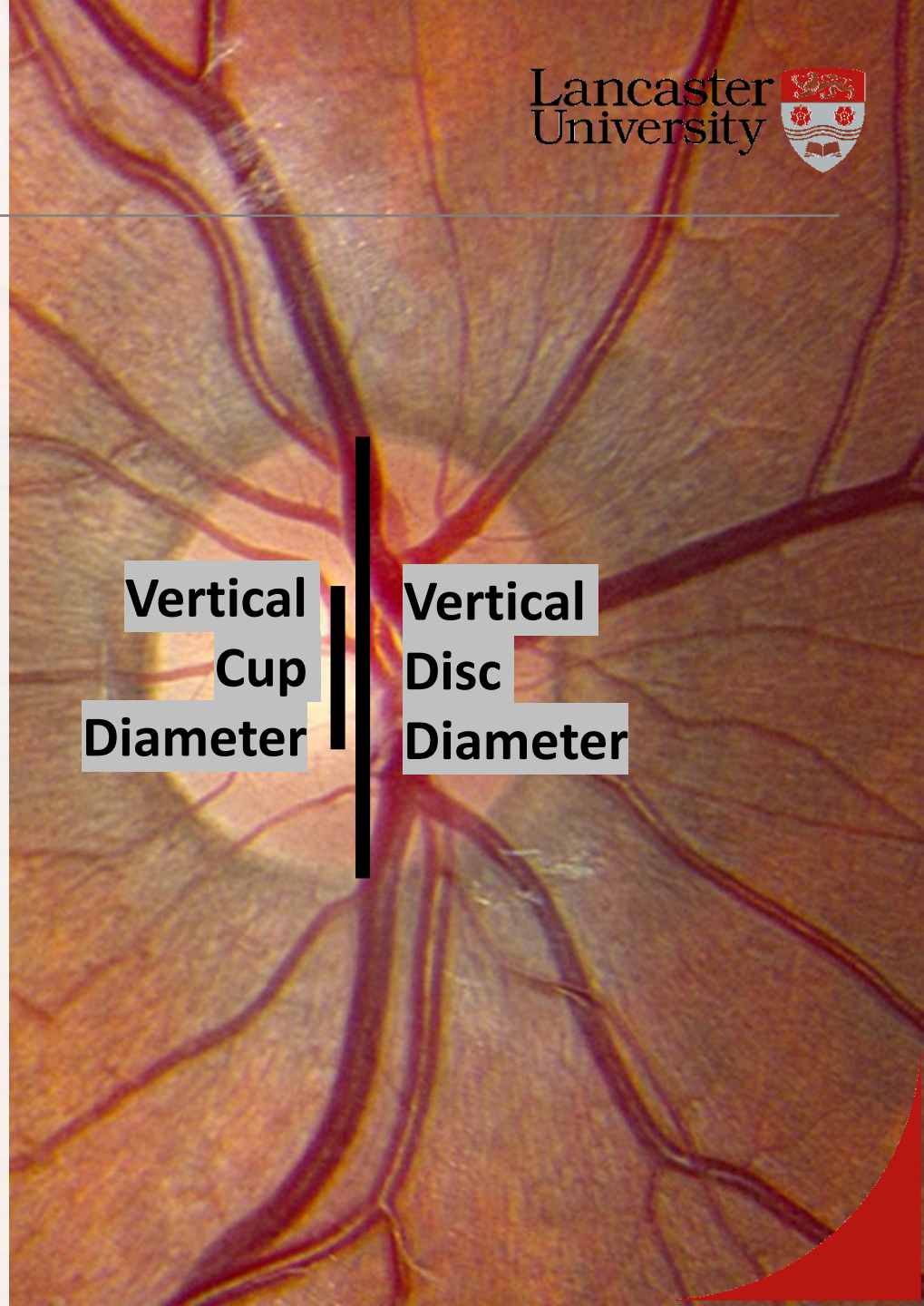
Sensitivity: 0.35

Specificity: 0.94

Precision: 0.66

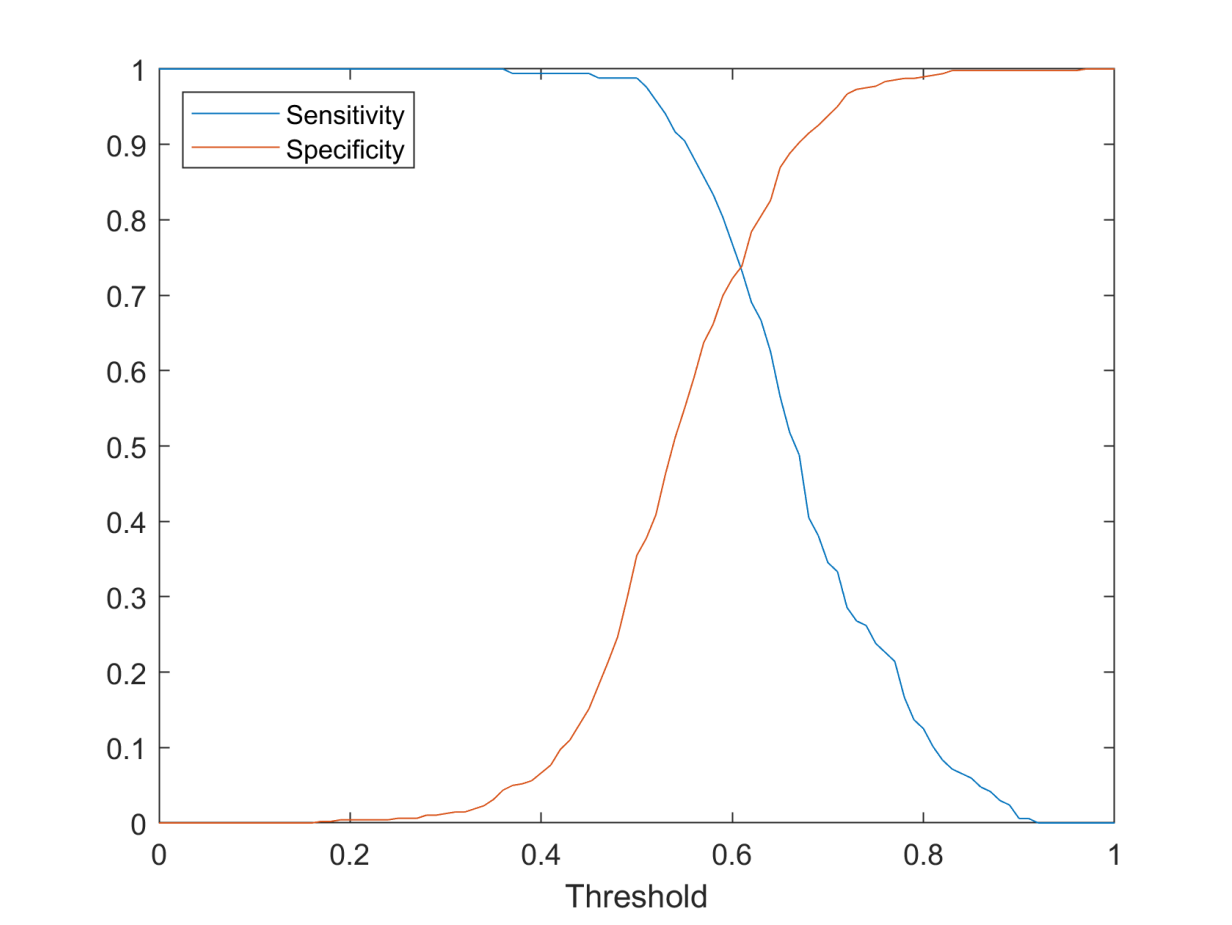
F_1 Score: 0.45

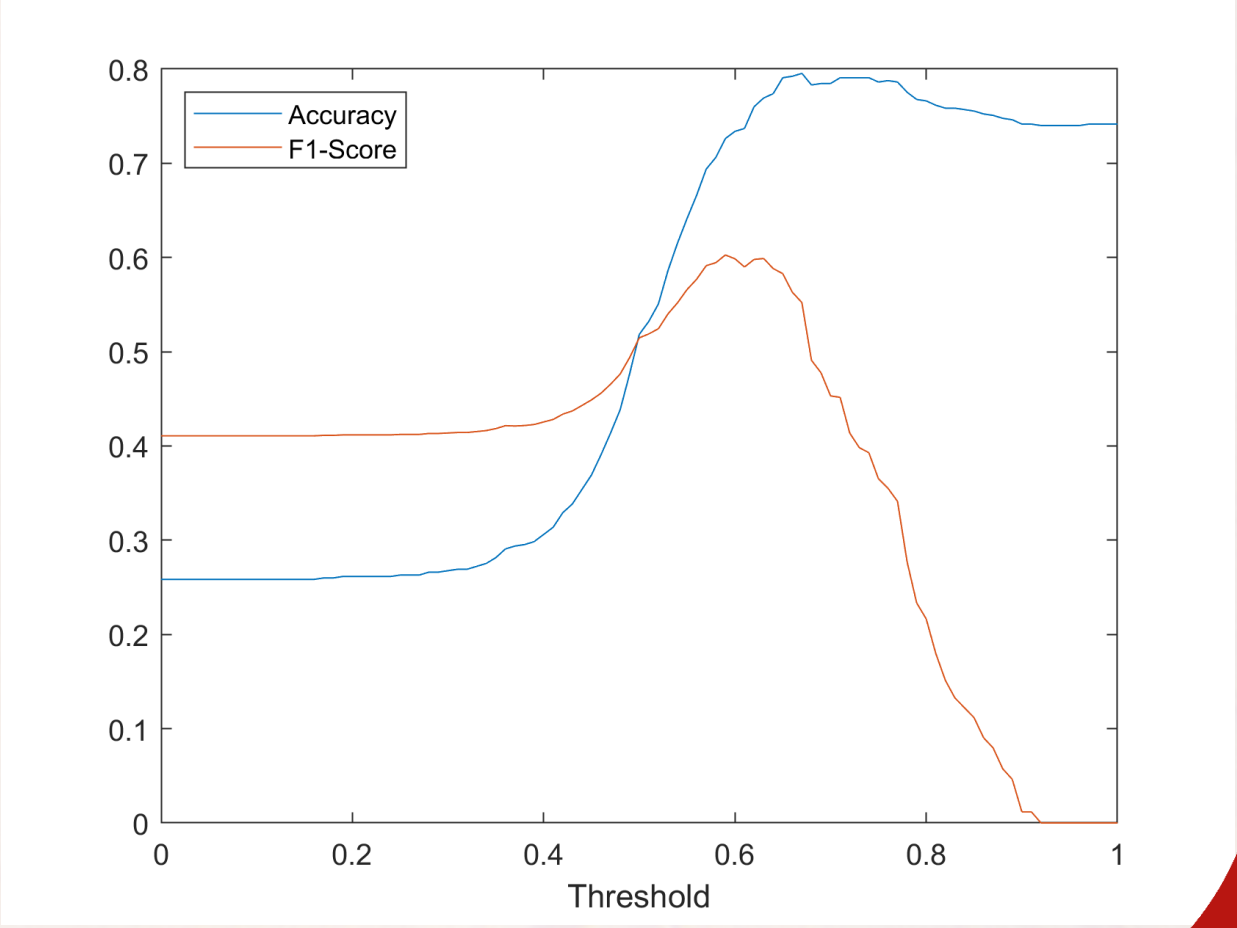
		True Class	
		Positive	Negative
Predicted Class	Positive	59	29
	Negative	109	453



Example: Clinical Test

What if we vary the threshold?





Example: Clinical Test

So how to choose threshold?

Best Accuracy	**Best F_1 Score**	**Best Youden Index**	**Best Precision**
Threshold: 0.67	Threshold: 0.59	Threshold: 0.59	Threshold: 0.83
Accuracy: 0.80	Accuracy: 0.73	Accuracy: 0.73	Accuracy: 0.76
Sensitivity: 0.49	Sensitivity: 0.80	Sensitivity: 0.80	Sensitivity: 0.07
Specificity: 0.90	Specificity: 0.70	Specificity: 0.70	Specificity: 1.00
Precision: 0.64	Precision: 0.48	Precision: 0.48	Precision: 0.92
F_1 Score: 0.55	F_1 Score: 0.60	F_1 Score: 0.60	F_1 Score: 0.13

At 0.95 sensitivity, we have: 0.41 specificity

Example: Security System

False Acceptance Rate:

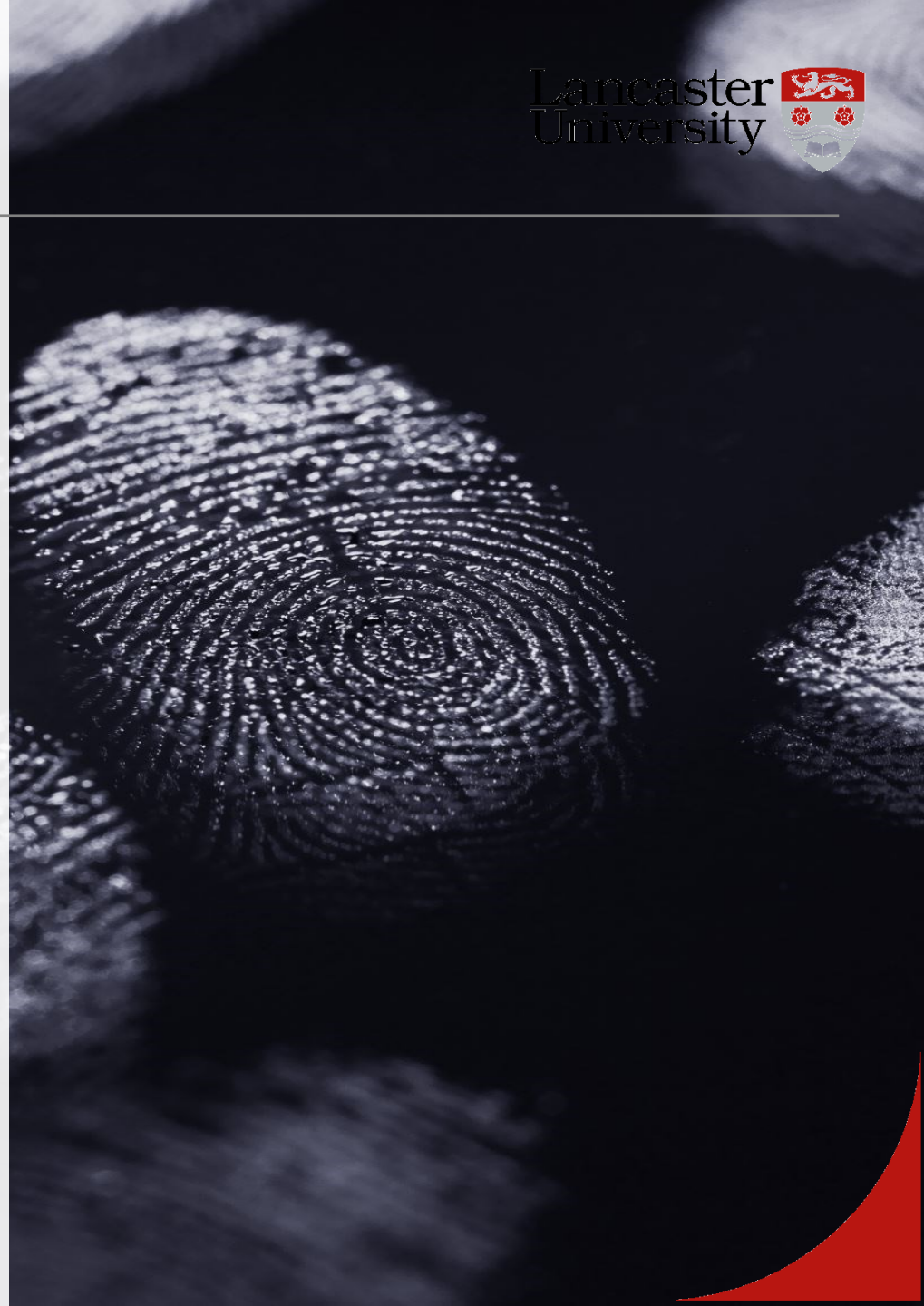
$$FAR = \frac{Incorrect\ Authentication}{Total\ Attempts}$$

False Rejection Rate:

$$FRR = \frac{Incorrect\ Rejection}{Total\ Attempts}$$

Aim: minimise both FAR and FRR.

Equal Error Rate: when FAR = FRR



Example: Security System

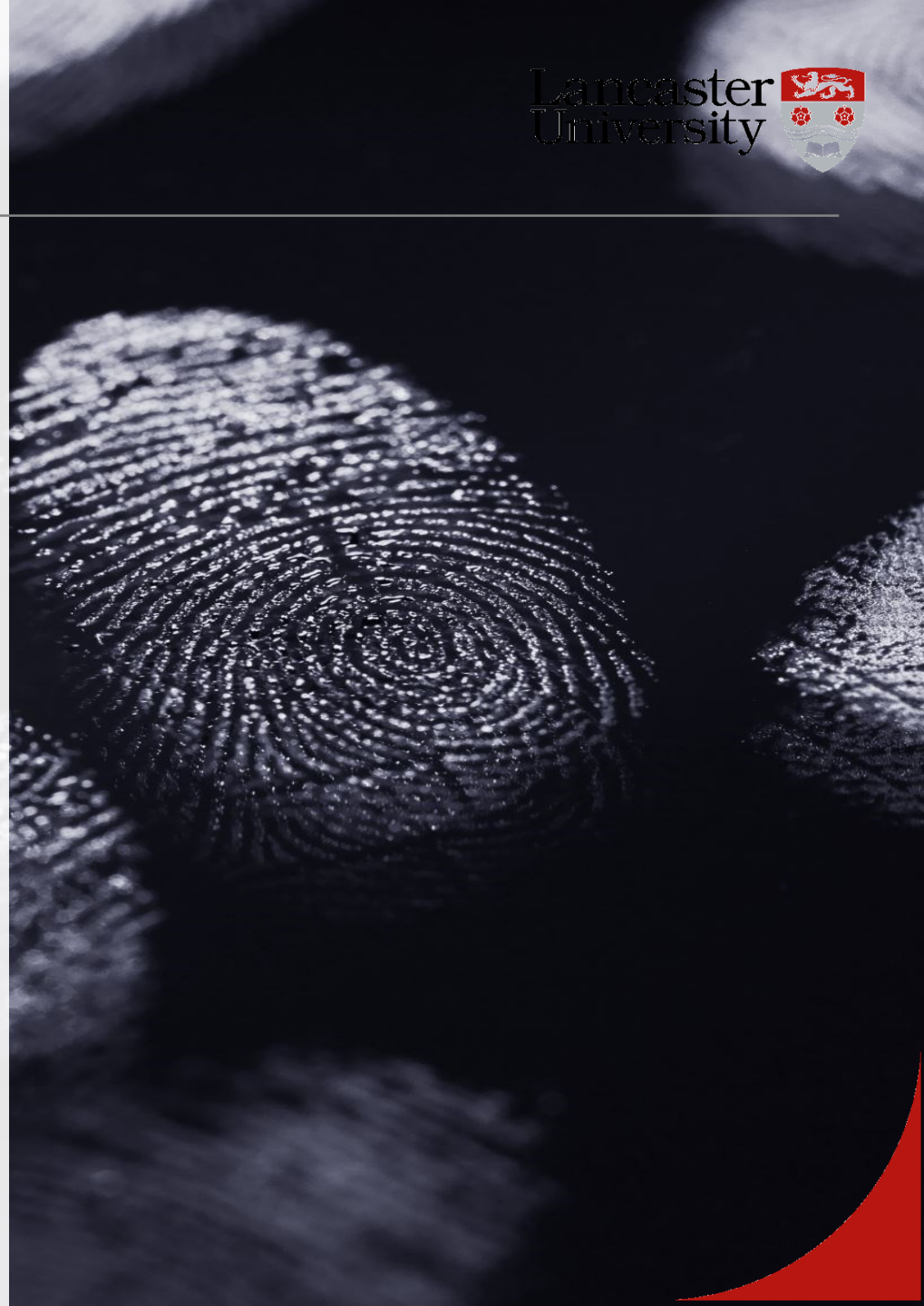
False Acceptance Rate:

If 100 people (excluding you) try to access your phone by fingerprint, what proportion will gain access?

False Rejection Rate:

Using your own phone, what proportion of times will your fingerprint be rejected?

Lower EER => Higher Accuracy

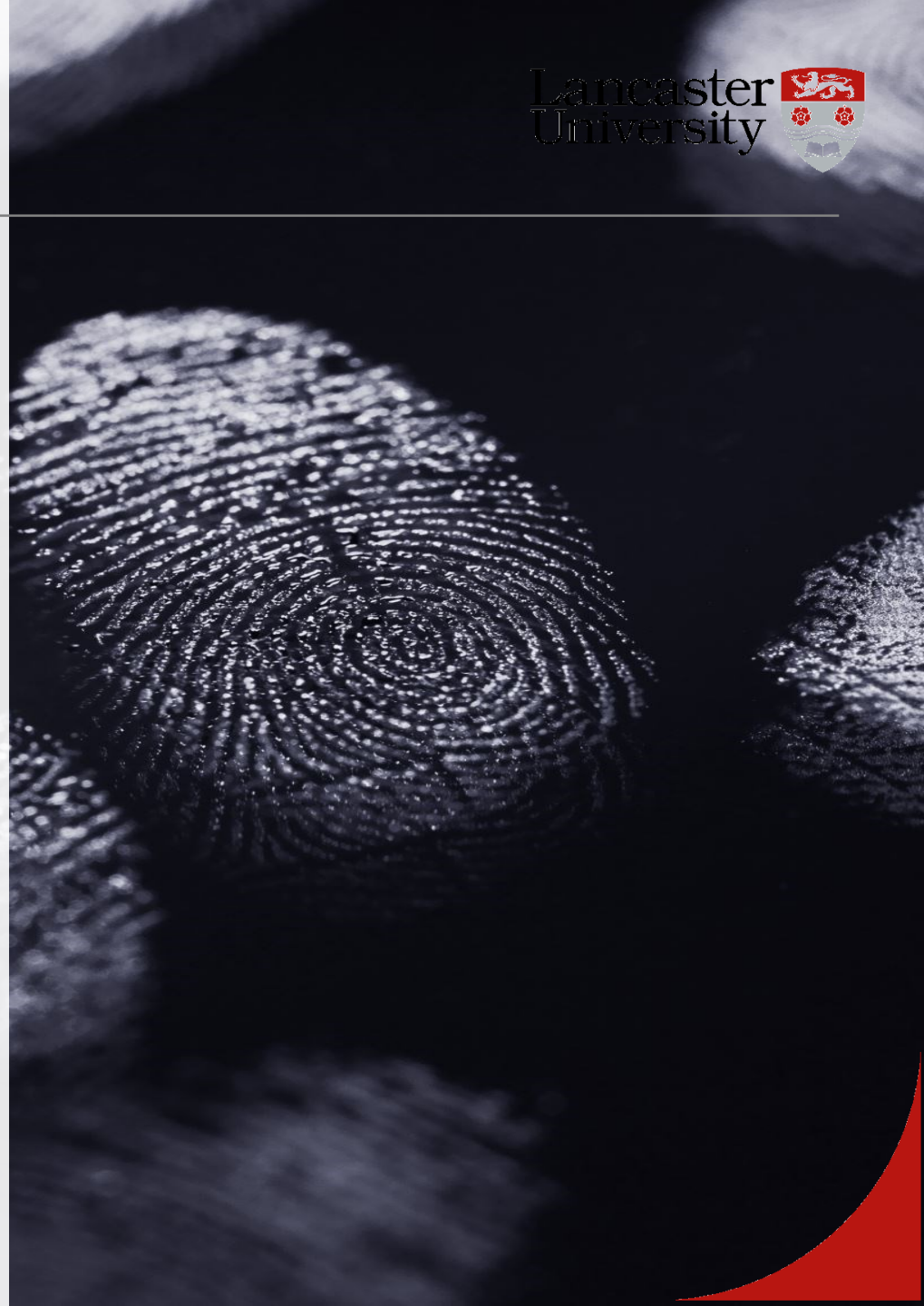


Example: Security System

Experimental Setting:

- Several “true” candidates
- Many “false” candidates
- Attempt with a large subset / whole set

Lower EER => Higher Accuracy



Summary

- Choice of metric depends on problem, data and what you want to know
- There are many more error metrics to consider, depending on subject and problem

Non-categorical labels:

$$Accuracy = \|l - \hat{l}\|_2$$

Can use other suitable distance function

Categorical labels:

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

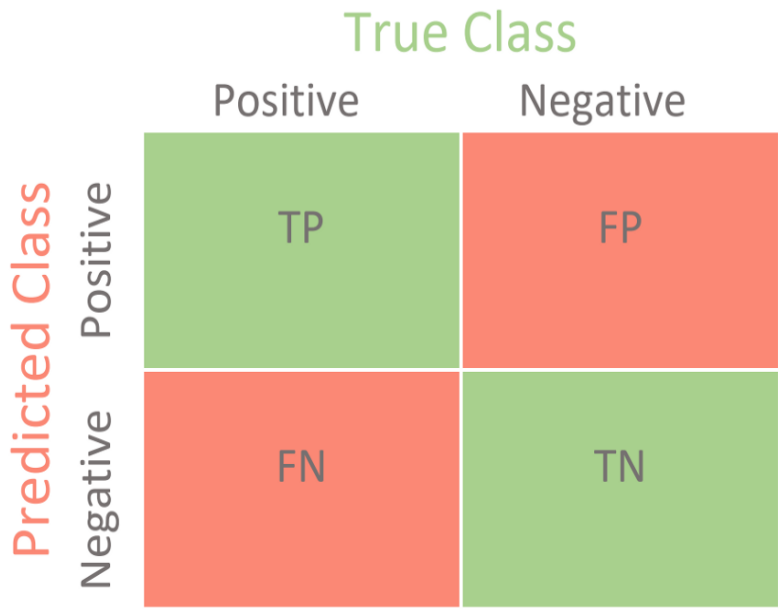
$$Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

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

$$NPV = \frac{TN}{TN + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

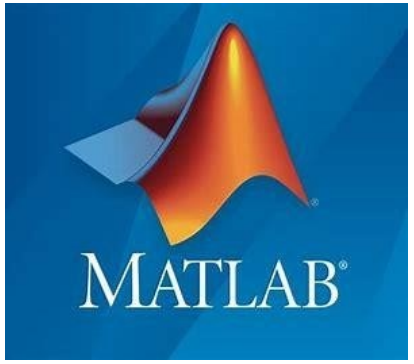


Coming up…

Lab: Clustering and Classification

- Materials on Moodle

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076



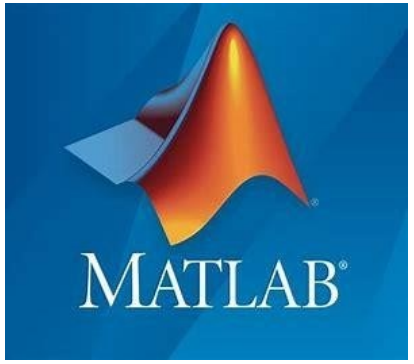
Week 5 Lectures: Classification and ANN

- More advanced approaches to classification
- Artificial neural networks

Reminder

Coursework: Task on Moodle

- 20% of Overall Mark
- Feature Extraction and Comparison of ML approaches
- Due Friday 18th November 2022 @ 4pm (Week 6)
- Submission on Moodle
- Marks in 3 weeks (Week 9)



SCC361: Artificial Intelligence

Week 5: Support Vector Machines and Artificial Neural Networks

Dr Bryan M. Williams
School of Computing and Communications, Lancaster University
Office: InfoLab21 C40 Email: b.williams6@lancaster.ac.uk

Attendance Check-in

Be sure to check in to all timetabled sessions using Attendance Check-in

To check in:

- Check the **Attendance Hub** in iLancaster
- Click **Check In**
- Wait for the “You are checked in” confirmation page
- Here is a the demo

Please DO NOT leave a timetabled session without your attendance being registered

Lecture Plan

Weeks 1-5

1. Introduction to Artificial Intelligence and Machine Learning
2. Features in Machine Learning and Feature Extraction
3. Computer Vision and Natural Language Processing
4. Clustering and Classification
5. **Support Vector Machines and Artificial Neural Networks**

Weeks 6-10:

6. Genetic Algorithms
7. Naïve Bayesian Classifier
8. Decision Tree Classifier
9. Introduction to Deep Neural Networks
10. Introduction to Convolutional Neural Networks

Last Week

Lancaster University

Week 1: Intro to AI and ML

- Definitions
- Goals of AI
- AI in Society
- Machine Learning
- Supervised vs Unsupervised

Week 2: Features and Extraction

- Features and Feature Vectors
- Feature Labels
- Filters
- 1-Hot Encoding
- Bag of Words

Week 3: CV and NLP

- Convolution
- Parts of Speech Tagging
- GL: Forensics
- GL: Environmental Science

Week 4: Clustering and Classification

- **Hierarchical Clustering**
- **K-Means Clustering**
- **Classification Algorithms**
- **Nearest Neighbour**

Week 5: SVM and Artificial Neural Networks

- Measures of Success and Error Measures
- Support Vector Machines
- Training and Solution Techniques
- ANNs and Components

Machine Learning Paradigm

Training Data
The observed set of example data

Test Data
Previously unseen data generated by the same process as the training data

Training process

Inferring or learning something about the process that generated the training data

Prediction

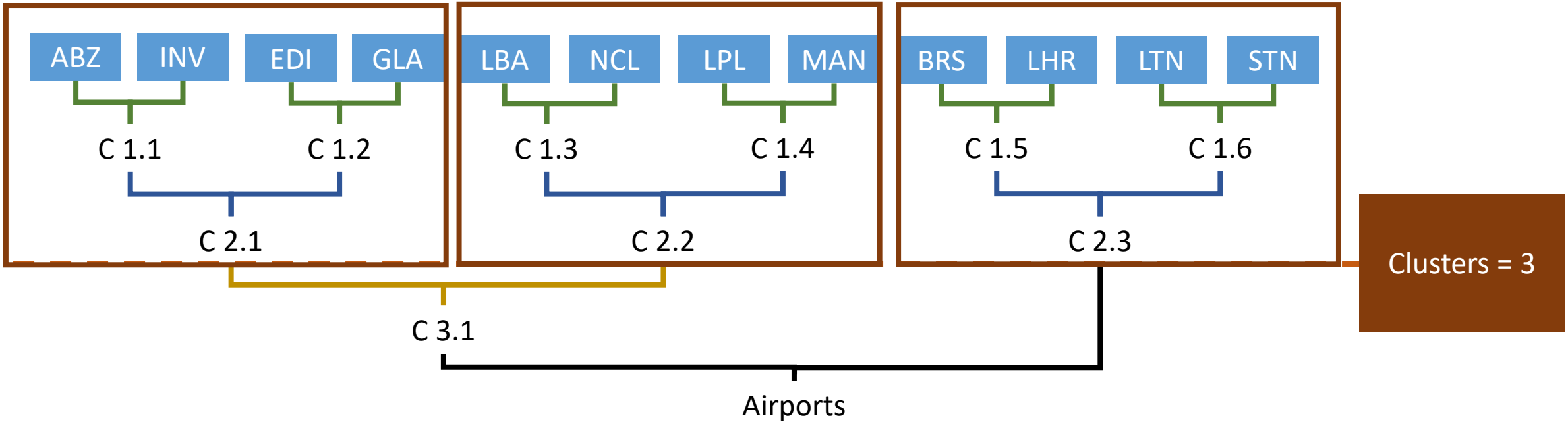
Using the inference to make predictions about the test data

Evaluation

Comparing the prediction outputs with the correct test data outputs

Dendrogram

We can obtain different clusterings of the data by splitting on different levels



K-Means Clustering

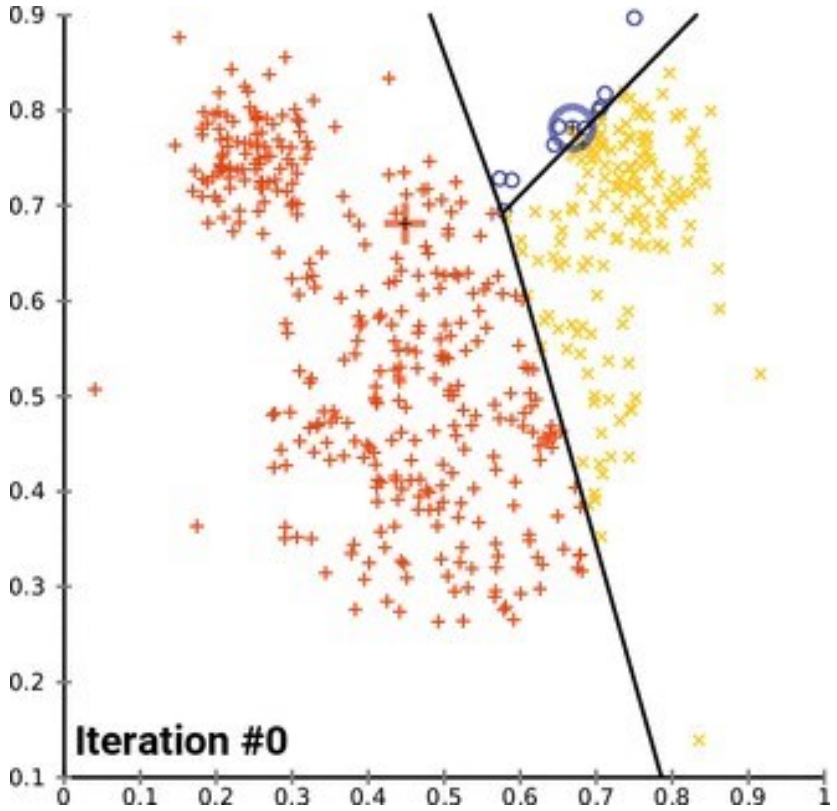
K-Means clustering is an **optimisation problem**

Aim:
Minimize the within-cluster sum of squares (WCSS) i.e. variance

Given n observations $(x_1, x_2, \dots, x_n)$, where each is a d-dimensional real vector, K-Means partitions the n observations into k sets S where $k \leq n$:

$$S = \{S_1, S_2, \dots, S_k\}$$

Such that the variance of each subset S_i is minimised.



Source: K-Means Clustering on Wikipedia

Classification: K-Nearest Neighbour

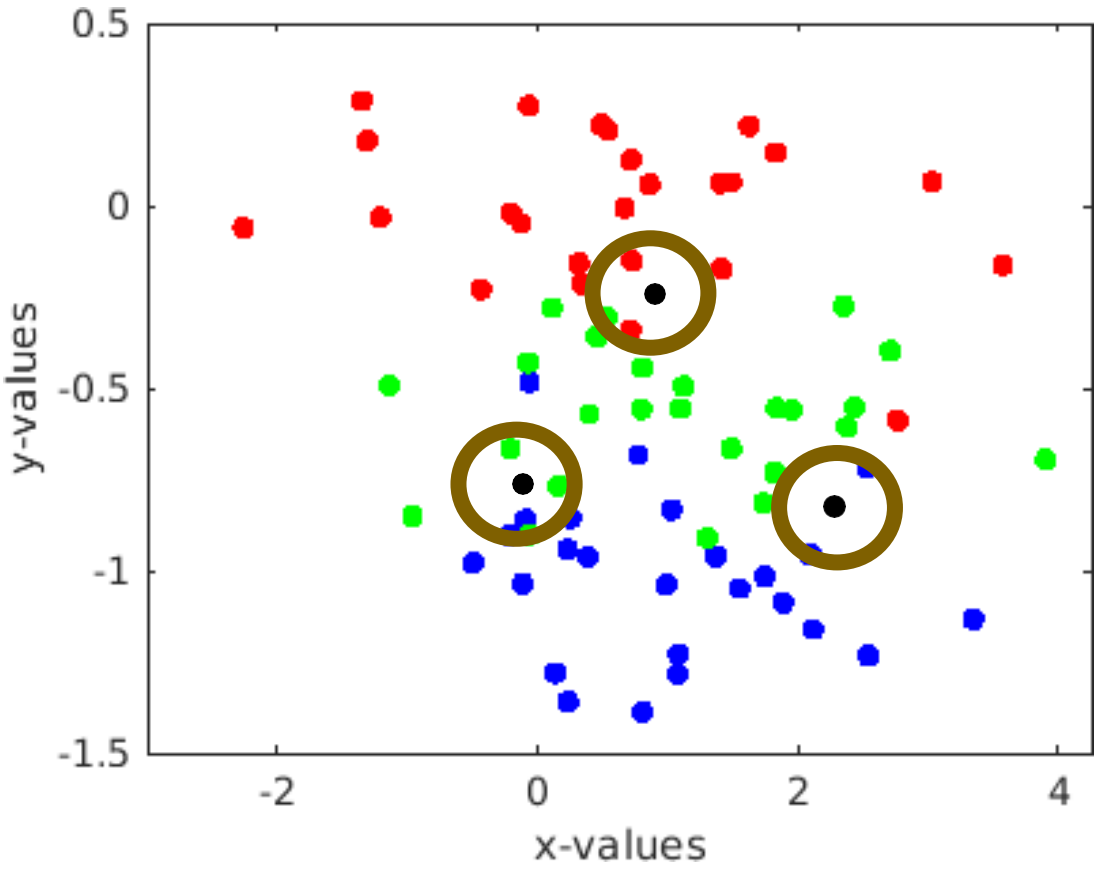
Given **training data** and **a test point**

Look at the k most similar examples

Assign the majority class label k = number of nearest neighbours to search for

Special case: $k = 1$

- 1 nearest neighbour



Tutorial Questions

1. How does training and evaluation work in MATLAB?

2. What role does correlation play?

3. Why do we do cross-validation?

```
knnmodel = fitcknn(trfeatures, trlabels);

predicted = predict(knnmodel, tefeatures)

accuracy = compare(predicted, telabels);
```

This Week

Lancaster University

Week 1: Intro to AI and ML

- Definitions
- Goals of AI
- AI in Society
- Machine Learning
- Supervised vs Unsupervised

Week 2: Features and Extraction

- Features and Feature Vectors
- Feature Labels
- Filters
- 1-Hot Encoding
- Bag of Words

Week 3: CV and NLP

- Convolution
- Parts of Speech Tagging
- GL: Forensics
- GL: Environmental Science

Week 4: Clustering and Classification

- Hierarchical Clustering
- K-Means Clustering
- Classification Algorithms
- Nearest Neighbour

Week 5: SVM and Artificial Neural Networks

- **Measures of Success and Error Measures**
- **Support Vector Machines**
- **Training and Solution Techniques**
- **ANNs and Components**

Expected Learning Outcomes

After this week, we aim to be able to

- be aware of and understand different error measures
- understand and implement support vector machines
- explain the concept of Artificial Neural Networks
- understand common ANN terms such as
  - activation functions
  - gradient descent and backpropagation
  - overfitting and dropout
  - multilayer neural networks

Measures of Success and Error Measures

Machine Learning Paradigm

Training Data
The observed set of example data

Test Data
Previously unseen data generated by the same process as the training data

Training process

Inferring or learning something about the process that generated the training data

Prediction

Using the inference to make predictions about the test data

Evaluation

Comparing the prediction outputs with the correct test data outputs

Evaluation

Suppose we have a method f and a test set $S_x = \{X_1, \dots, X_n\}$ where $X_i = (\boxed{x_i}, \boxed{l_i})$

data

label

Our method estimates the label as

$$\widehat{l_i} = f(x_i)$$

If the model is suitable, we should have

$$\widehat{l_i} = l_i \text{ for all } i = 1, \dots, n$$

$$Accuracy = \frac{Number\ of\ test\ elements\ where\ the\ estimated\ label\ is\ correct}{Number\ of\ test\ elements}$$

Accuracy

Example

x_1: $l_1 = \text{"dog"}$ $\hat{l_1} = f(x_1) = \text{"dog"}$ ✓

x_2: $l_2 = \text{"dog"}$ $\hat{l_2} = f(x_2) = \text{"dog"}$ ✓

x_3: $l_3 = \text{"cat"}$ $\hat{l_3} = f(x_3) = \text{"cat"}$ ✓

x_4: $l_4 = \text{"cat"}$ $\hat{l_4} = f(x_4) = \text{"dog"}$ ✗

x_5: $l_5 = \text{"cat"}$ $\hat{l_5} = f(x_5) = \text{"cat"}$ ✓

$$Accuracy = \frac{\#\{\text{Correct Predictions}\}}{|S_x|}$$

$$\#\{\text{Correct Predictions}\} = 4$$

$$|S_x| = 5$$

$$Accuracy = \frac{4}{5} = 0.8 = 80\%$$

Accuracy

Non-categorical

What if model output is not categorical?

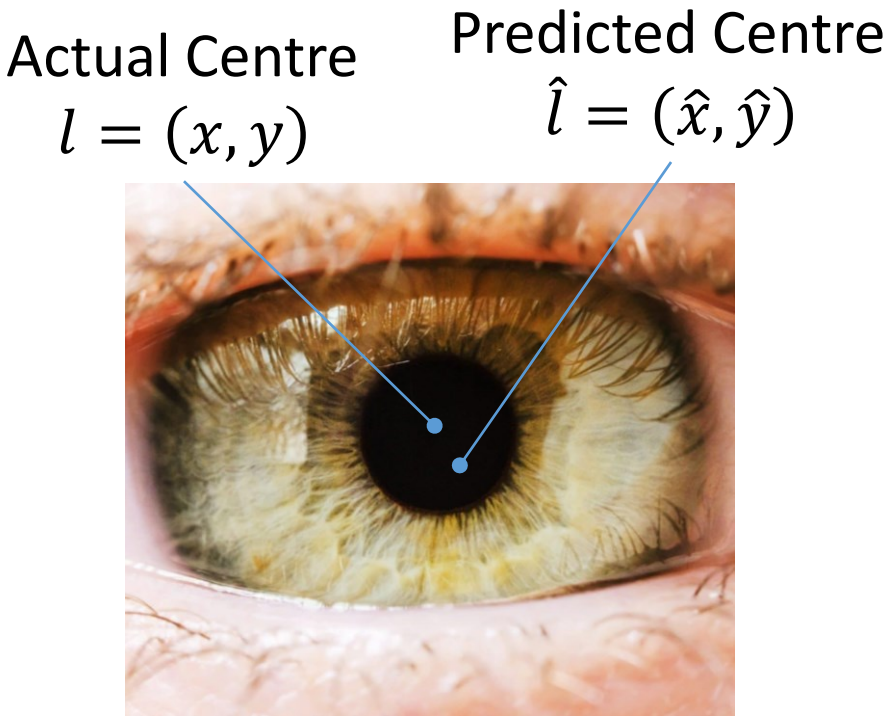
Example: determine location of centre of pupil

More meaningful: **distance** such as L^2-norm:

$$Accuracy = \|l - \hat{l}\|_2 = \sqrt{(x - \hat{x})^2 + (y - \hat{y})}$$

E.g. $l = (95,101)$, $\hat{l} = (100,103)$

$$Accuracy = \|l - \hat{l}\|_2 = \sqrt{(95 - 100)^2 + (101 - 103)}$$
$$= \sqrt{(-5)^2 + (-2)^2} = \sqrt{25 + 4} = \sqrt{29} \approx 5.385$$



Confusion Matrices

- Confusion Matrices give us more insight.
  - **TP**: We predicted **positive** and we were **right**
  - **TN**: We predicted **negative** and we were **right**
  - **FP**: We predicted **positive** and we were **wrong**
  - **FN**: We predicted **negative** and we were **wrong**

Predicted Class \ True Class	Positive	Negative
Positive	TP	FP
Negative	FN	TN

Examples:

- Positive = Glaucoma, Negative = Healthy
- Positive = Cats, Negative = Dogs

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

Confusion Matrices

- Accuracy is important as an **overall** view.

Example:

- Suppose we are developing a new automated technique for Glaucoma diagnosis
- Our technique is 95% accurate on a test set of 10,000 people
- It is wrong for 5%
  - But which ones???

Healthy eyes

Periphial vision loss due to glaucoma

Confusion Matrices

Glaucoma (GL) Diagnosis Example:

- Our technique is 95% accurate on a test set of 10,000 people
- Our test set includes
  - 500 GL Patients
  - 9500 Non-GL Patients
- Scenario 1: Our results:
  - All GL Patients diagnosed correctly
  - 9000 Healthy patients diagnosed correctly
  - 500 Healthy patients mis-diagnosed

		True Class	
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.95$$

Confusion Matrices

Glaucoma (GL) Diagnosis Example:

- Our technique is 95% accurate on a test set of 10,000 people
- Our test set includes
  - 500 GL Patients
  - 9500 Non-GL Patients
- Scenario 2: Our results:
  - All Healthy Patients diagnosed correctly
  - All DL patients mis-diagnosed

		True Class	
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} = 0.95$$

Confusion Matrices

What other measures can we consider?

- **Sensitivity**, recall, hit rate, true positive rate
- **Specificity**, selectivity, true negative rate
- **Precision**, positive predicted value
- **Negative predictive value**
- Miss rate, false negative rate
- Fall out, false positive rate
- False discovery rate
- False omission rate
- **F1 score**, dice coefficient

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Sensitivity and Specificity

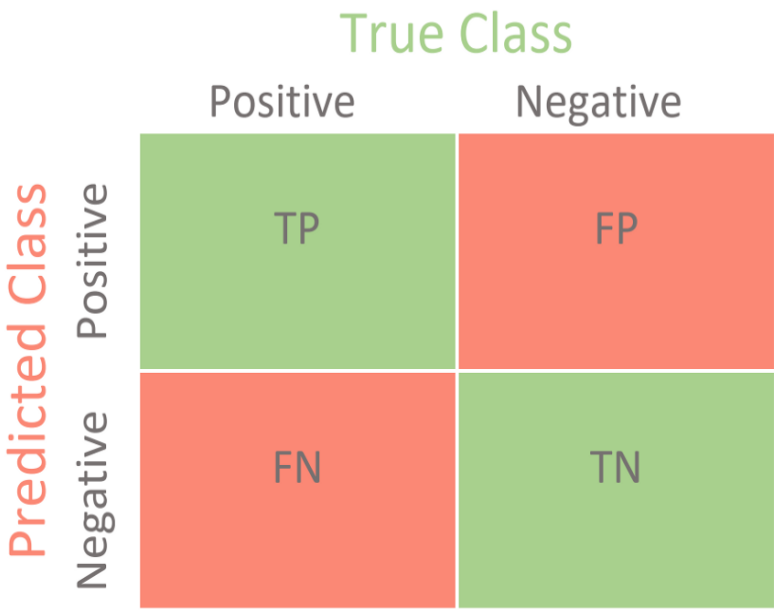
Sensitivity: Probability of *positive* outcome if truly positive

$$Sensitivity = \frac{TP}{TP + FN} = \frac{TP}{|True\ Positives|}$$

Specificity: Probability of *negative* outcome if truly *negative*

$$Specificity = \frac{TN}{TN + FP} = \frac{TN}{|True\ Negatives|}$$

- Usually consider sensitivity and specificity together.
- Aim for high sensitivity and high specificity.
- Can use trade-off thresholds.



Precision and Negative Predicted Value (NPV)

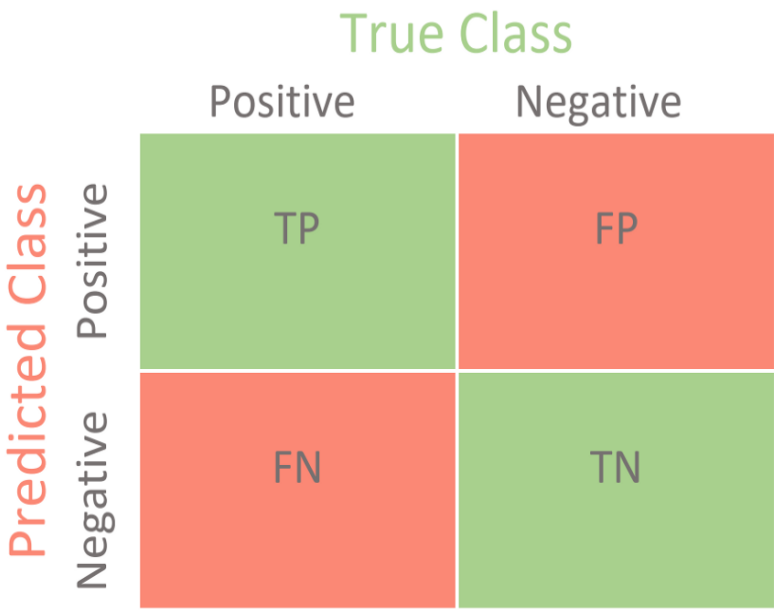
Precision: Proportion of *truly positive* outcomes to *positive predictions*

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{|Predicted\ Positives|}$$

NPV: Proportion of *truly negative* outcomes to *predicted negative*

$$NPV = \frac{TN}{TN + FN} = \frac{TN}{|Predicted\ Negatives|}$$

- Should consider precision and NPV together.
- Aim for high precision and NPV.
- Often consider **precision and recall** (sensitivity) together



F_1 Score

It can be difficult to compare a pair of values

Aim to reduce to a single value

We use harmonic mean when interested in average *rate*

F_1 score is the harmonic mean of precision and recall

$$F_1 = \frac{2}{recall^{-1} + precision^{-1}} = 2\frac{precision \cdot recall}{precision + recall}$$

$$= \frac{2TP}{2TP + FP + FN}$$

For our examples:

Example 1: $F_1 = \frac{2 \cdot 500}{2 \cdot 500 + 500 + 0} = \frac{1000}{1500} \approx 0.66$

Example 2: $F_1 = \frac{2 \cdot 0}{2 \cdot 0 + 0 + 500} = 0$

Confusion Matrices

Back to our example

Measure	Scenario 1	Scenario 2
TP	500	0
TN	9000	9500
FP	500	0
FN	0	500
Accuracy	0.95	0.95
Sensitivity	1	0
Specificity	0.947	1
Precision	0.5	Undefined
NPV	1	0.95
F1 Score	0.667	0

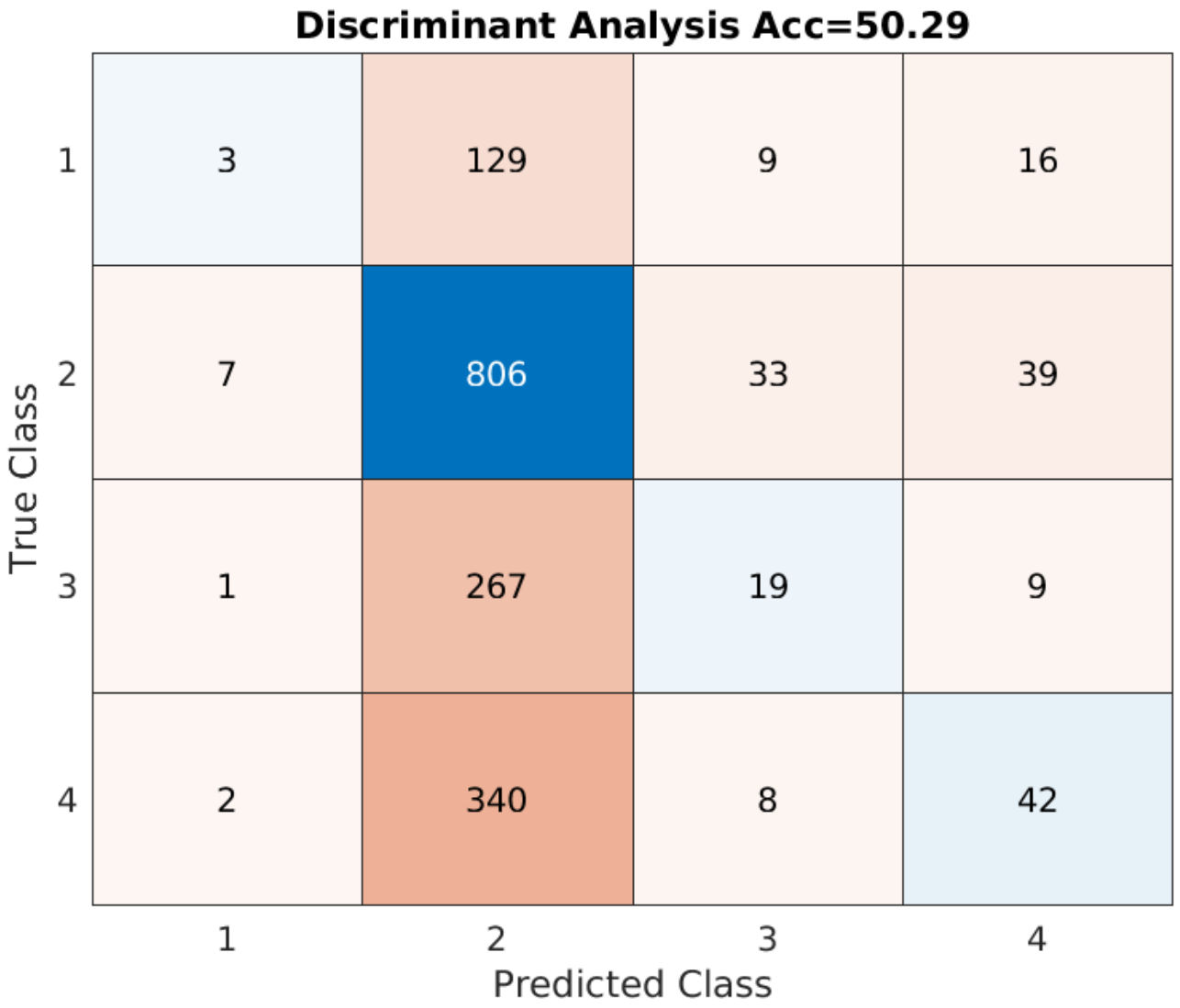
Scenario 1	**True Class**		
Predicted Class		GL	Healthy
	GL	500	500
	Healthy	0	9000

Scenario 2	**True Class**		
Predicted Class		GL	Healthy
	GL	0	0
	Healthy	500	9500

Non-Binary Confusion Matrices

Compute accuracy in the same way.

There exist generalised versions of F1 Score etc but this is not covered here.



Crum WR, Camara O, Hill DL. Generalized overlap measures for evaluation and validation in medical image analysis. IEEE transactions on medical imaging. 2006 Oct 30;25(11):1451-61.

Example: Security System

False Acceptance Rate:

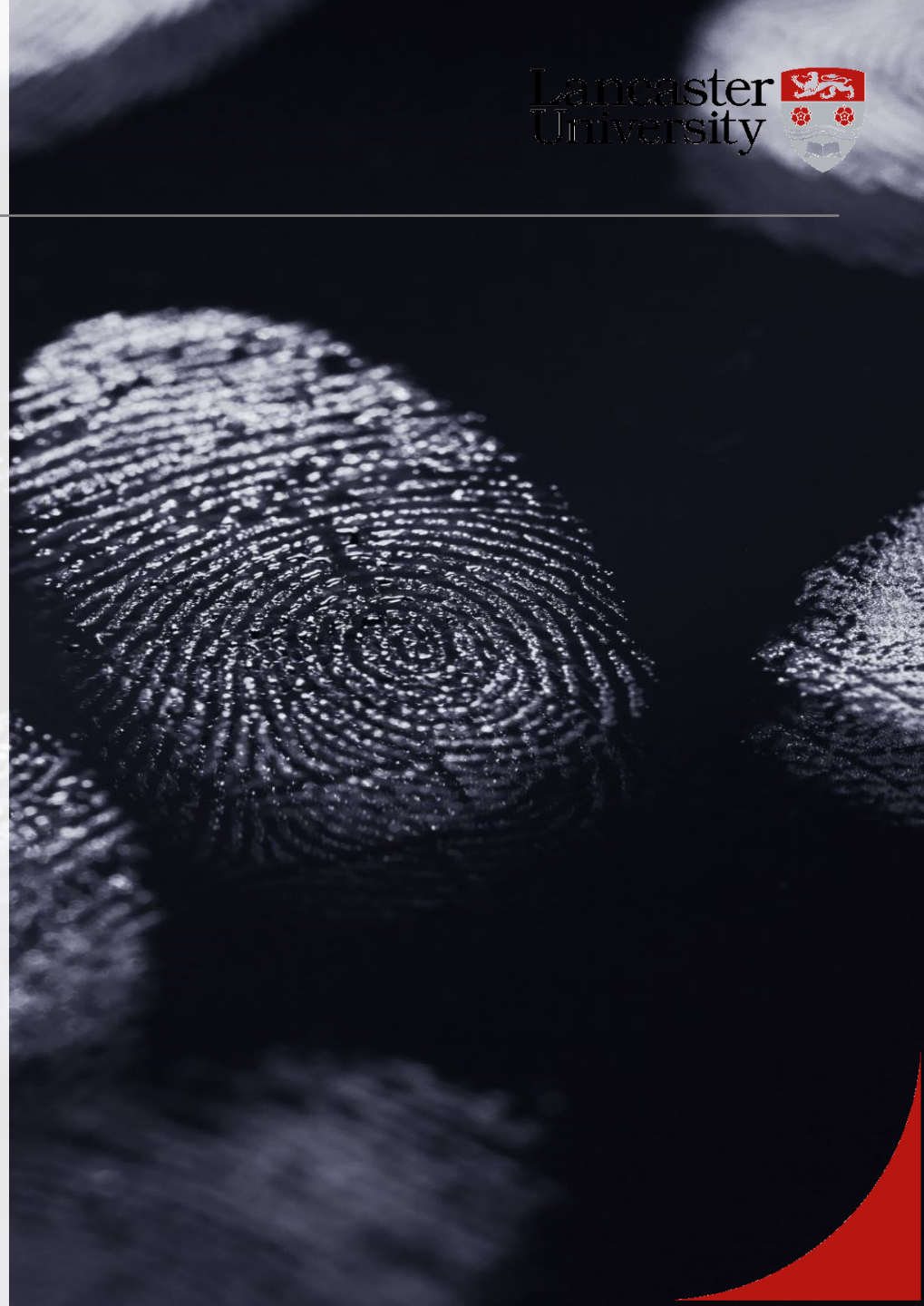
$$FAR = \frac{Incorrect\ Authentication}{Total\ Attempts}$$

False Rejection Rate:

$$FRR = \frac{Incorrect\ Rejection}{Total\ Attempts}$$

Aim: minimise both FAR and FRR.

Equal Error Rate: when FAR = FRR



Summary

- Choice of metric depends on problem, data and what you want to know
- There are many more error metrics to consider, depending on subject and problem

Non-categorical labels:

$$Accuracy = \|l - \hat{l}\|_2$$

Can use other suitable distance function

Categorical labels:

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

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

$$Specificity = \frac{TN}{TN + FP}$$

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

$$NPV = \frac{TN}{TN + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Support Vector Machines
SVM

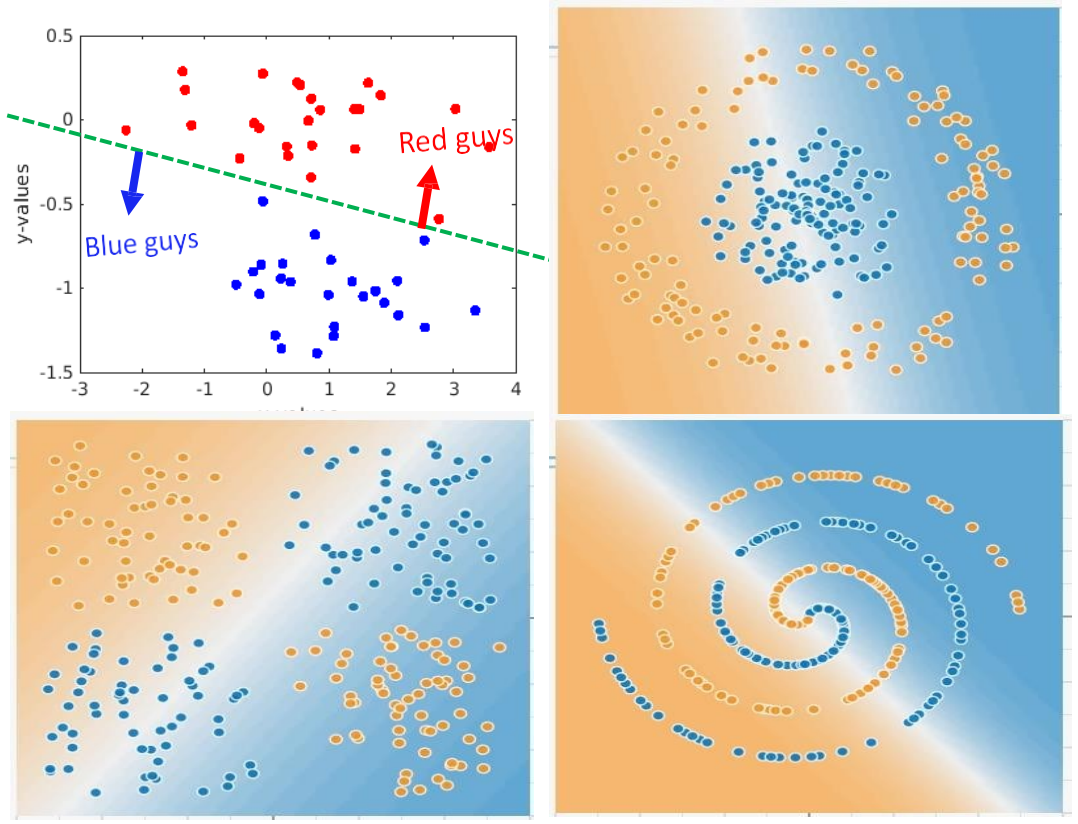
ML: Remaining Problems

Typical datasets are large in size

Data points are not usually well separated by lines

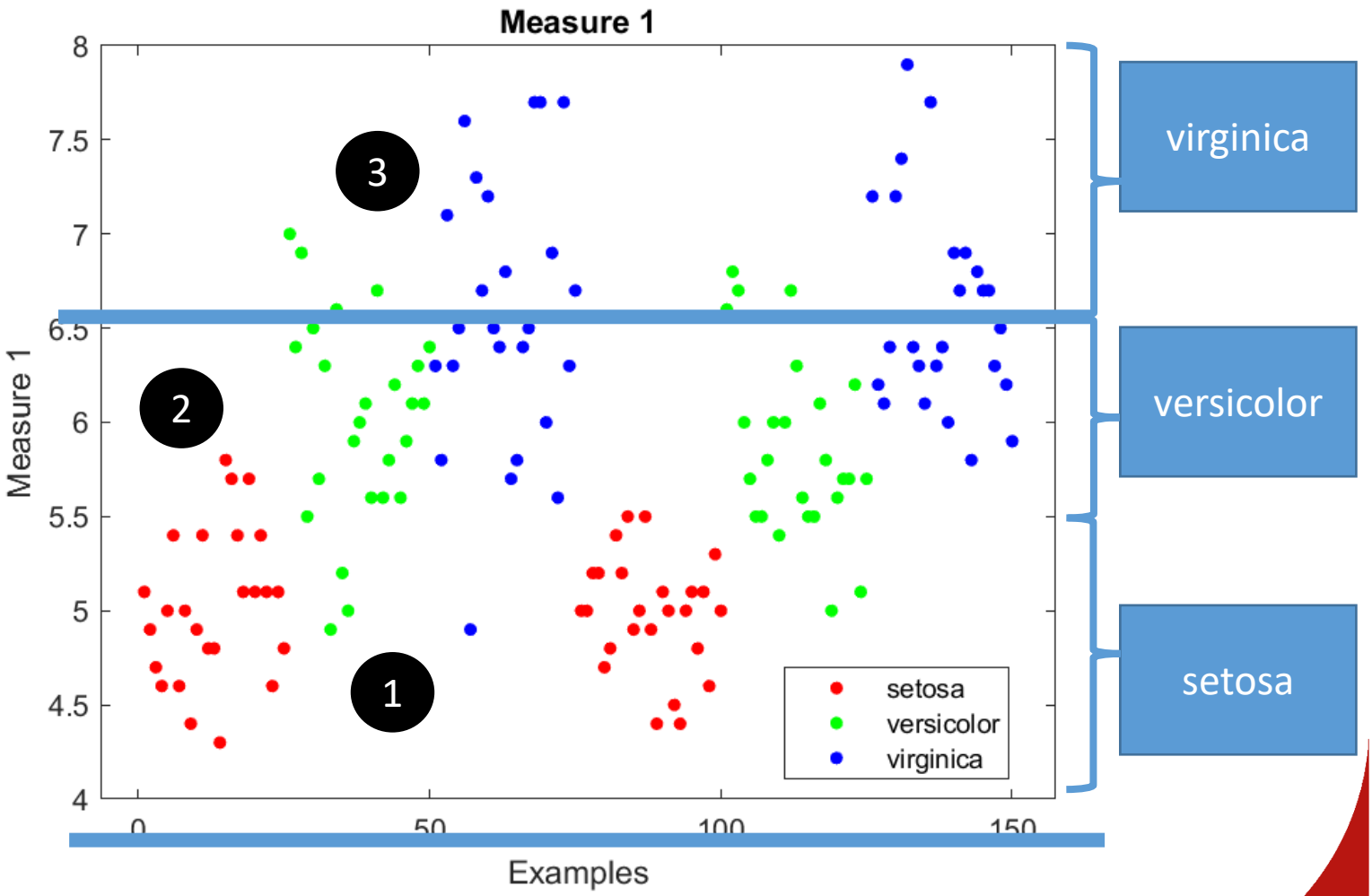
Real world datasets are complex

Classic ML Algorithms struggle to learn from this data

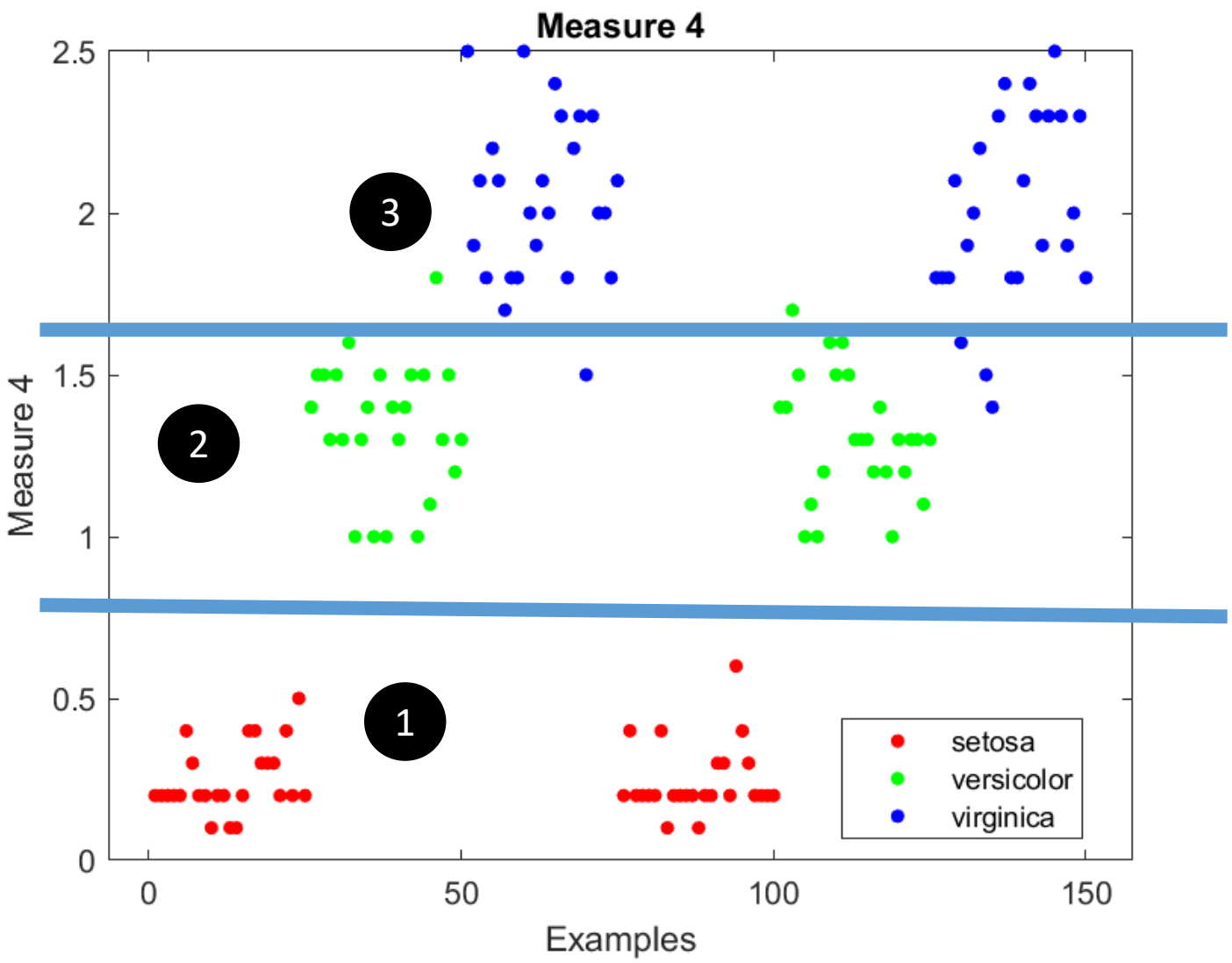


Decision Making

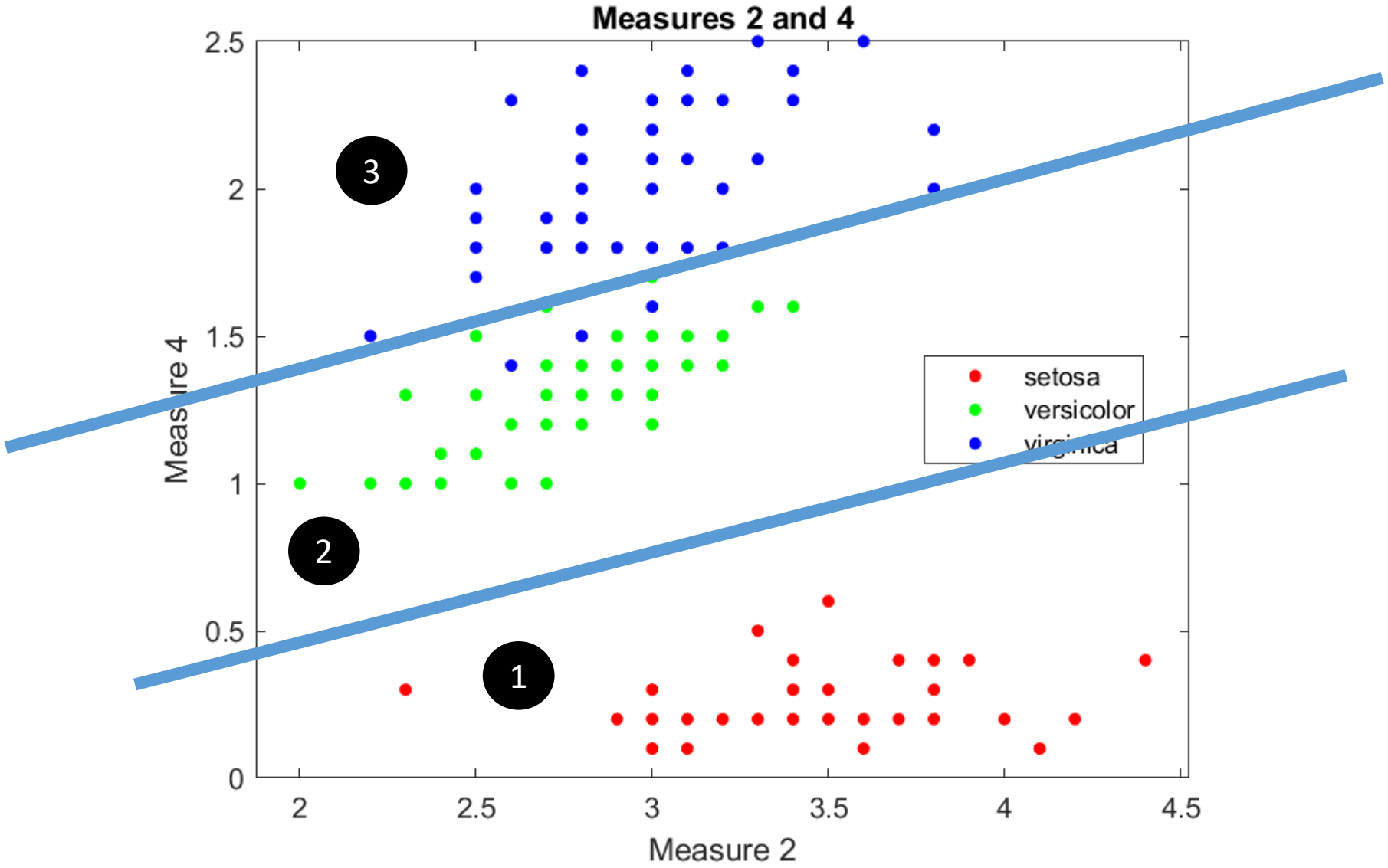
Measure1	Measure2	Measure3	Measure4	Class1	Class2
5.1	3.5	1.4	0.2	1	setosa
4.9	3	1.4	0.2	1	setosa
4.7	3.2	1.3	0.2	1	setosa
4.6	3.1	1.5	0.2	1	setosa
5	3.6	1.4	0.2	1	setosa
5.4	3.9	1.7	0.4	1	setosa
4.6	3.4	1.4	0.3	1	setosa
5	3.4	1.5	0.2	1	setosa
4.4	2.9	1.4	0.2	1	setosa



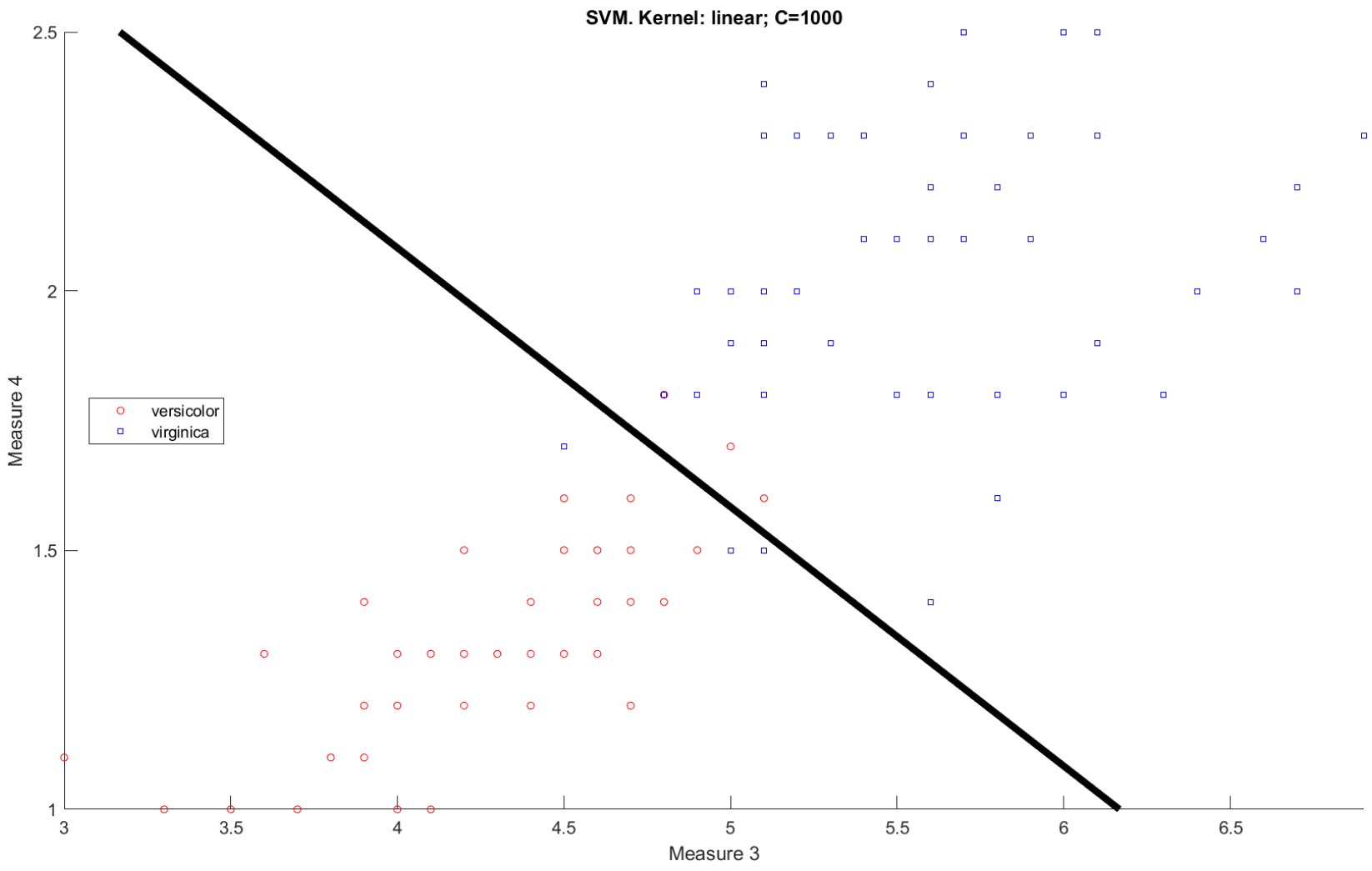
Decision Making



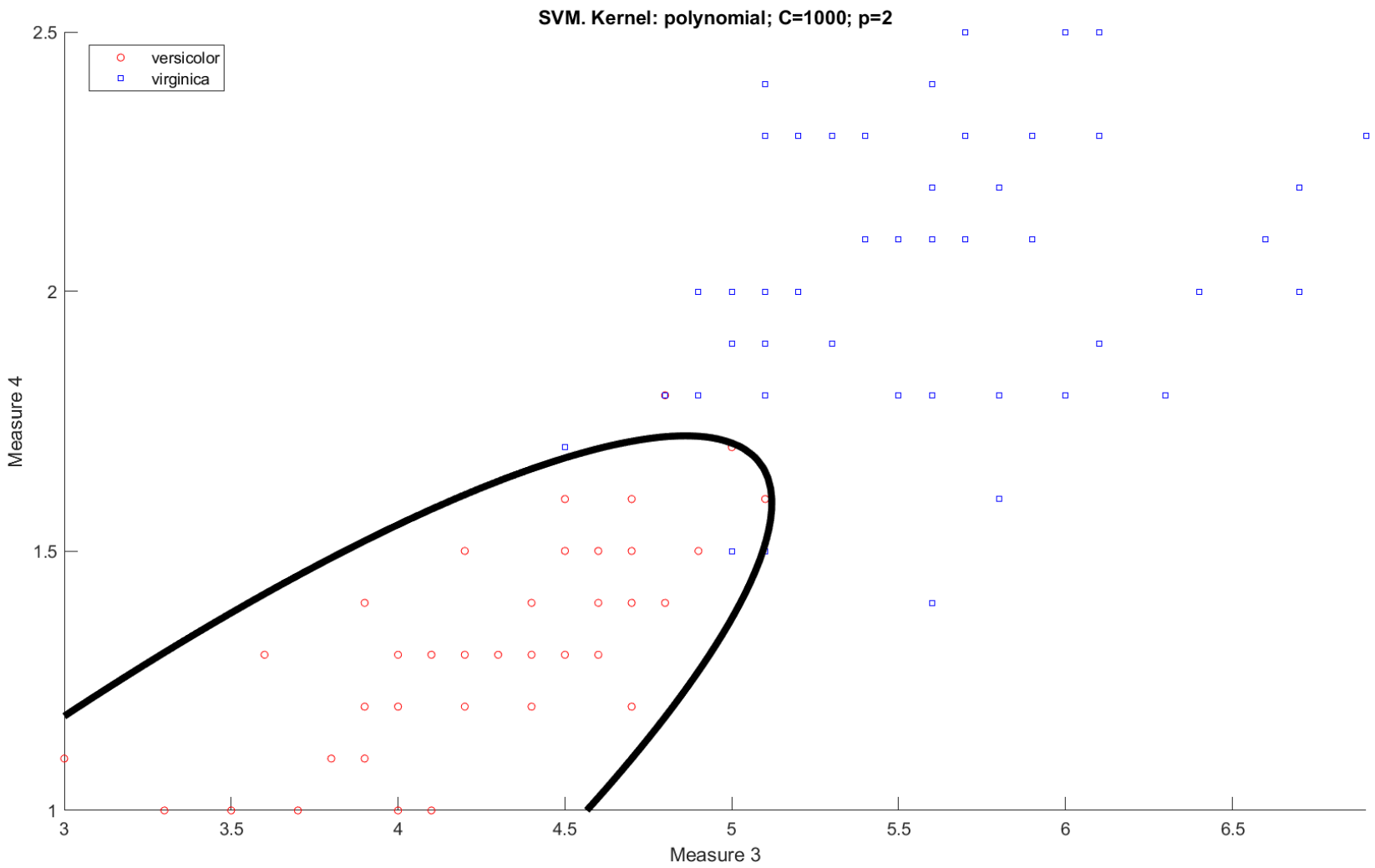
Decision Making



Support Vector Machines



Support Vector Machines



Working through SVM

We are given a training dataset S of n items

$$S = \{X_1, X_2, \dots, X_n\}$$

Each item belongs to

- *either* class 1 denoted by S_1
- *or* class 2 denoted by S_2

I.e. either $X_i \in S_1$ or $X_i \in S_2$

- No item can belong to both classes simultaneously
- Every item belongs to a class

I.e. for $i = 1, \dots, n$, either $X_i \in S_1$ or $X_i \in S_2$

Building SVM

Example Dataset – Computer Vision

S: A set of images; $n = 12$

$$S = \{X_1, \dots, X_{12}\} = \{X_1, X_2, X_3, X_4, X_5, X_6, X_7, X_8, X_9, X_{10}, X_{11}, X_{12}\}$$

Building SVM

Example Dataset – Computer Vision

S_1: Images with Dogs

S_2: Images with Cats

$$S_1 = \{X_1, X_3, X_5, X_8, X_9, X_{11}\}$$

$$S_2 = \{X_2, X_4, X_6, X_7, X_{10}, X_{12}\}$$

Building SVM

Example Dataset – Computer Vision

S_1: Images with People

S_2: Images without People

$$S_1 = \{X_4, X_5, X_6, X_8, X_{10}, X_{11}\}$$

$$S_2 = \{X_1, X_2, X_3, X_7, X_9, X_{12}\}$$

Building SVM

Example Dataset – Natural Language Processing

S: A set of words; $n = 10$

Announce (X_1), **Arrive** (X_2), **Computer** (X_3), **Facilitate** (X_4), **Newspaper** (X_5), **Render** (X_6), **Replacement** (X_7), **Thank** (X_8), **Two** (X_9), **Understanding** (X_{10})

S_1: Nouns in S

Computer (X_3), **Newspaper** (X_5), **Replacement** (X_7), **Two** (X_9), **Understanding** (X_{10})

S_2: Verbs in S

Announce (X_1), **Arrive** (X_2), **Facilitate** (X_4), **Render** (X_6), **Thank** (X_8)

S_1: Starting with a vowel

Announce (X_1), **Arrive** (X_2), **Understanding** (X_{10})

S_2: Starting with a consonant

Computer (X_3), **Facilitate** (X_4), **Newspaper** (X_5), **Render** (X_6), **Replacement** (X_7), **Thank** (X_8), **Two** (X_9)

Building SVM

Problem statement

Each item X_i in the set S is a pair, including data and a label:

$$X_1 = (x_1, y_1), X_2 = (x_2, y_2), \dots, X_n = (x_n, y_n)$$

where

- Each x_i is a p-dimensional real vector
- Each y_i is a corresponding label for x_i. For this, we will use

$$y_i = \begin{cases} 1 & if\ X_i \in S_1 \\ -1 & if\ X_i \in S_2 \end{cases}$$

Building SVM

Problem statement

AIM:

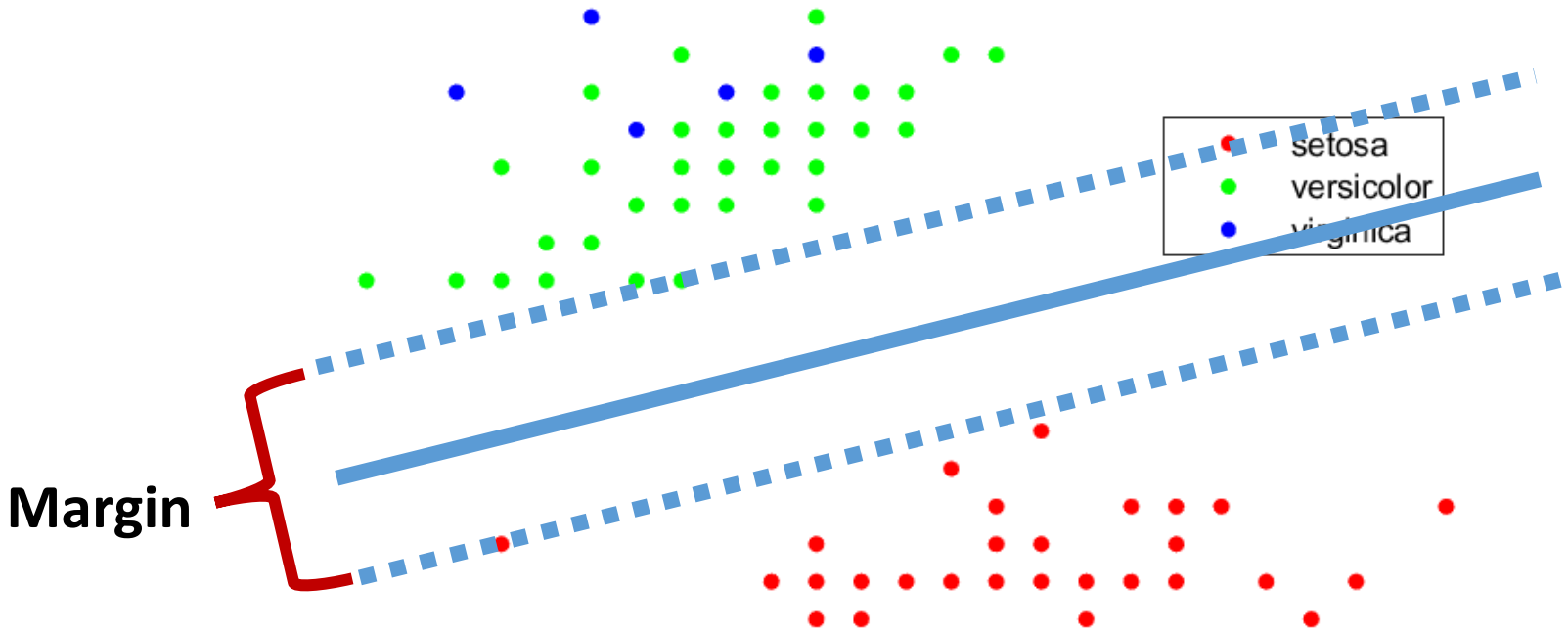
To automatically determine the correct label y_i given the data x_i.

? ? ?

Building SVM

SVM Approach

SVM Approach:
Find the "maximum-margin hyperplane" dividing the points $x_i \in S_1$ from the points $x_i \in S_2$ so that the distance between the hyperplane and the nearest point x_i from either group is maximised.



Building SVM

Hyperplanes

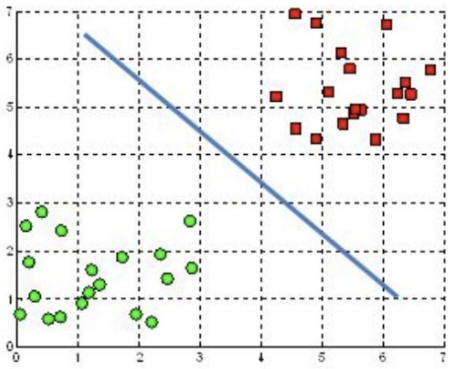
General equation for a hyperplane:

$$w^\top \boldsymbol{x} - b = 0, \qquad where\ \boldsymbol{x} = (x_1, x_2, \dots, x_n)$$

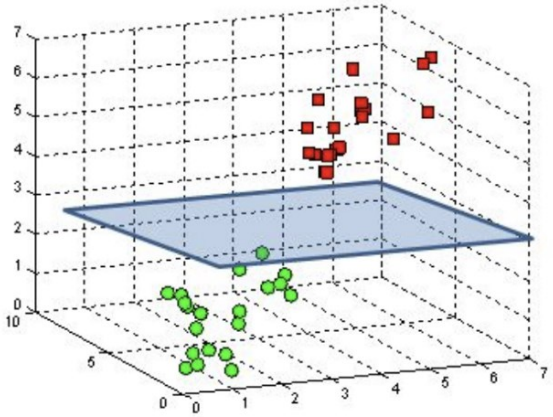
where

- w is the normal vector to the hyperplane
- $\frac{b}{\|w\|}$ is the distance from the origin to the hyperplane along the normal

A hyperplane in $\mathbb{R}^2$ is a line



A hyperplane in $\mathbb{R}^3$ is a plane



Building SVM

Margins

If the training data is linearly separable, e.g. can be separated with a straight line, we can define two parallel hyperplanes h_1, h_2 such that

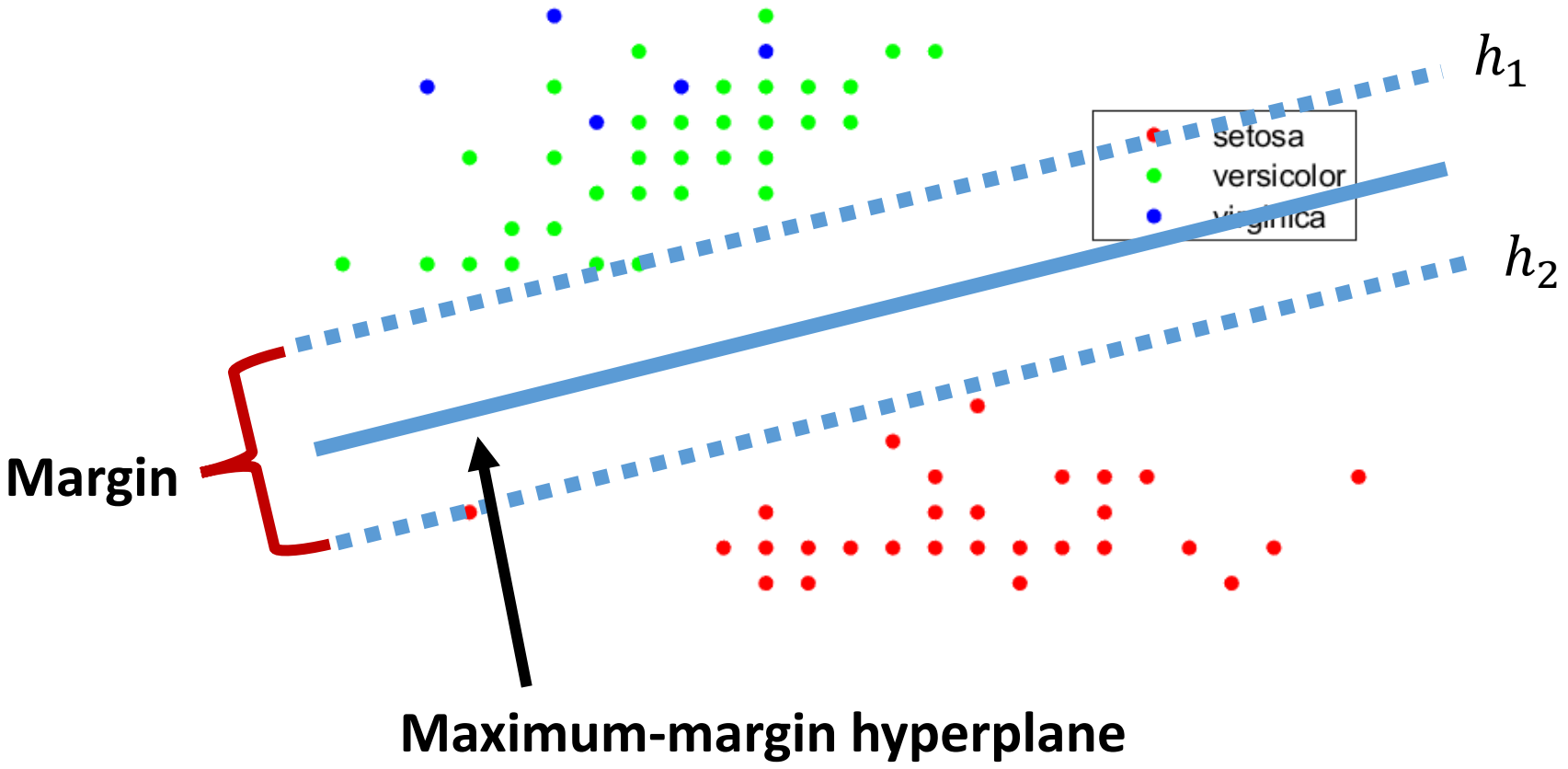
- The two classes of data are separated
  - Anything above h_1 is in class S_1
  - Anything below h_2 is in class S_2
- The distance between h_1 and h_2 is as large as possible

The region between the hyperplanes is called the ***margin***.

The hyperplane lying halfway between h_1 and h_2 is called the ***maximum-margin hyperplane.***

Building SVM

Margins



Building SVM

Margins

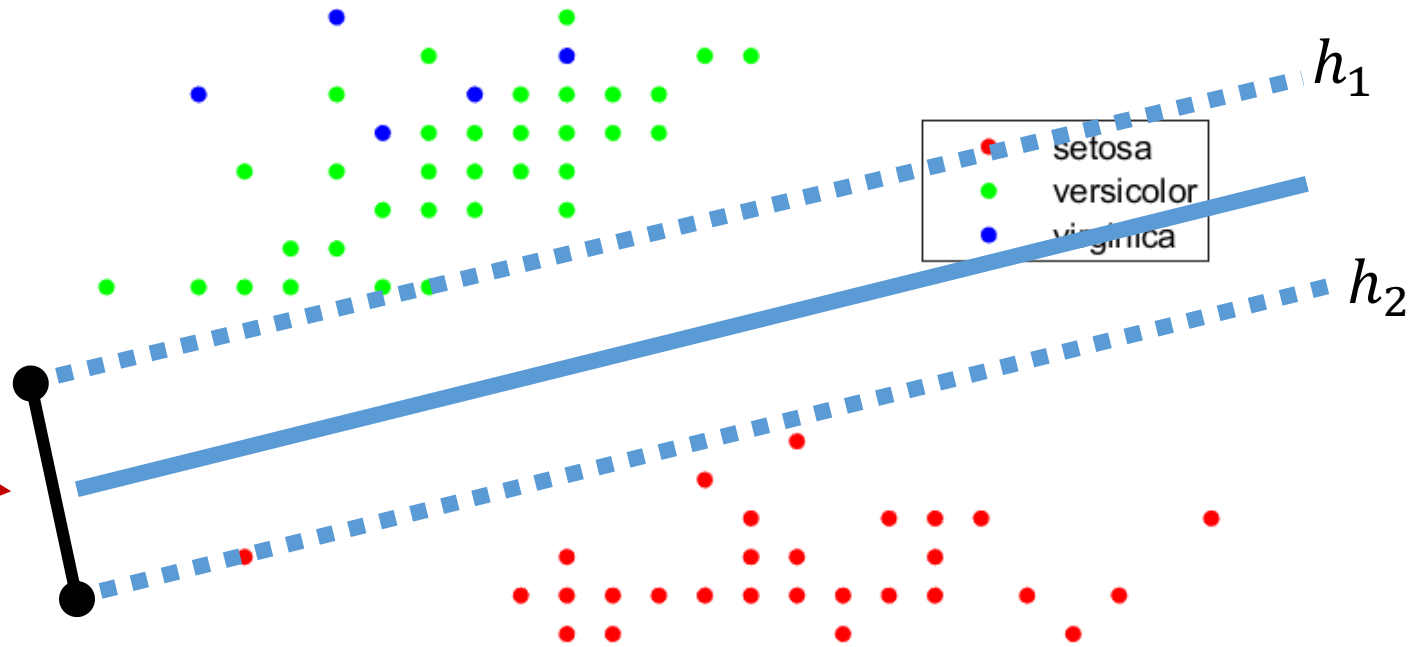
We can define these hyperplanes as

$$h_1\colon\ w^\top x - b = 1$$
$$h_2\colon w^\top x - b = -1$$

The distance between these is

$$\frac{2}{\|w\|}$$

To maximise this distance is the same thing as minimising $\|w\|$



Building SVM

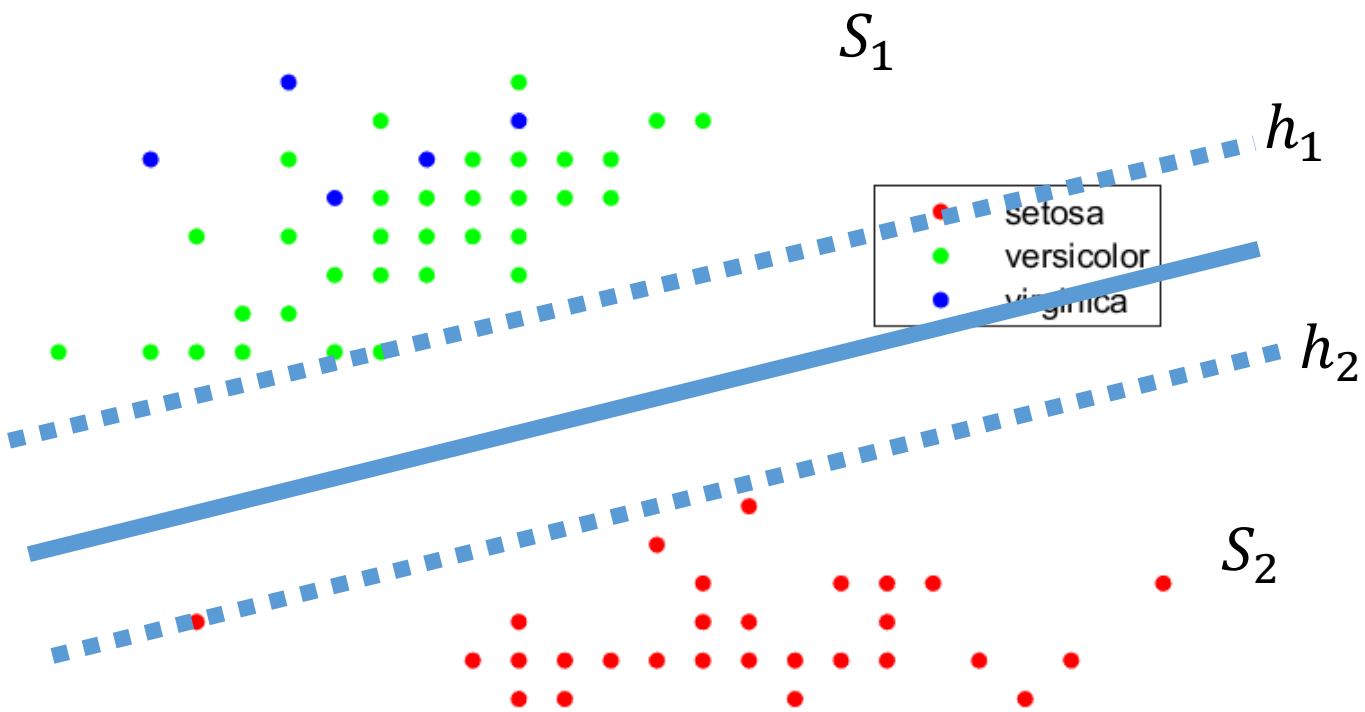
Constraints

In order to keep the points on the correct side of the boundary, we add the following constraints

- $w^\top x_i - b \geq 1$ if $y_i = 1$ (i.e. $x_i \in S_1$)
- $w^\top x - b \leq -1$ if $y_i = -1$ (i.e. $x_i \in S_2$)

We can rewrite this as

$$y_i\left(w^\top x_i - b\right) \geq 1 \text{ for all } i = 1, \dots, n$$



Building SVM

Aim

So our aim is to

$$\text{minimise } \|w\|$$

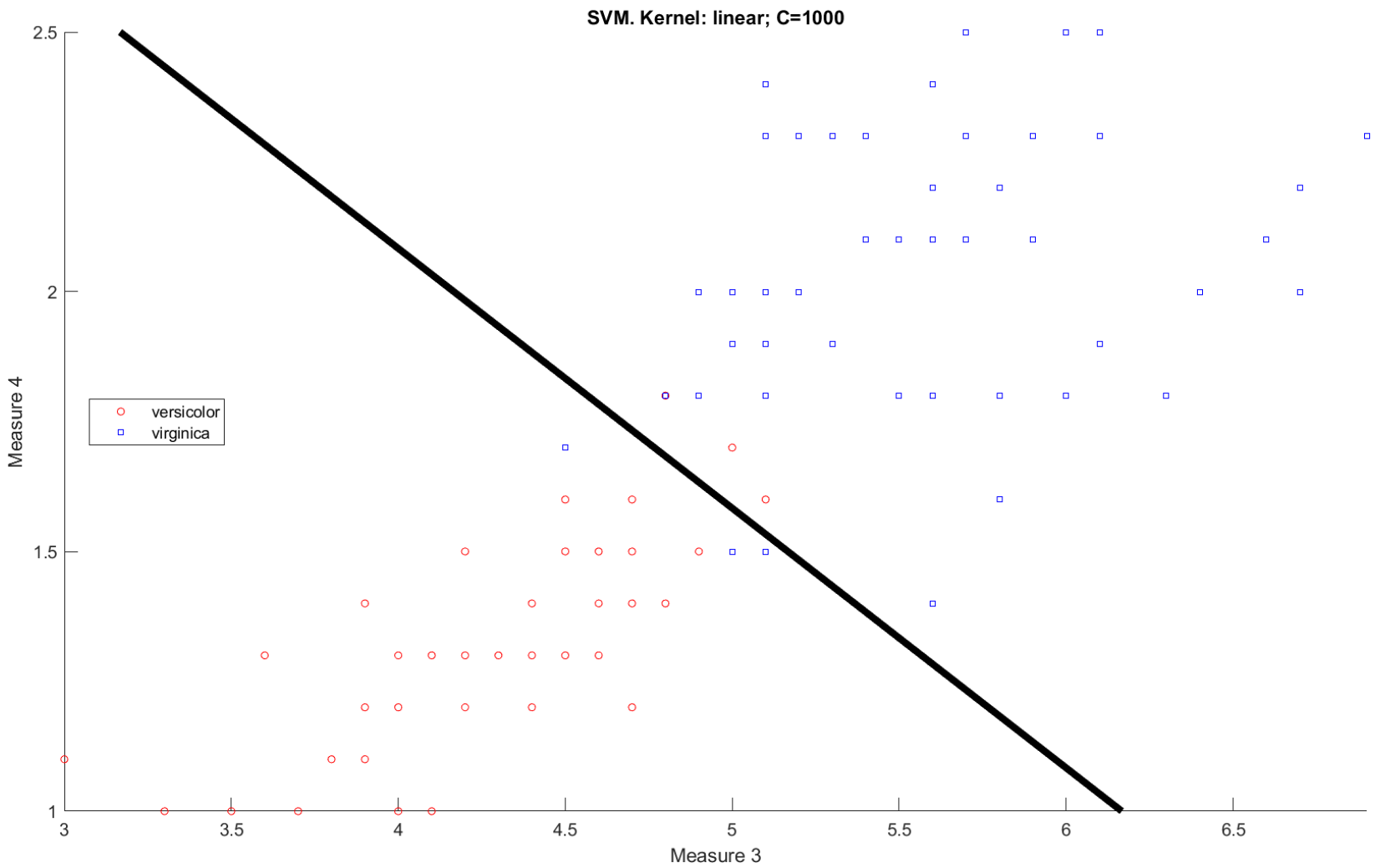
subject to the condition

$$y_i\left(w^\top x_i - b\right) \geq 1 \text{ for all } i = 1, \dots, n$$

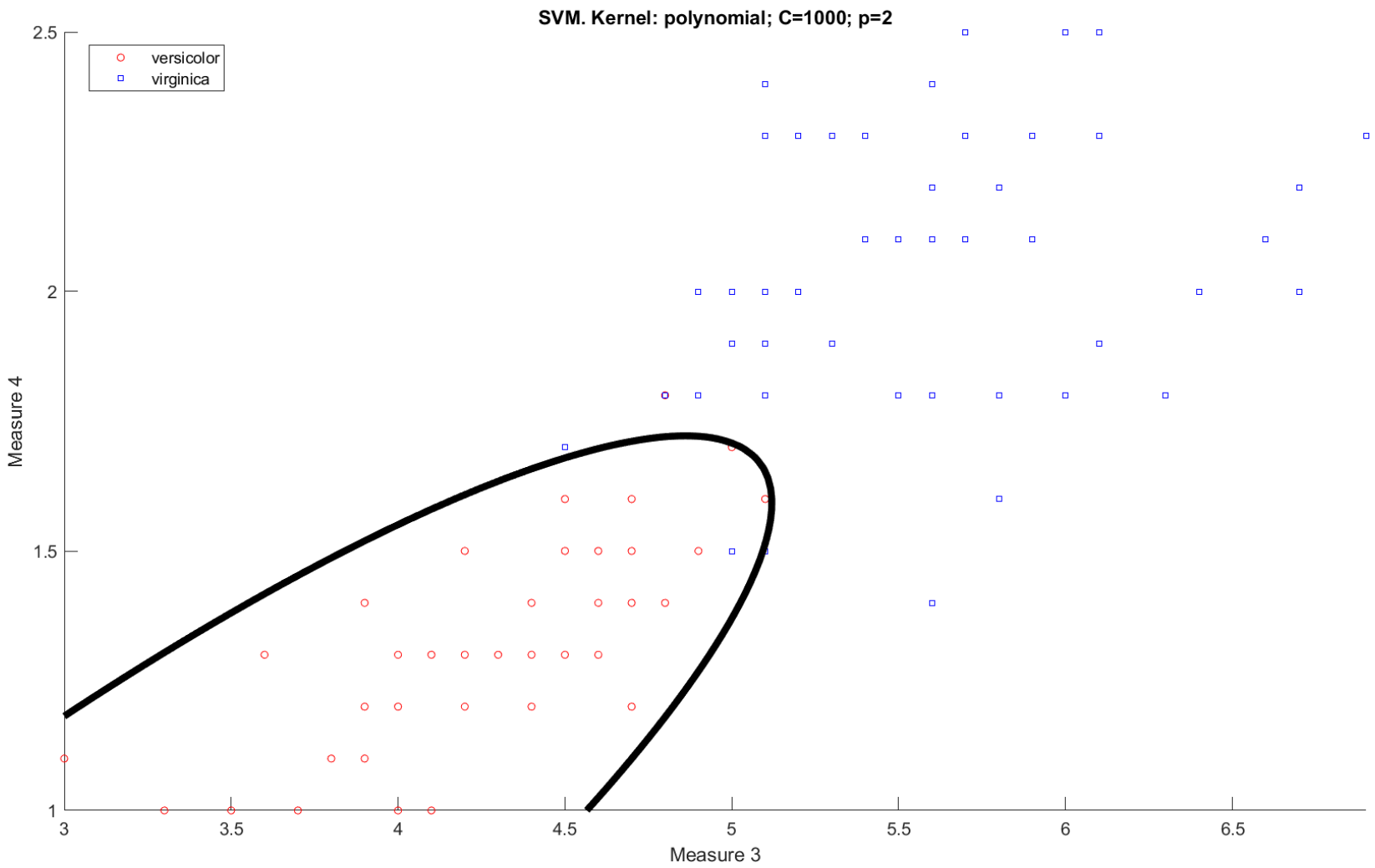
The values of w and b that solve this problem determine our classifier

$$x \mapsto \operatorname{sgn}\left(w^\top x - b\right)$$

Support Vector Machines



Support Vector Machines



Kernel Trick

We can create **nonlinear** classifiers by applying a kernel trick to maximum-margin hyperplanes

Linear:

$$k(x_i, x_j) = x_i \cdot x_j$$

Polynomial:

$$k(x_i, x_j) = (x_i \cdot x_j)^p$$

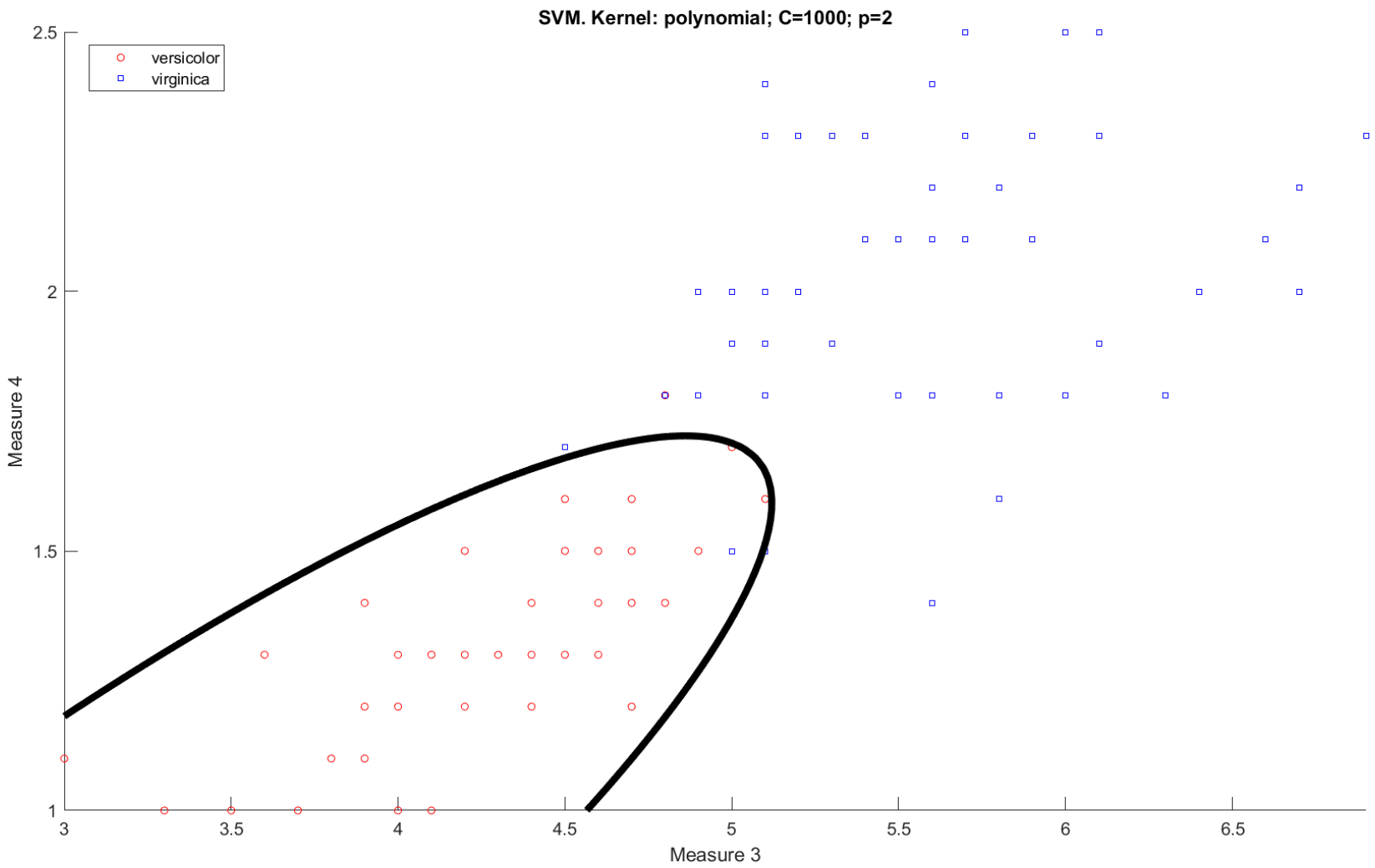
Radial Basis Function:

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right)$$

Sigmoid Function:

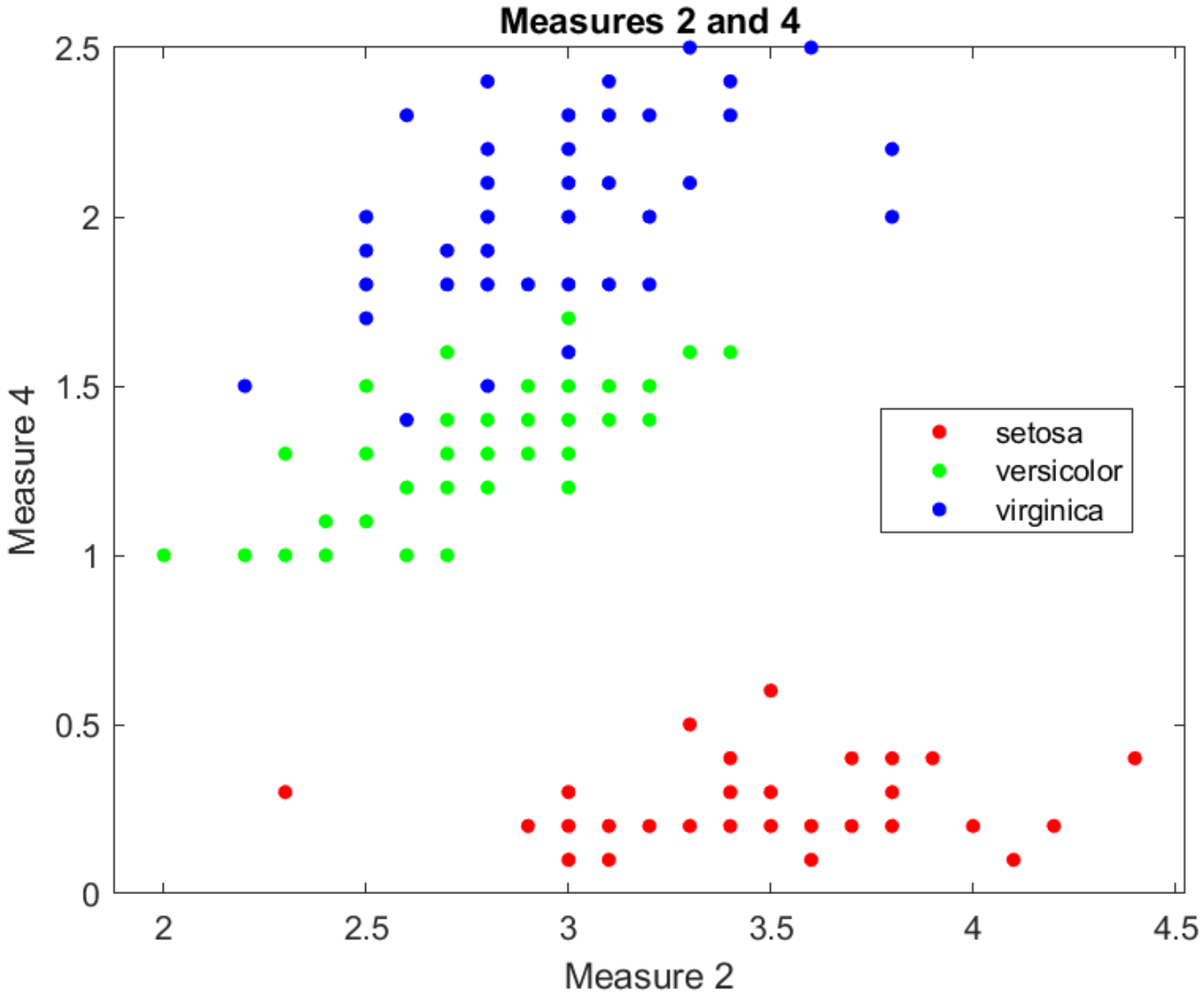
$$k(x_i, x_j) = \tanh(\kappa_1 + \kappa_2 x_i \cdot x_j)$$

Kernel Trick



Multiclass SVM

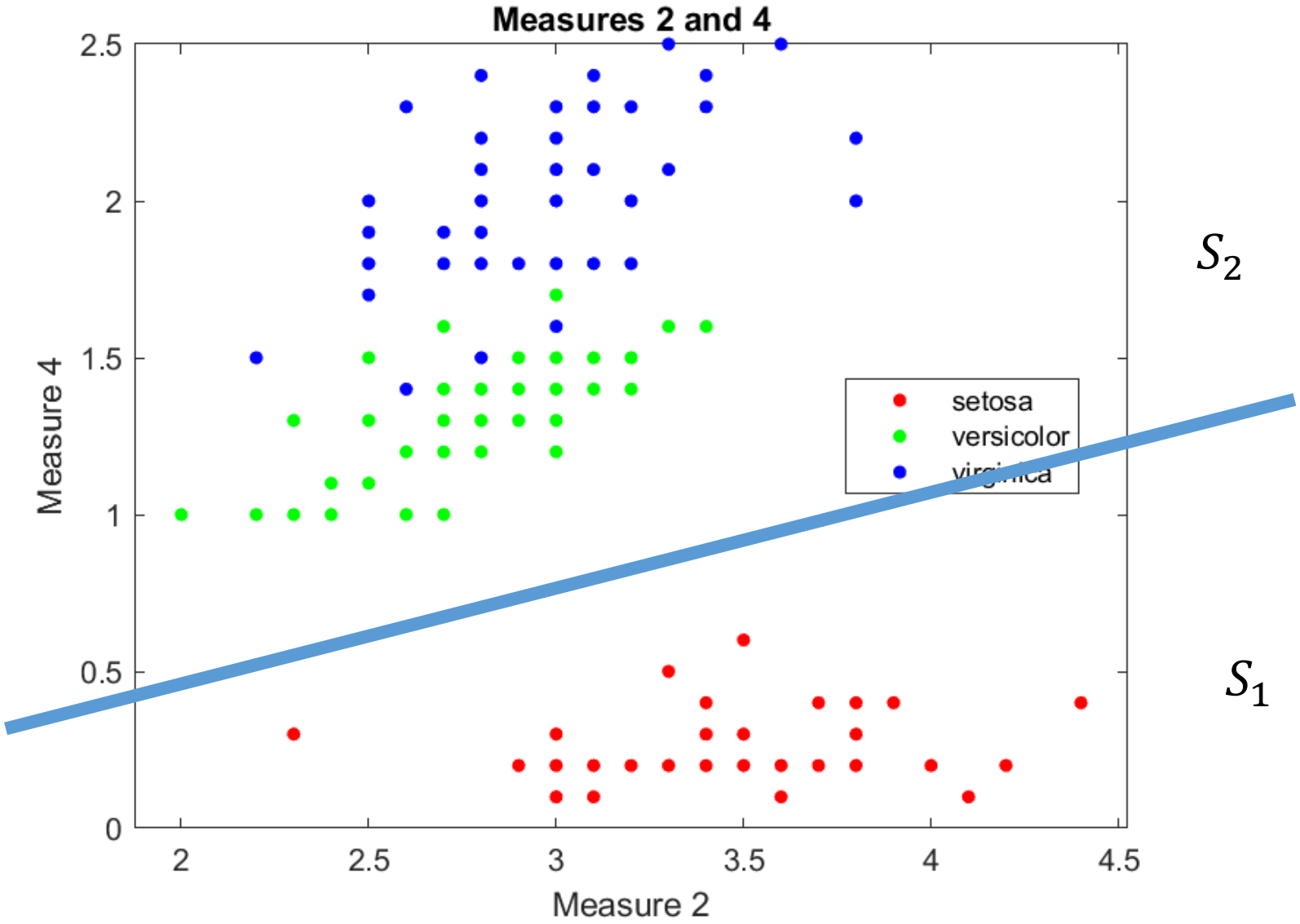
Aim: Create classifier for 3 classes



Multiclass SVM

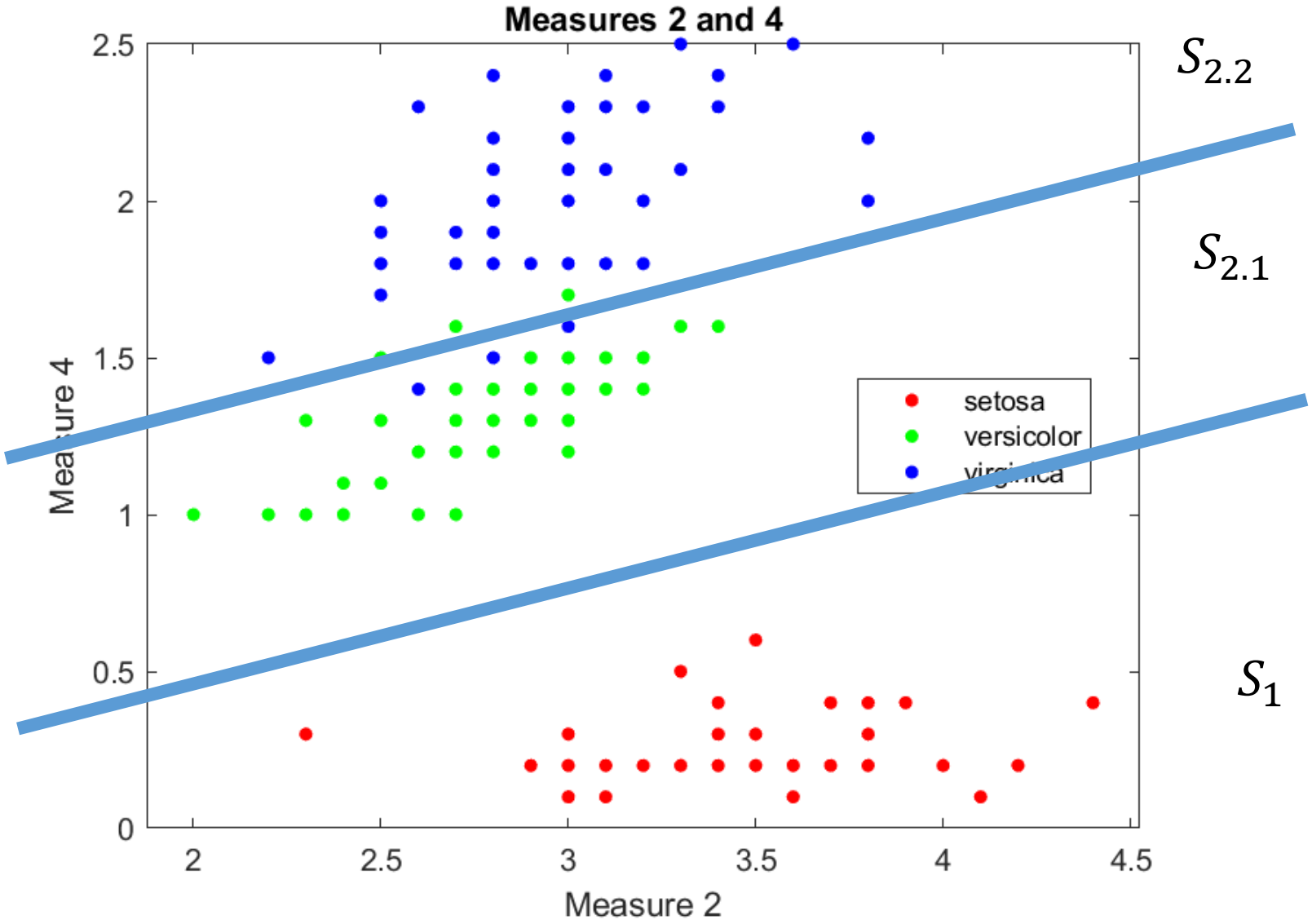
Step 1: combine 2 classes to form a single class

Then you have a binary problem.



Multiclass SVM

Step 2: Train a classifier to separate your combined class.



Artificial Neural Networks
ANN

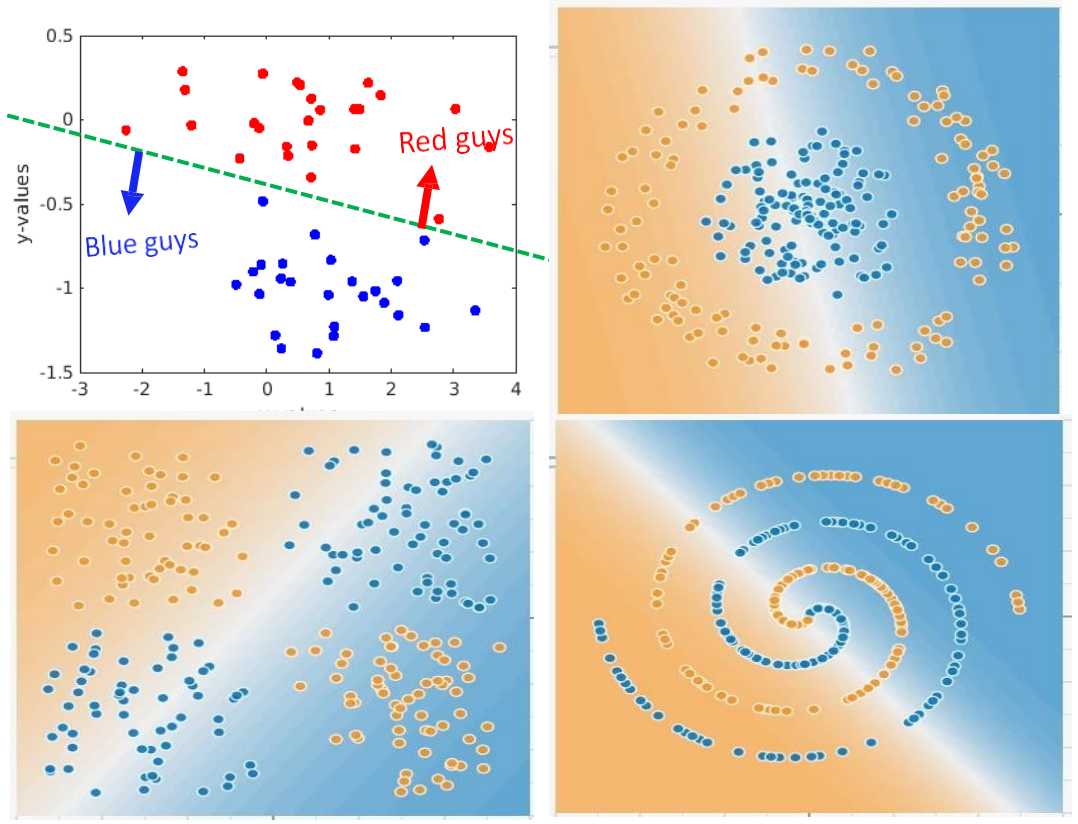
ML: Remaining Problems

Typical datasets are large in size

Data points are not usually well separated by lines

Real world datasets are complex

Classic ML Algorithms struggle to learn from this data



Remaining Problem

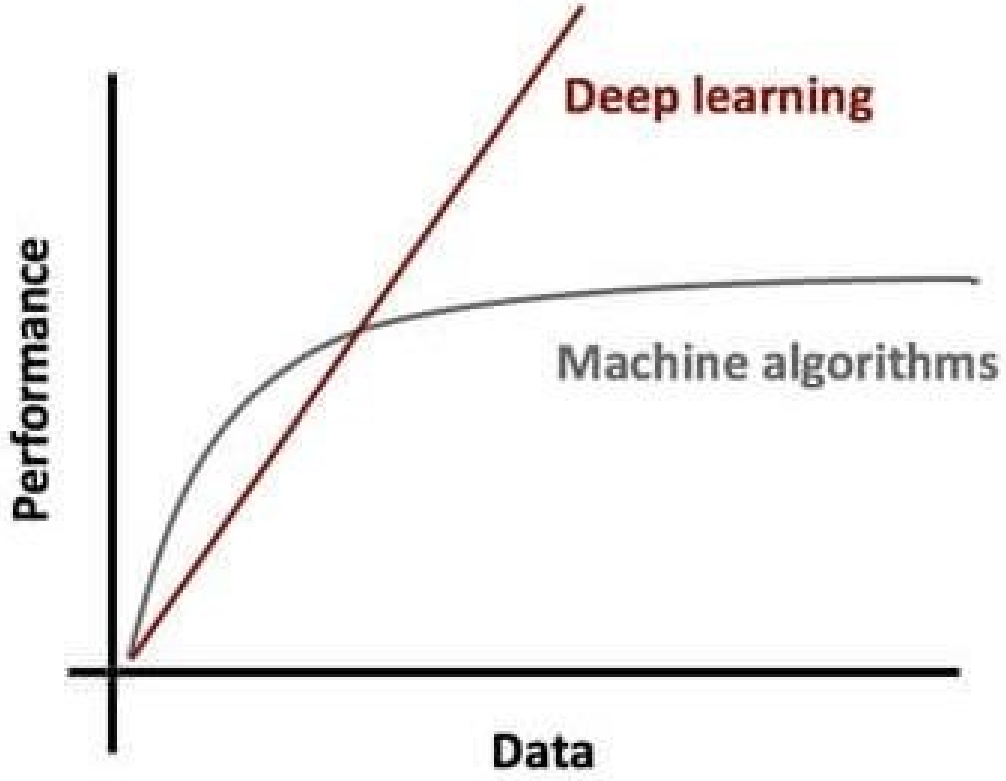
Typical datasets are large in size

Data points are not usually well separated by lines

Real world datasets are complex

ANNs can be scaled to perform well with these difficult data spaces.

Deep neural networks learn better by refining decisions at each layer



Source: https://en.wikipedia.org/wiki/Neuron

Learning the Mapping Function

$$\boldsymbol{f}: \mathbb{R}^n \longrightarrow \{1, \ldots, k\}$$

x_1	x_2	x_3	x_4	x_5	x_6	…	x_n

l_1
l_2
…
l_k

Feature vectors

Iris measurements
Pixel values from hand-written characters
Bag-of-word features from email texts
Temperature and humidity measurements

Classes, labels, categories

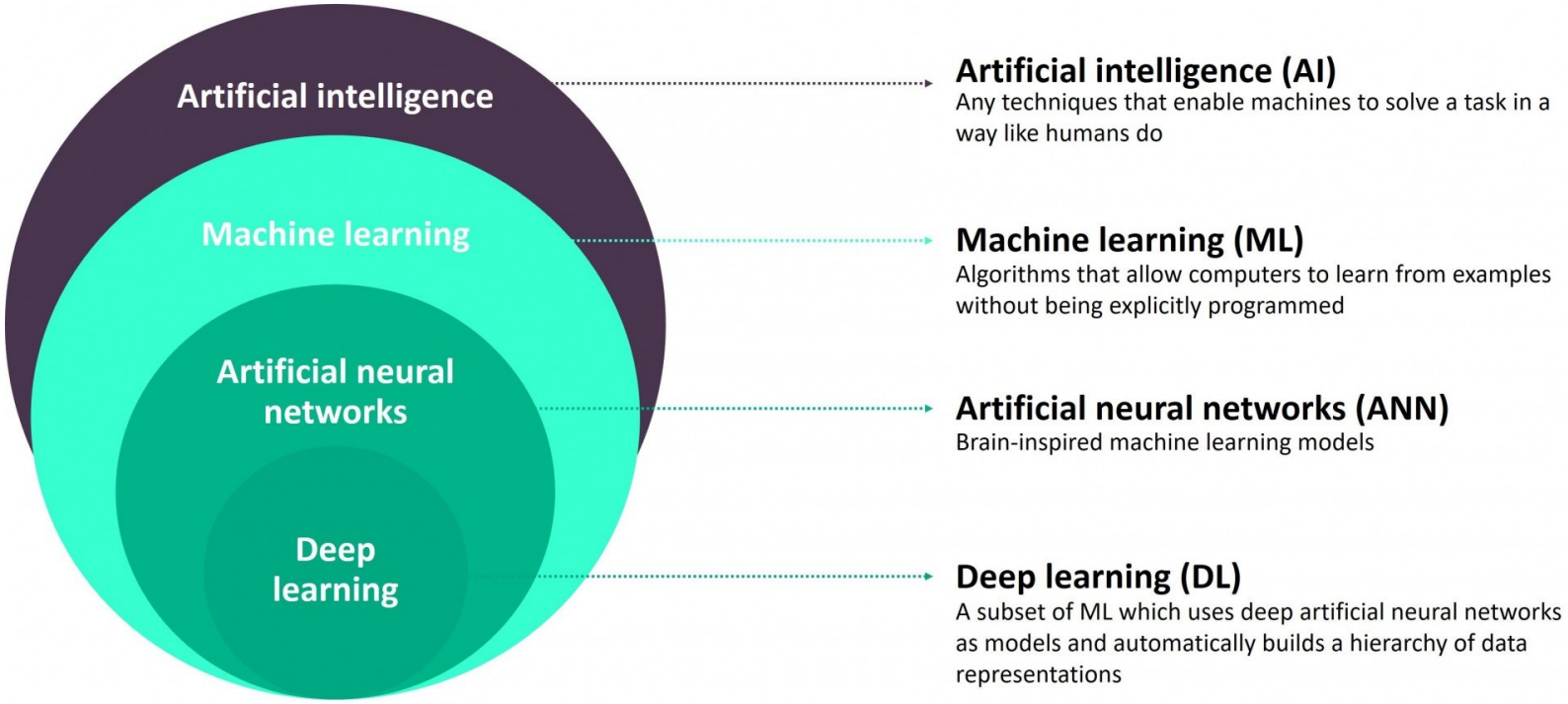
setosa, versicolor and virginica
The actual character (letters or digits)
Spam or Ham (i.e. Not spam)
Rain or No rain

What are Artificial Neural Networks?

They are mathematical models for learning inspired by the biological neural networks

They model mathematical function inputs to outputs based on the structure and parameters of the network

Allows for learning the network's parameters based on data



The Human Brain

- Our ability to think, learn and function is controlled by the brain
- It does this through the electrically excitable cells called neurons
- The brain contains ~86 billion neurons with special connections called **synapses**
- A group of connected neurons is called a **neural circuit**

Source: https://en.wikipedia.org/wiki/Neuron

The Neuron

- A typical neuron consists of these key parts

Soma

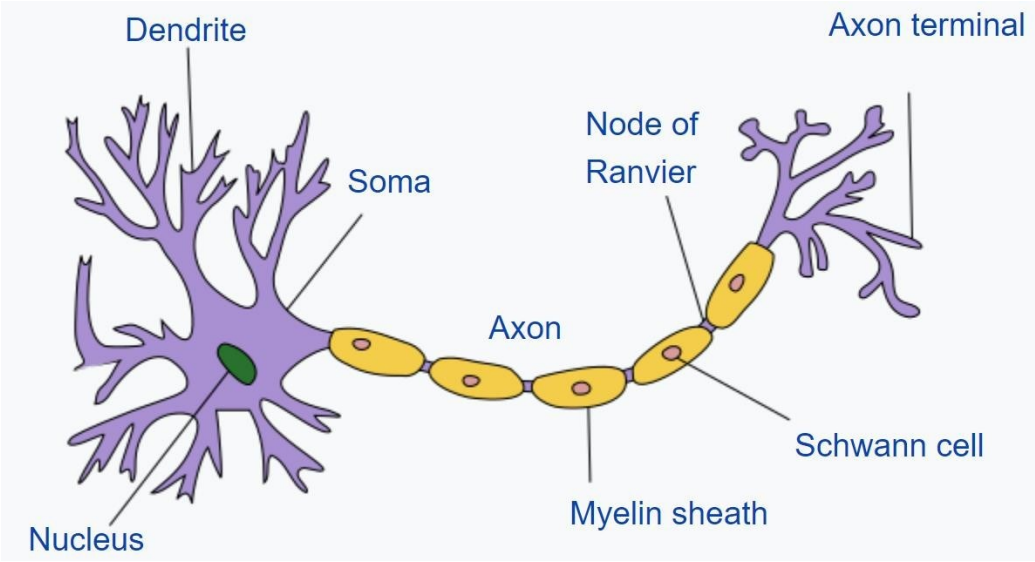
- The body of the neuron

Dendrites

- Cellular extensions with many branches

A single axon

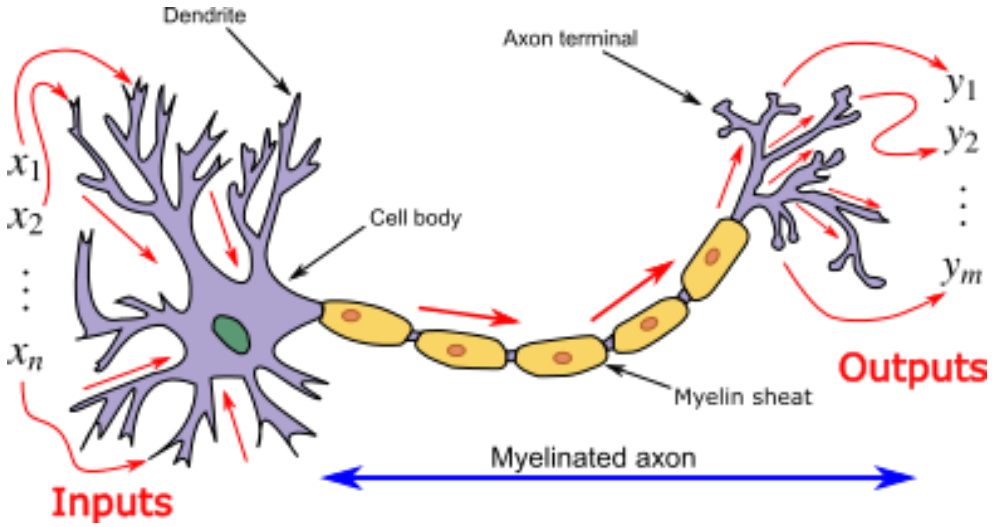
- Fine cable-like projections carrying nerve signals



Images Source: https://en.wikipedia.org/wiki/Neuron

Artificial Neural Networks

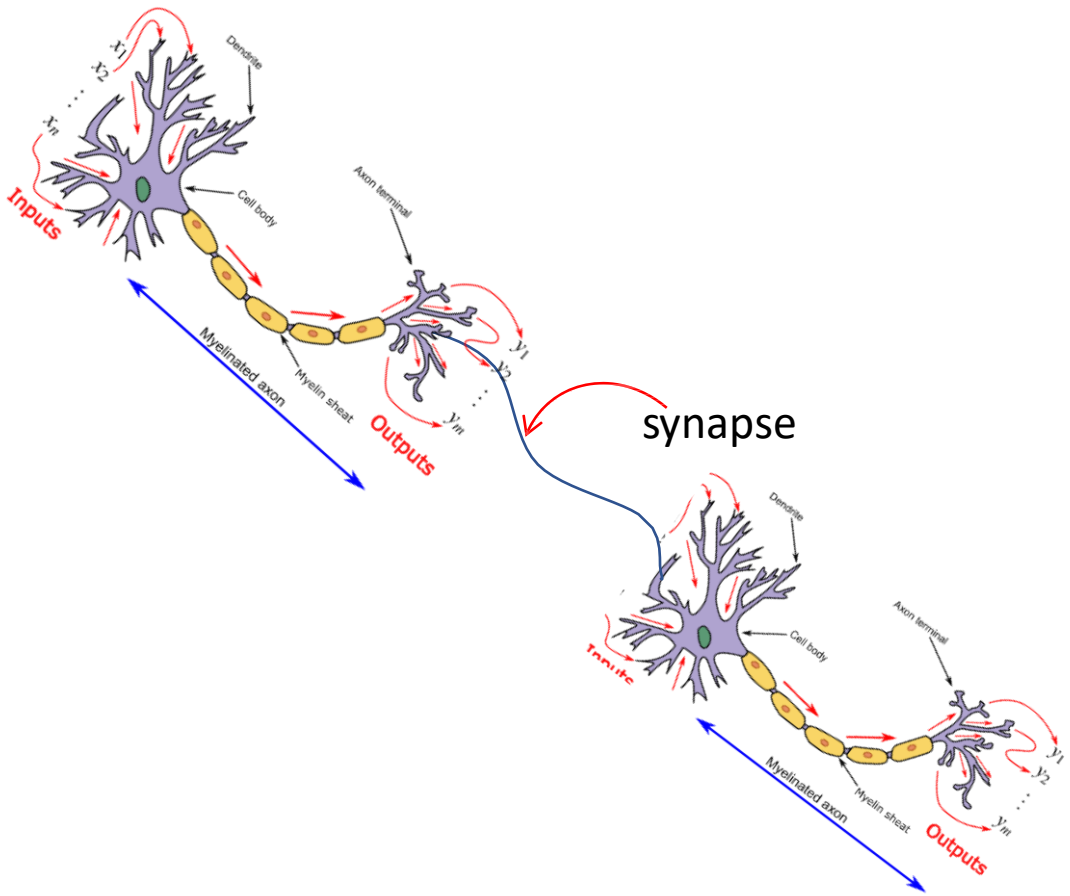
- ANN is a machine learning approach modelled after the human brain
- It is a network of artificial ‘neurons’
  - aka **units** or **nodes**



Images Source: https://en.wikipedia.org/wiki/Artificial_neural_network

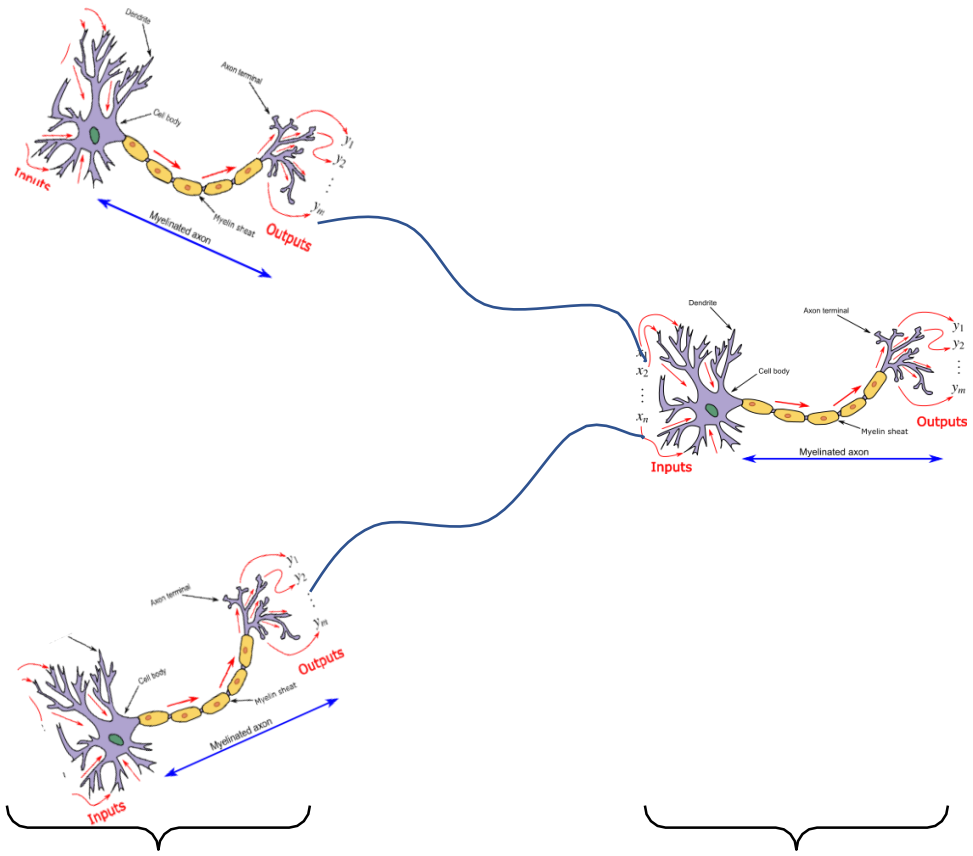
Artificial Neural Networks

- ANN is a machine learning approach modelled after the human brain
- It is a network of artificial ‘neurons’
  - aka **units** or **nodes**
- Each neuron receives an *input* signal and propagates same to a connected neuron through the **synapses**



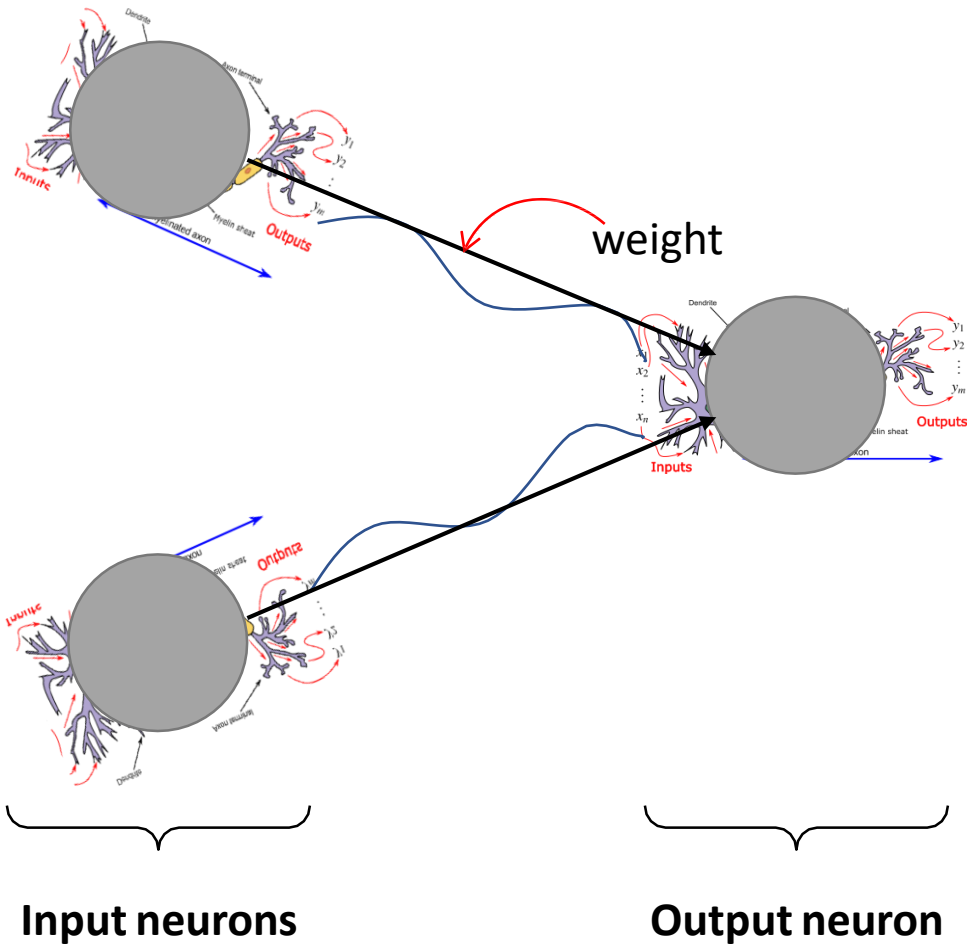
Artificial Neural Networks

- ANN is a machine learning approach modelled after the human brain
- It is a network of artificial ‘neurons’
  - aka **units** or **nodes**
- Each neuron receives an *input* signal and propagates same to a connected neuron through the **synapses**
- The *input neurons* are at one end while the *output* neuron is at the other



Artificial Neural Networks

- ANN is a machine learning approach modelled after the human brain
- It is a network of artificial ‘neurons’
  - aka **units** or **nodes**
- Each neuron receives an *input* signal and propagates same to a connected neuron through the **synapses**
- The *input neurons* are at one end while the *output* neuron is at the other

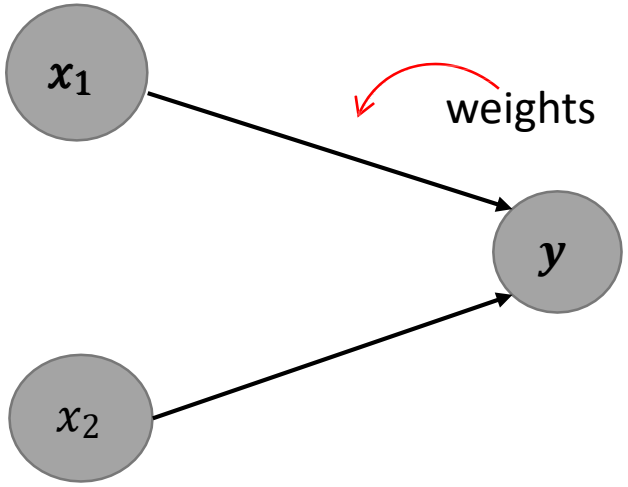


Hypothesis Function

$$\boldsymbol{f}: \mathbb{R}^n \longrightarrow \{1, \ldots, k\}$$

x_1	x_2

1: *Rain*
0: *No Rain*



Rain or No rain

Temperature and **humidity** measurements

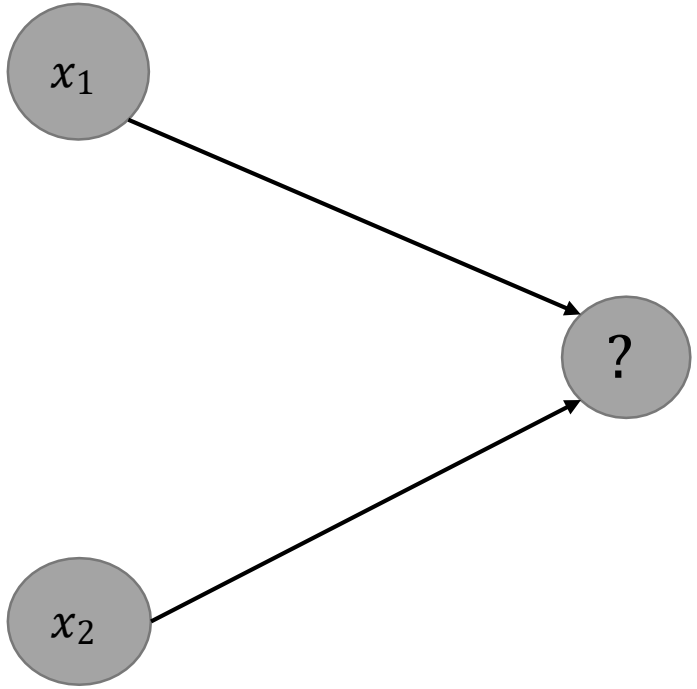
Hypothesis Function

$$\boldsymbol{f}: \mathbb{R}^n \longrightarrow \{1, \dots, k\}$$

Required function or solution but we don't know what it is

$$\boldsymbol{h}(\boldsymbol{x}_1, \boldsymbol{x}_2)$$

So we begin by defining a hypothesis function

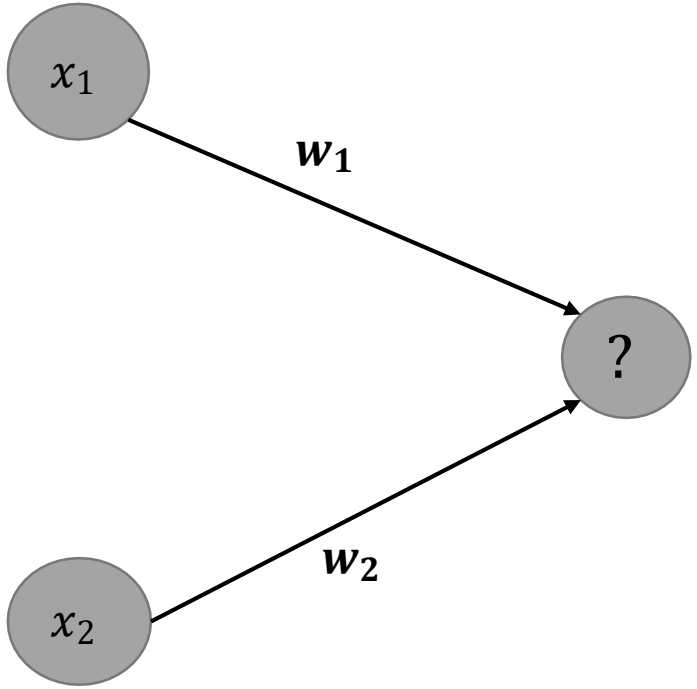


Hypothesis Function

$$\boldsymbol{f}: \mathbb{R}^n \longrightarrow \{1, \dots, k\}$$

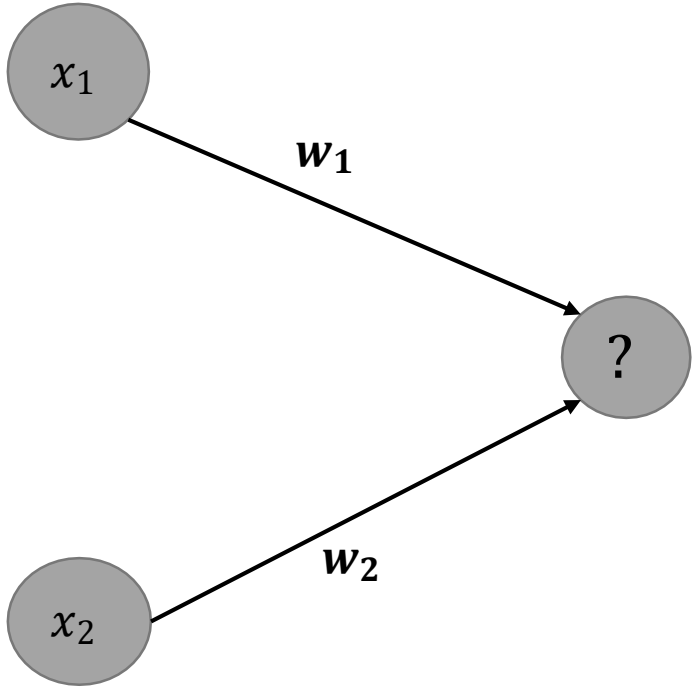
Required function or solution but we don't know what it is

$$\boldsymbol{h}(\boldsymbol{x}_1, \boldsymbol{x}_2)$$



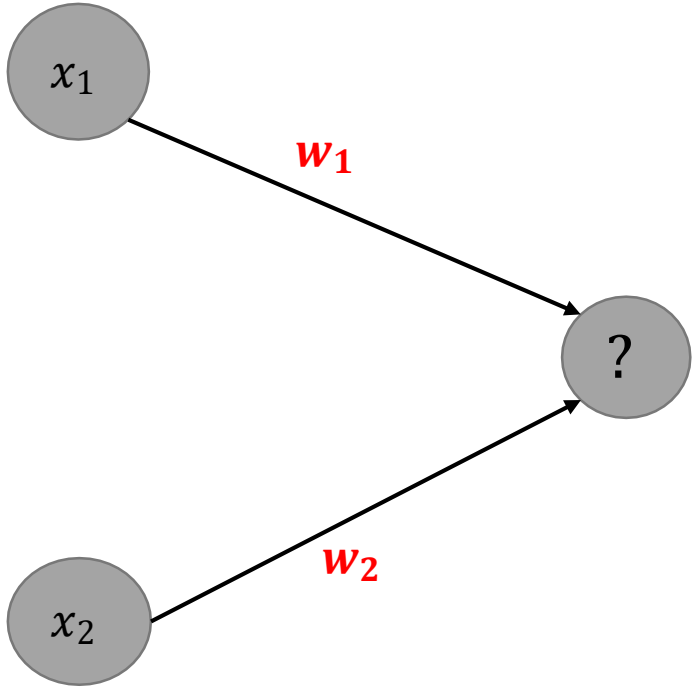
Hypothesis Function

$$\boldsymbol{h(x_1, x_2) = w_1 x_1 + w_2 x_2}$$



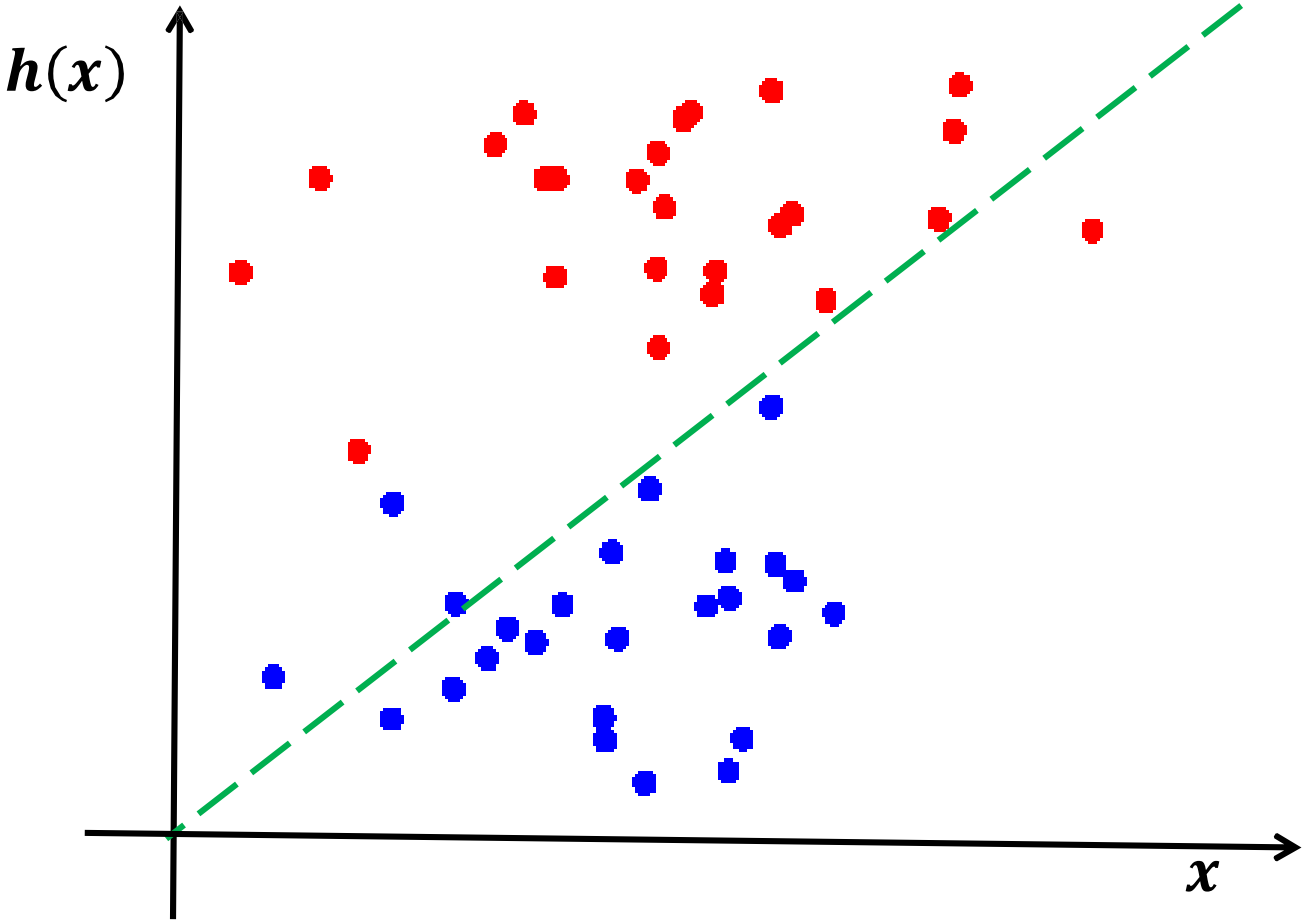
Hypothesis Function

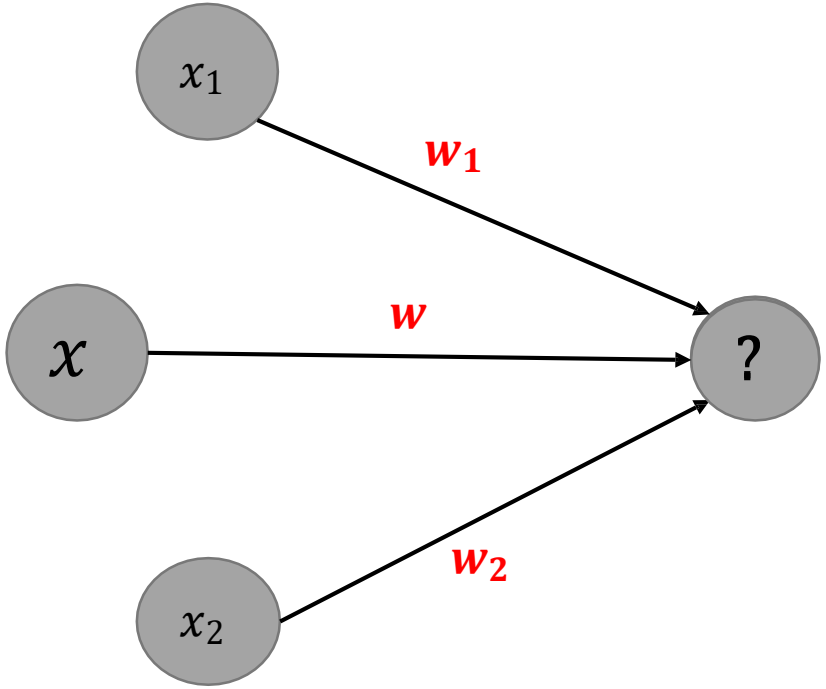
$$h(x_1, x_2) = w_1 x_1 + w_2 x_2$$



Hypothesis Function - Bias

$$h(x_1, x_2) = w_1 x_1 + w_2 x_2$$

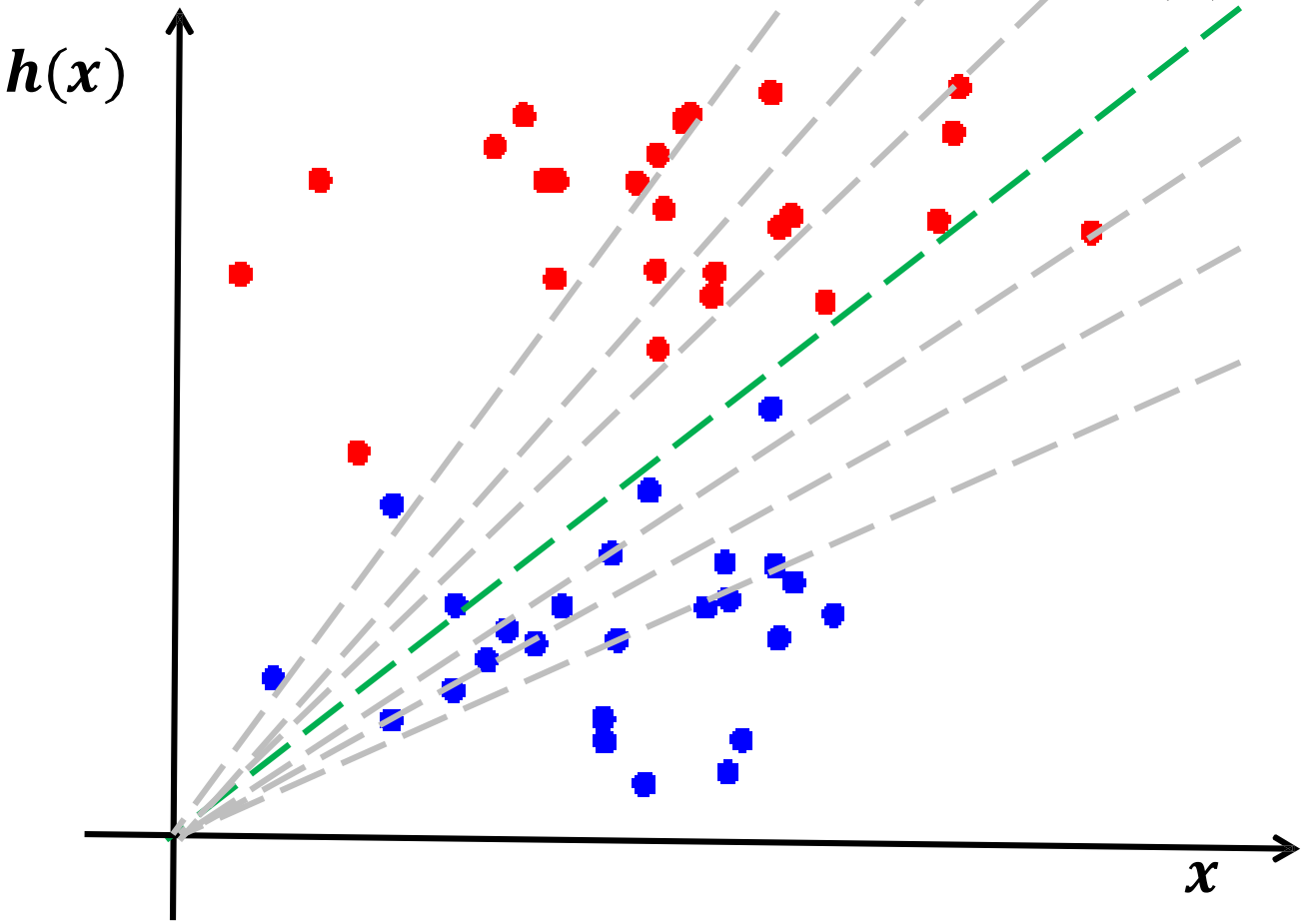


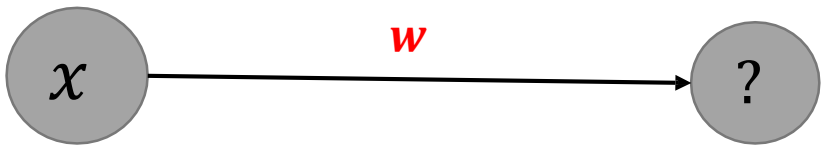


Hypothesis Function - Bias

$$y = mx$$

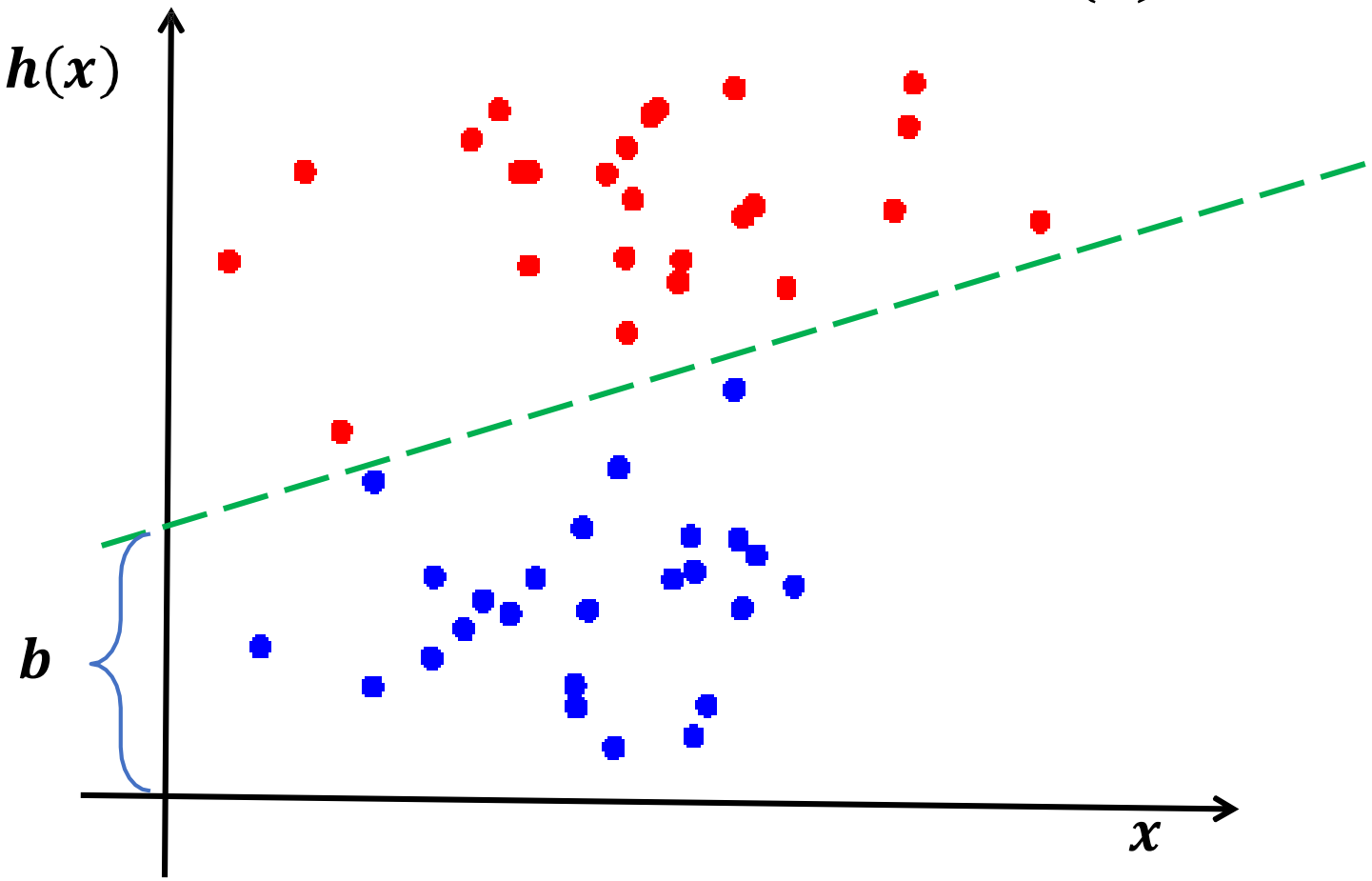
$$h(x) = wx$$

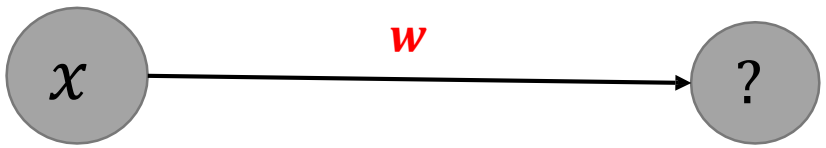




Hypothesis Function - Bias

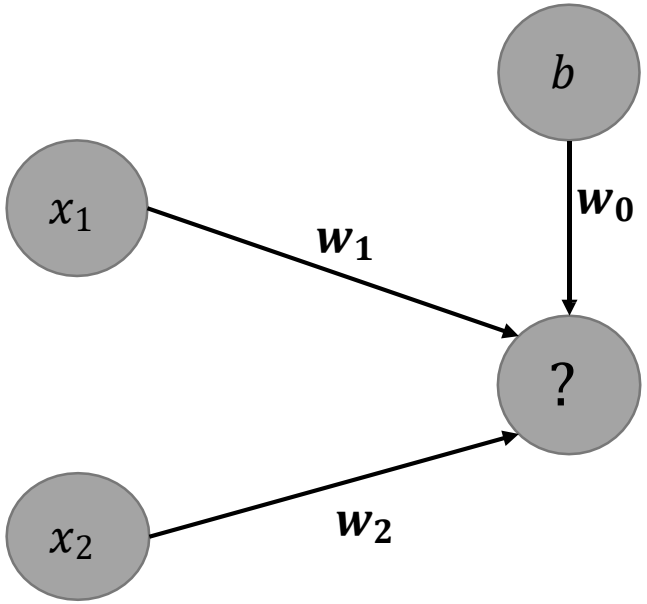
$$y = mx + c$$
$$h(x) = wx + b$$





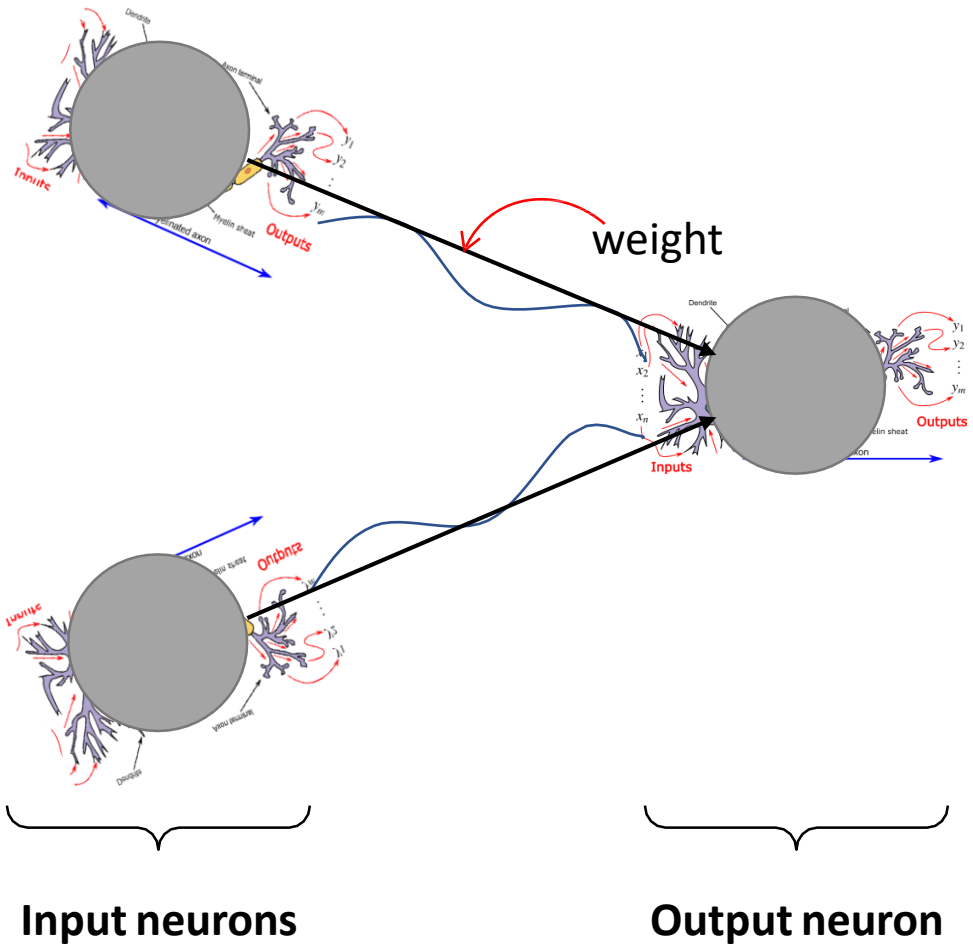
Hypothesis Function - Bias

$$\boldsymbol{h(x_1, x_2) = w_0 b + w_1 x_1 + w_2 x_2}$$



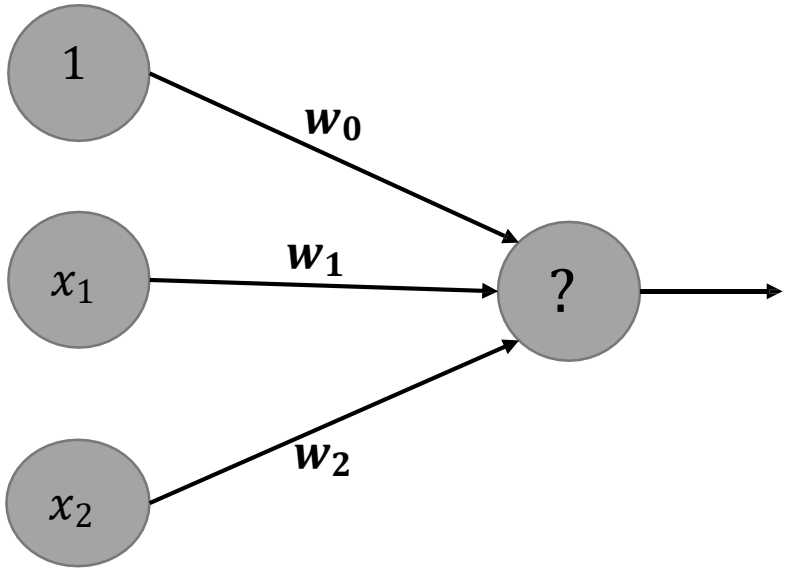
Yesterday: Artificial Neural Networks

- ANN is a machine learning approach modelled after the human brain
- It is a network of artificial ‘neurons’
  - aka **units** or **nodes**
- Each neuron receives an *input* signal and propagates same to a connected neuron through the **synapses**
- The *input neurons* are at one end while the *output* neuron is at the other



Hypothesis Function

$$\boldsymbol{h(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2}$$

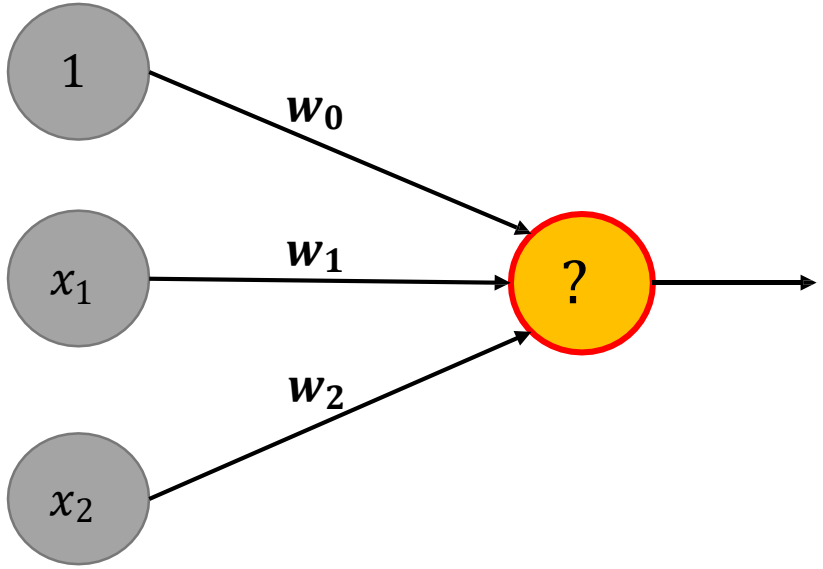


Hypothesis Function - Example

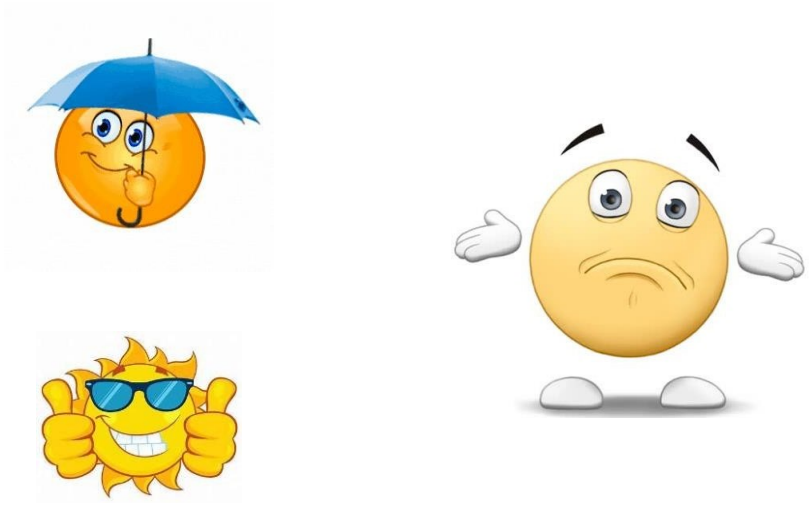
$$h(x_1, x_2) = w_0 + w_1 x_1 + w_2 x_2$$

Given these values for the variables of the hypothesis function

- What will be the outcome of the hypothesis function?
- Will it **rain** or **not**?



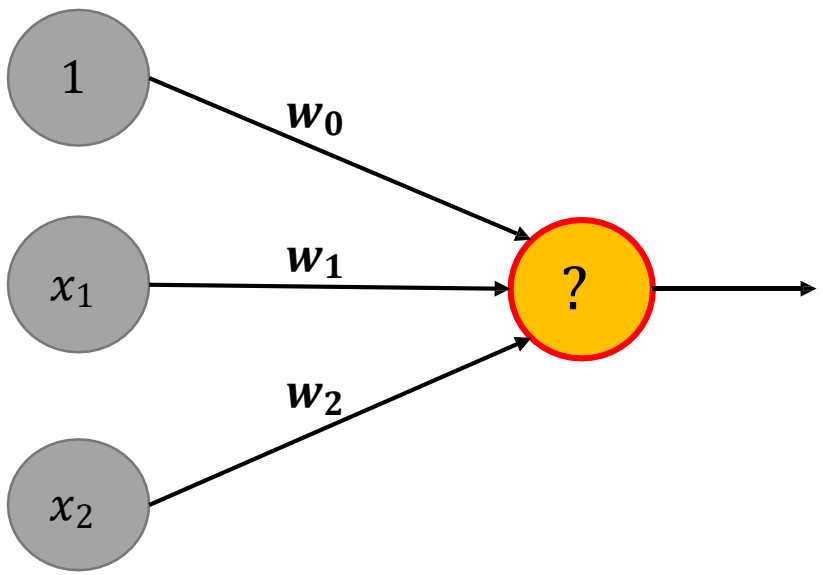
x_1	9
x_2	0.83
w_0	1.512
w_1	3.674
w_2	-2.418



Source: [Wikipedia Activation Functions]

Activation Function

$$h(x_1, x_2) = g(w_0 + w_1 x_1 + w_2 x_2)$$

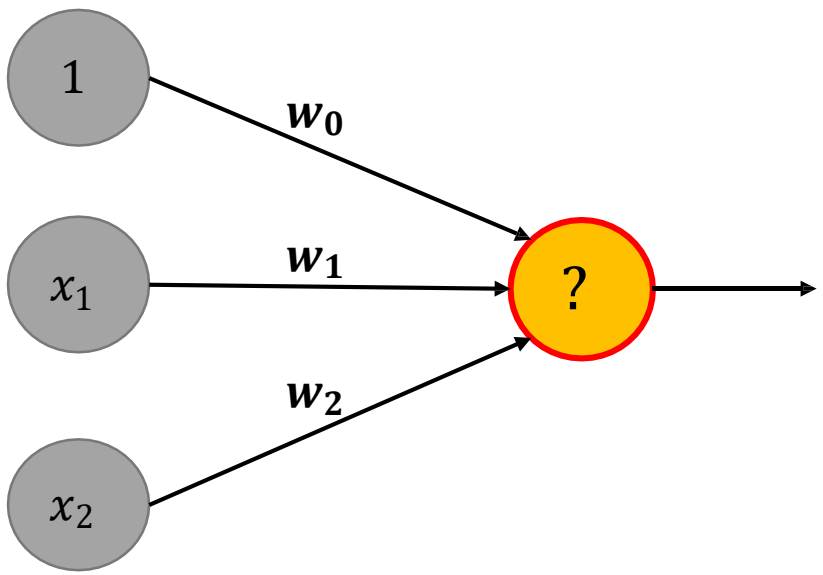


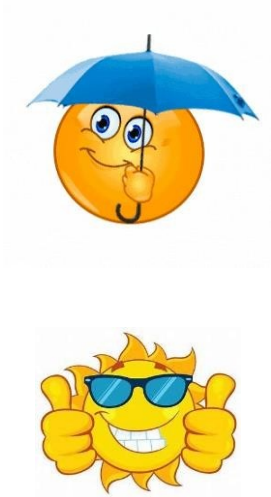
- The **activation function** transforms the output of our hypothesis function to the predicted value or class
- It defines the threshold and **activates** the right outputs node
- Examples include:
  - Binary step function
  - Logistic Sigmoid function
  - Rectified Linear Units (ReLU)

Source: [Wikipedia Activation Functions]

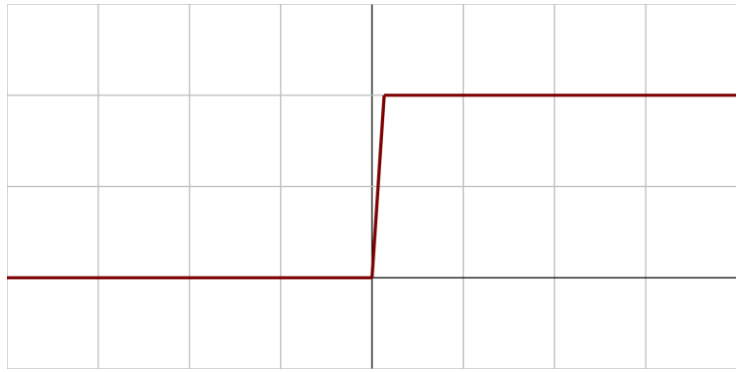
Activation Function

$$h(x_1, x_2) = g(w_0 + w_1 x_1 + w_2 x_2)$$





$$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$

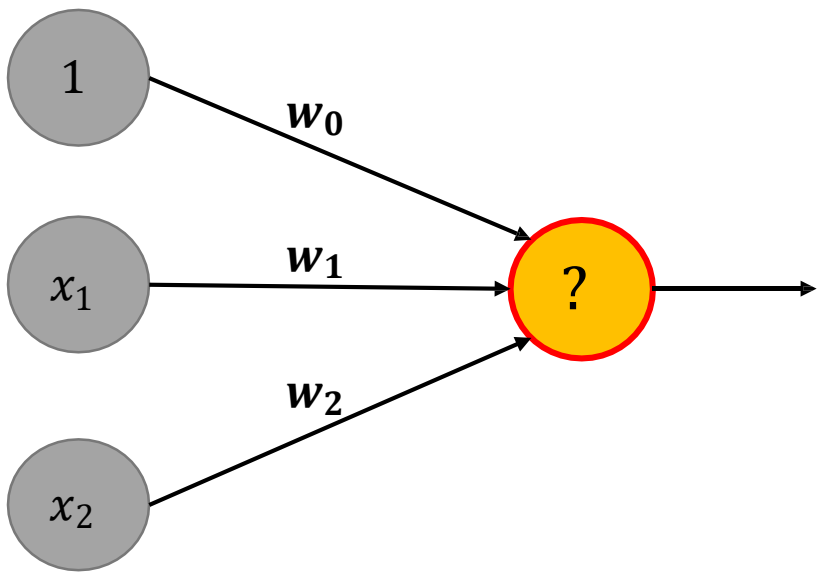


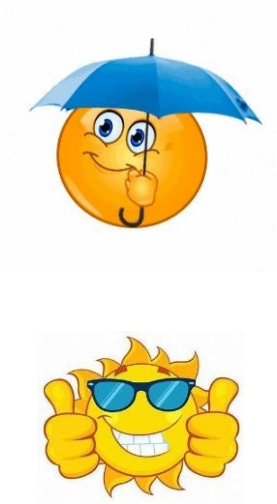
Binary Step Function

Source: [Wikipedia Activation Functions]

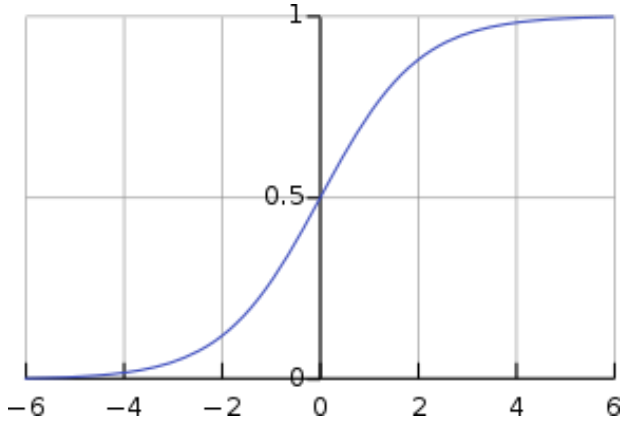
Activation Function

$$h(x_1, x_2) = g(w_0 + w_1 x_1 + w_2 x_2)$$





$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1}$$

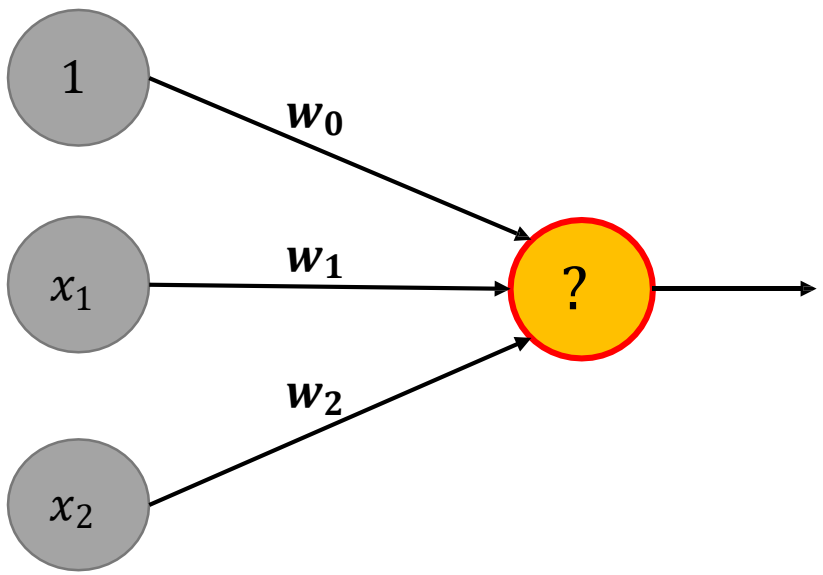


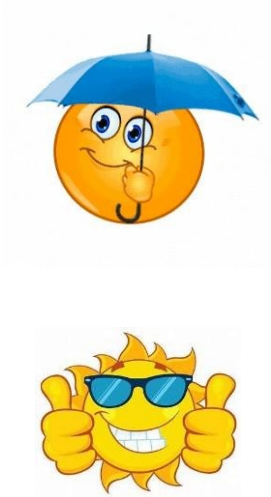
Logistic Sigmoid Function

Source: Wikipedia Activation Functions

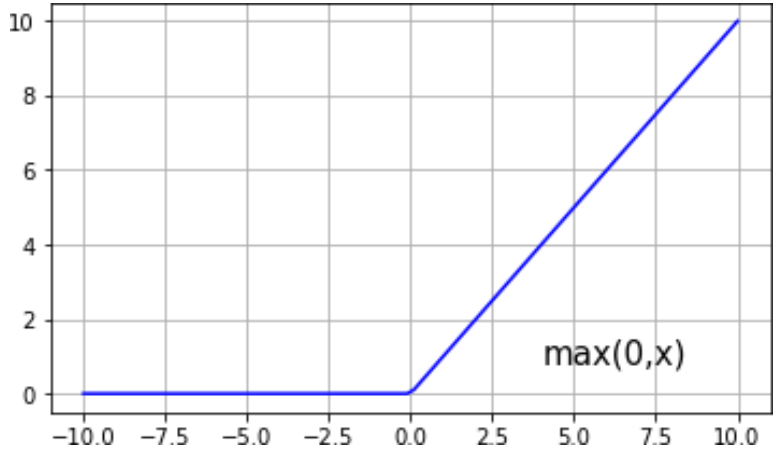
Activation Function

$$h(x_1, x_2) = g(w_0 + w_1 x_1 + w_2 x_2)$$





$$f(x) = \begin{cases} 0 & \text{for } x \leq 0 \\ x & \text{for } x > 0 \end{cases} = \max\{0, x\}$$



Rectified Linear Unit

Source: [Wikipedia Activation Functions]

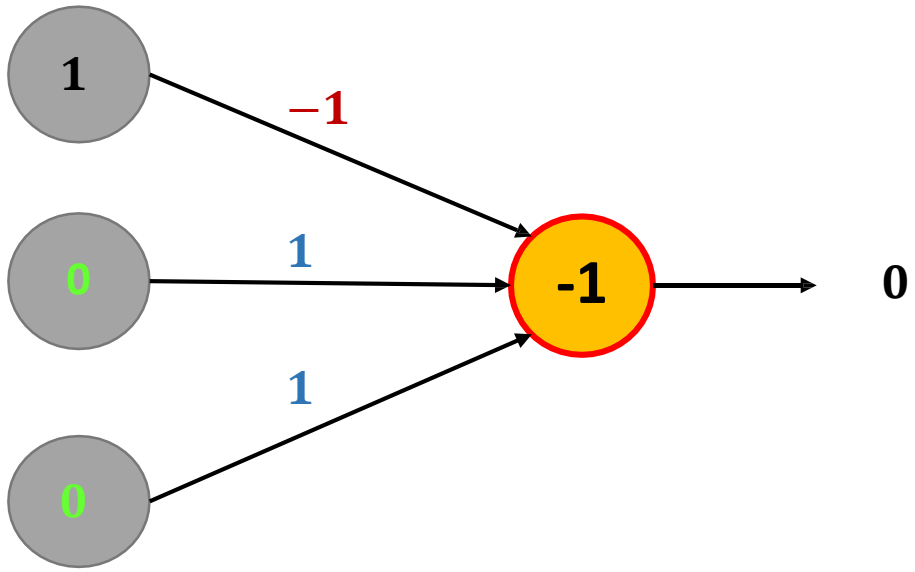
The OR Function

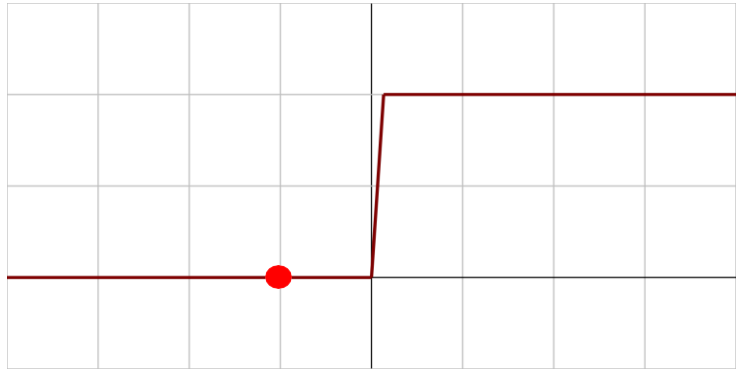
$$OR(x_1, x_2) = \begin{cases} 1 \text{ if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 \text{ otherwise} \end{cases}$$

$$h(x_1, x_2) = g(-1 + 1 \cdot x_1 + 1 \cdot x_2)$$

x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

$$g(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$





Binary Step Function

Source: [Wikipedia Activation Functions]

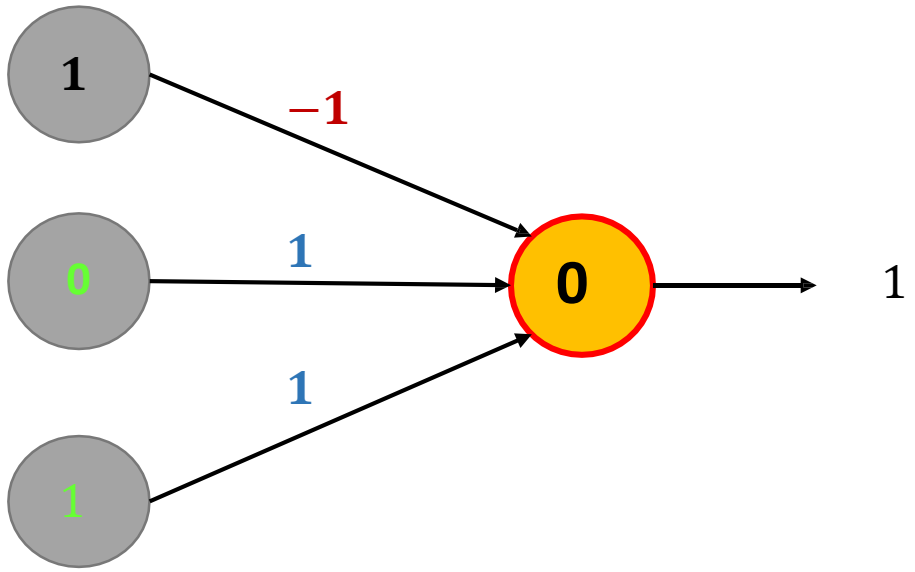
The OR Function

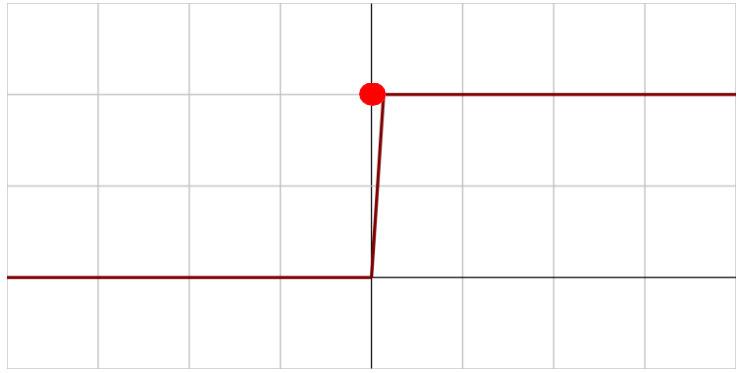
$$OR(x_1, x_2) = \begin{cases} 1 \text{ if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 \text{ otherwise} \end{cases}$$

$$h(x_1, x_2) = g(-1 + 1 \cdot x_1 + 1 \cdot x_2)$$

x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

$$g(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$





Binary Step Function

Source: [Wikipedia Activation Functions]

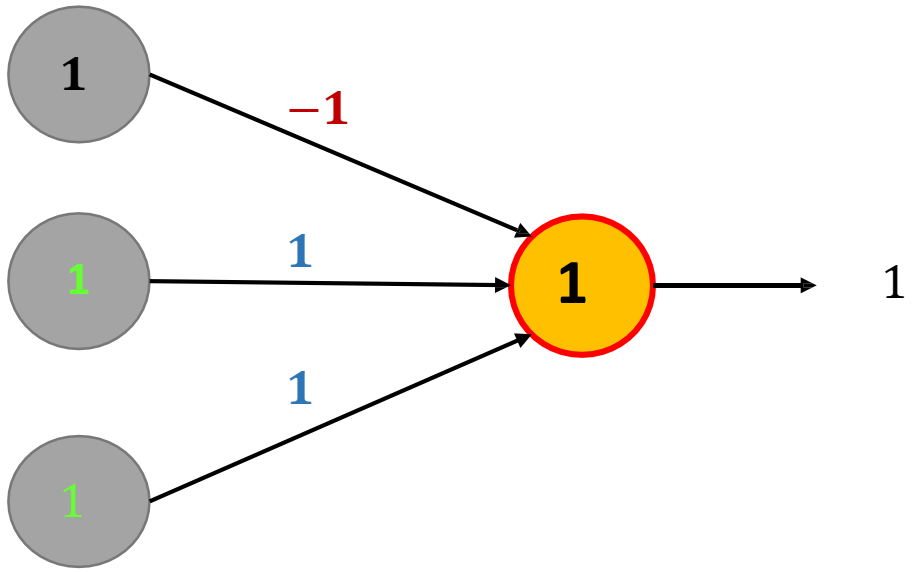
The OR Function

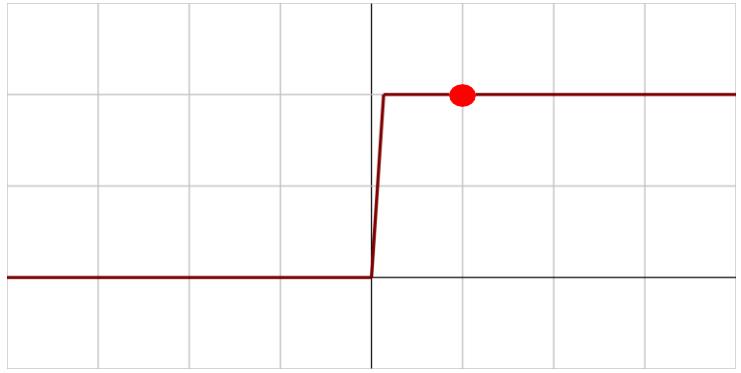
$$OR(x_1, x_2) = \begin{cases} 1 \text{ if } x_1 = 1 \text{ or } x_2 = 1 \\ 0 \text{ otherwise} \end{cases}$$

$$h(x_1, x_2) = g(-1 + 1 \cdot x_1 + 1 \cdot x_2)$$

x_1	x_2	$x_1 + x_2$
0	0	0
0	1	1
1	0	1
1	1	1

$$g(x) = \begin{cases} 0 \text{ for } x < 0 \\ 1 \text{ for } x \geq 0 \end{cases}$$





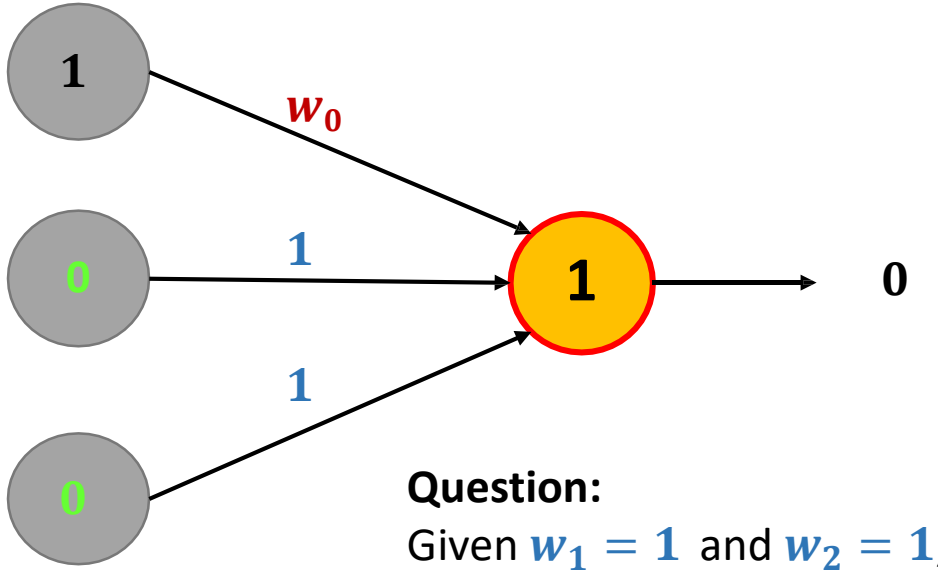
Binary Step Function

Source: [Wikipedia Activation Functions]

The AND Function

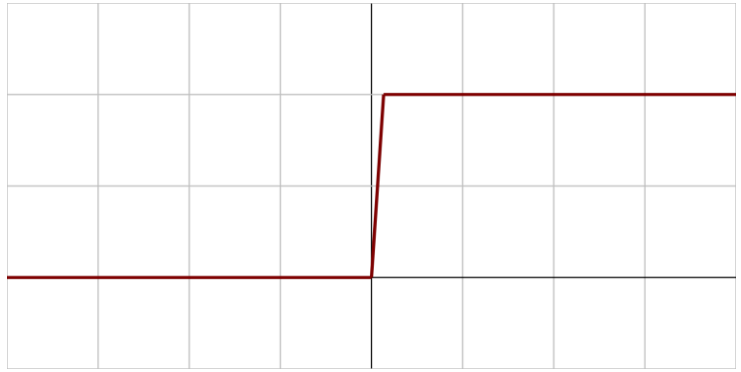
$$AND(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ and } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$h(x_1, x_2) = g(w_0 + 1 \cdot x_1 + 1 \cdot x_2)$$



x_1	x_2	$x_1 \cdot x_1$
0	0	0
0	1	0
1	0	0
1	1	1

$$g(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



Binary Step Function

Question:
Given $w_1 = 1$ and $w_2 = 1$, what value of w_0 do you think our neural network should learn in order to solve the AND function?

Source: Wikipedia Activation Functions

The AND Function

$$AND(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ and } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

$$h(x_1, x_2) = g(w_0 + 1 \cdot x_1 + 1 \cdot x_2)$$

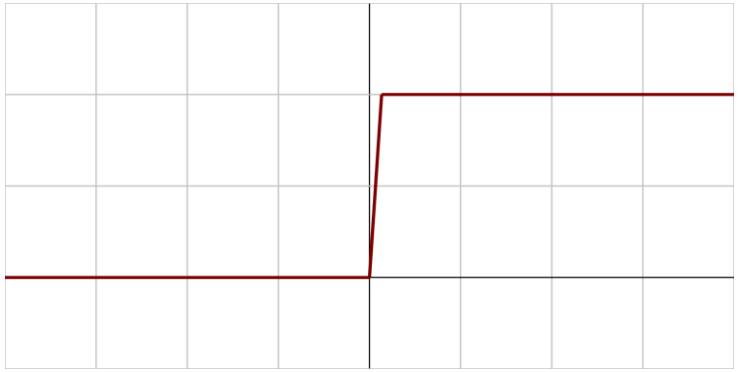
Question:
Given $\boldsymbol{w_1 = 1}$ and $\boldsymbol{w_2 = 1}$, what value of $\boldsymbol{w_0}$ do you think our neural network should learn in order to solve the AND function?

$$\begin{Bmatrix} x \geq 0 & \text{if} & x_1 = 1 \text{ and } x_2 = 1 \\ x < 0 & & \text{otherwise} \end{Bmatrix}$$

$$\begin{Bmatrix} w_0 + 2 \geq 0 \\ w_0 + 1 < 0 \\ w_0 < 0 \end{Bmatrix} \Rightarrow w_0 = -2$$

x_1	x_2	$x_1 \cdot x_1$
0	0	0
0	1	0
1	0	0
1	1	1

$$g(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



Binary Step Function

Source: Wikipedia Activation Functions

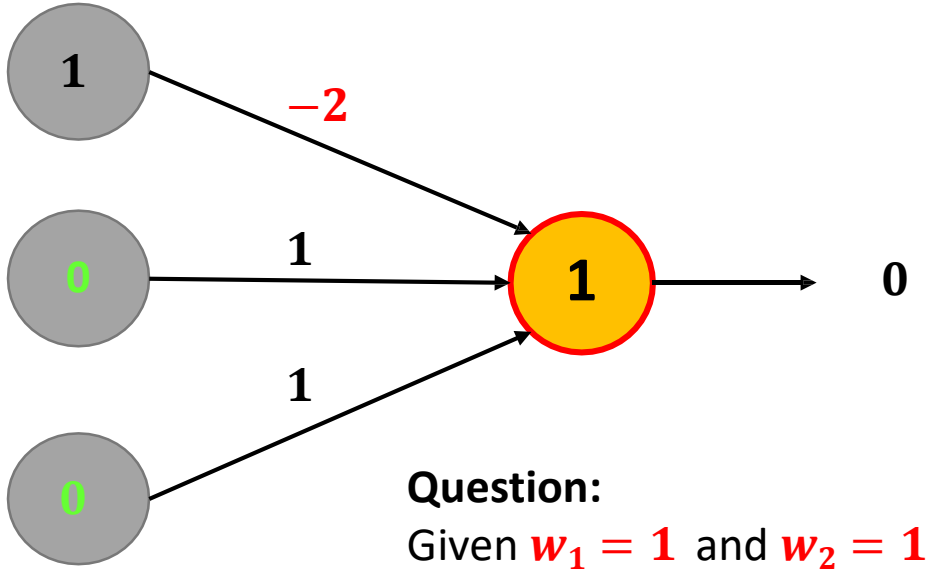
The AND Function

$$AND(x_1, x_2) = \begin{cases} 1 & \text{if } x_1 = 1 \text{ and } x_2 = 1 \\ 0 & \text{otherwise} \end{cases}$$

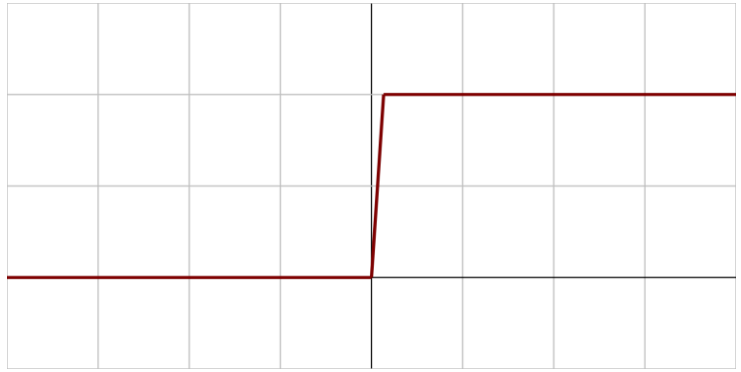
$$h(x_1, x_2) = g(-2 + 1 \cdot x_1 + 1 \cdot x_2)$$

x_1	x_2	$x_1 \cdot x_1$
0	0	0
0	1	0
1	0	0
1	1	1

$$g(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$$



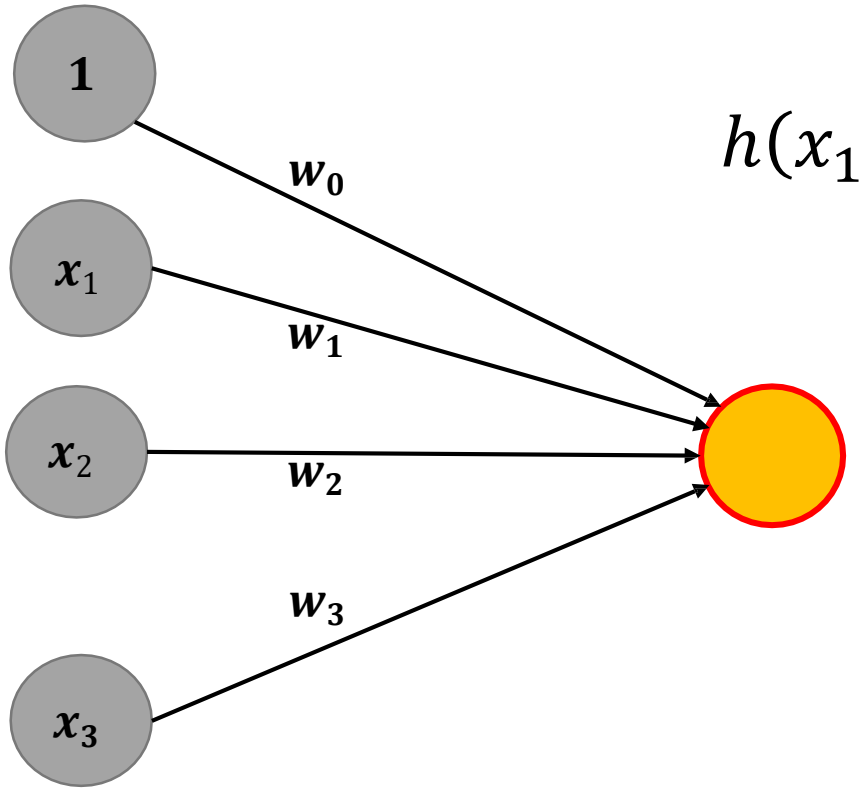
Question:
Given $w_1 = 1$ and $w_2 = 1$ What value of w_0 do you think our neural network should learn in order to solve the AND function?



Binary Step Function

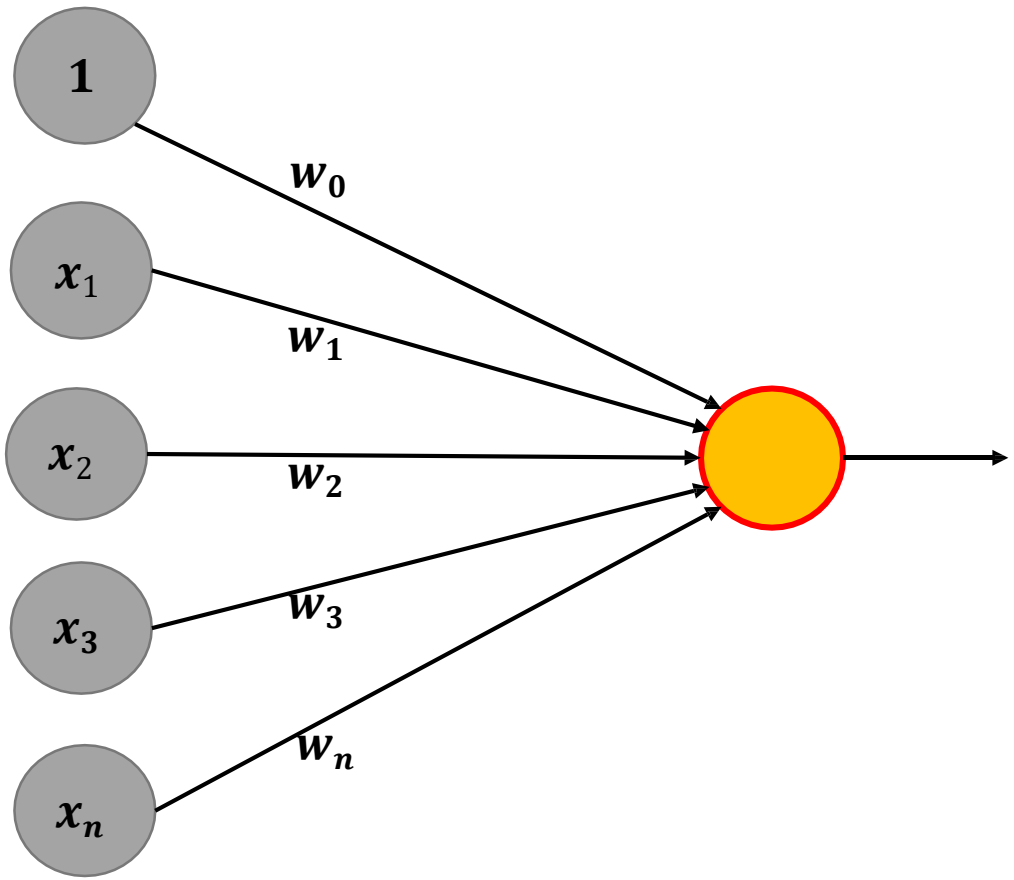
Source: [Wikipedia Activation Functions]

Generalising – 3 inputs



$$h(x_1, x_2, x_3) = g(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3)$$

Generalising – n inputs



$$h(x_1, \dots, x_n) = g\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

Training the Neural Network

Training the Network

- Involves learning how to adjust the weights

Aim

- The network learns this by choosing different weights to minimise the loss or errors.

Loss Function

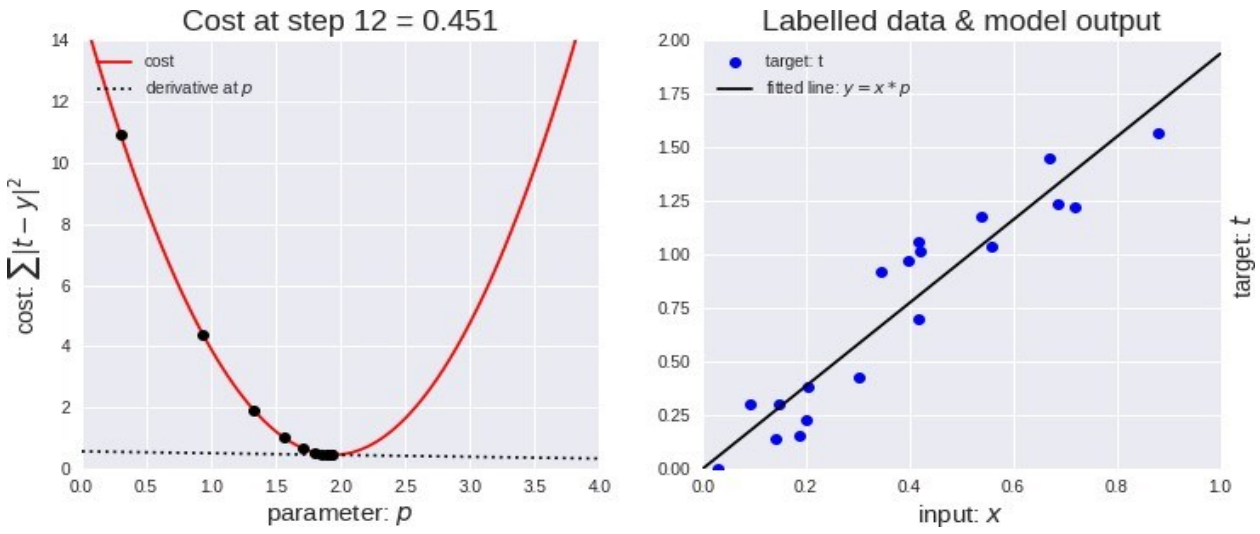
- The errors are determined by some loss function that compares its output with the true output

Solution Technique

- The weights are adjusted using a technique called the gradient descent
- An algorithm for minimizing the loss when training a neural network

Gradient Descent

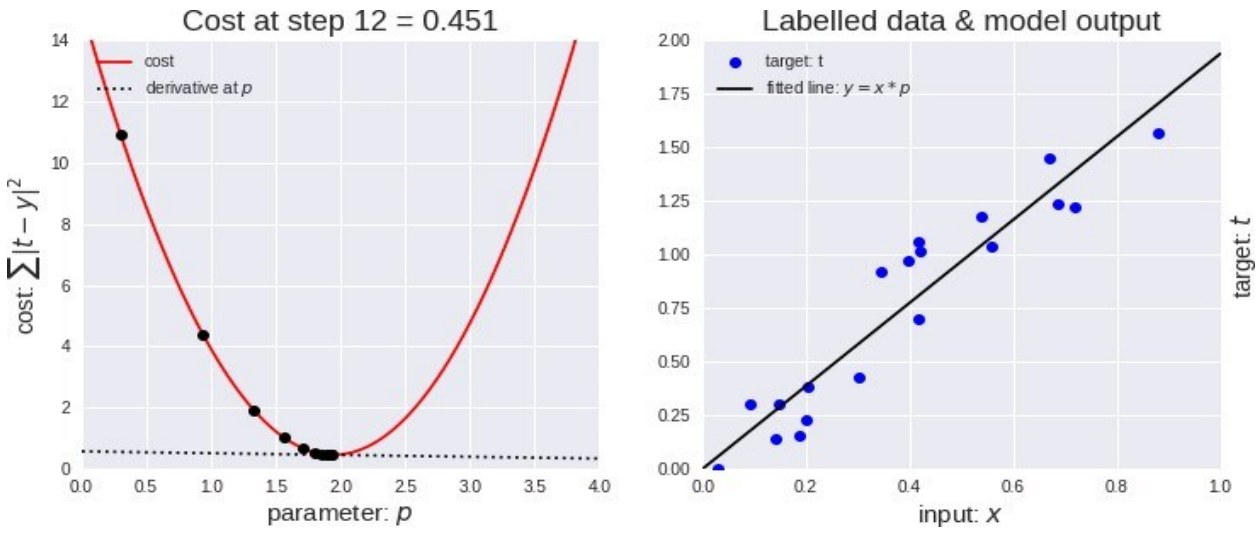
- Start with a random set of weights
- Repeat
  - Calculate the gradient based on all data points direction that decreases the loss
  - Update the weights according to the gradient



Gradient Descent

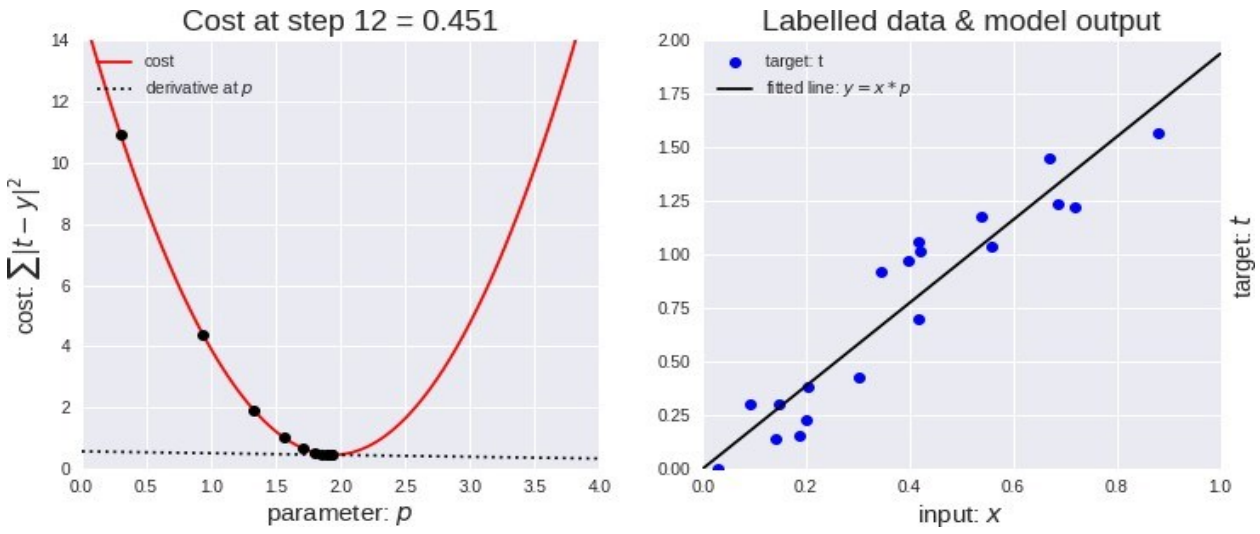
- Start with a random set of weights
- Repeat
  - Calculate the gradient based on **all data points** direction that decreases the loss
  - Update the weights according to the gradient

This can be computationally expensive!



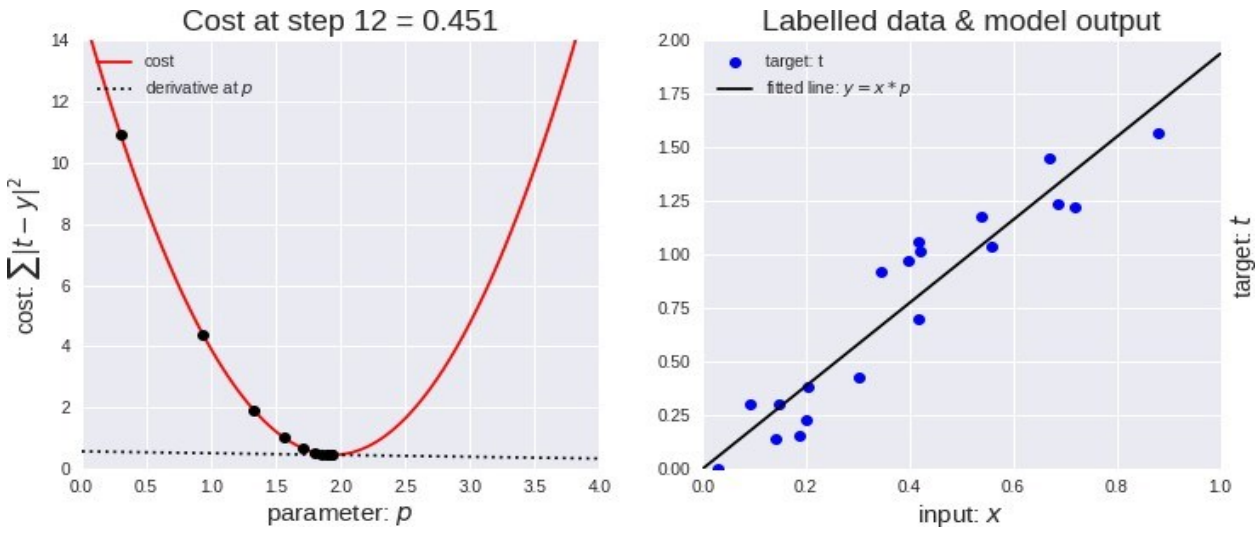
Stochastic Gradient Descent

- Start with a random set of weights
- Repeat
  - Calculate the gradient based on **one data point** direction that decreases the loss
  - Update the weights according to the gradient



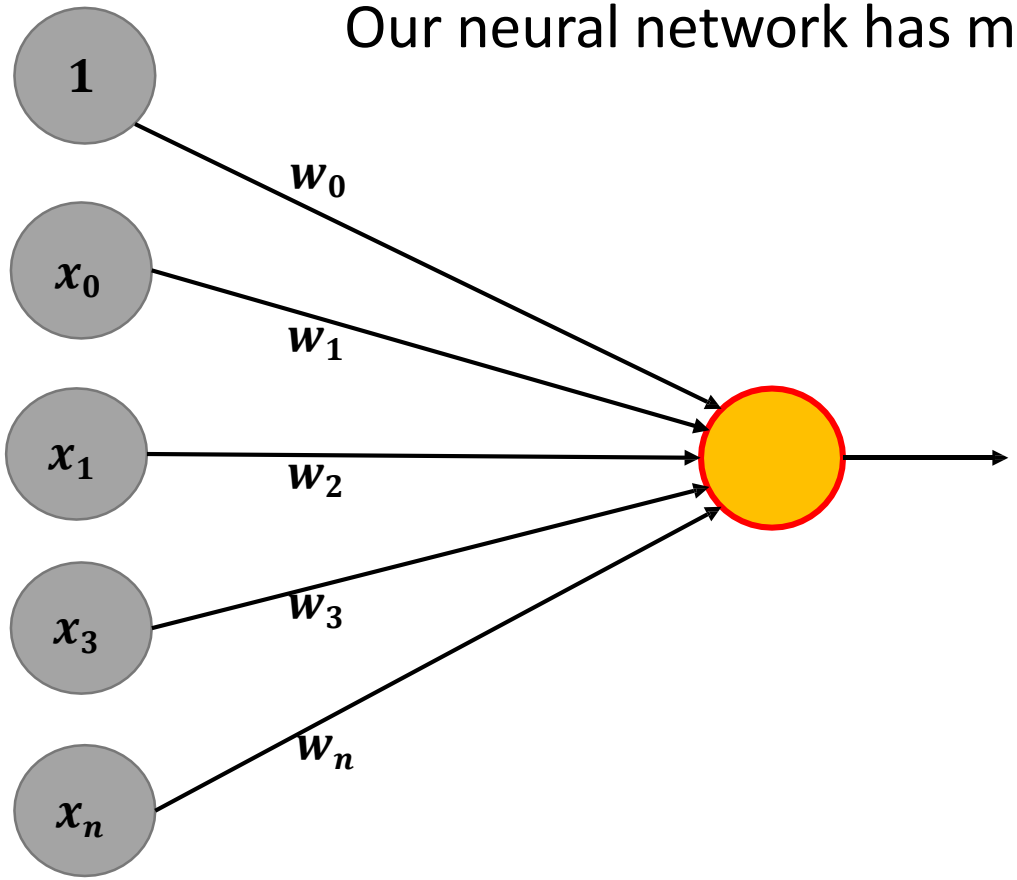
Mini-Batch Gradient Descent

- Start with a random set of weights
- Repeat
  - Calculate the gradient based on **one small batch data point** direction that decreases the loss
  - Update the weights according to the gradient



So far…

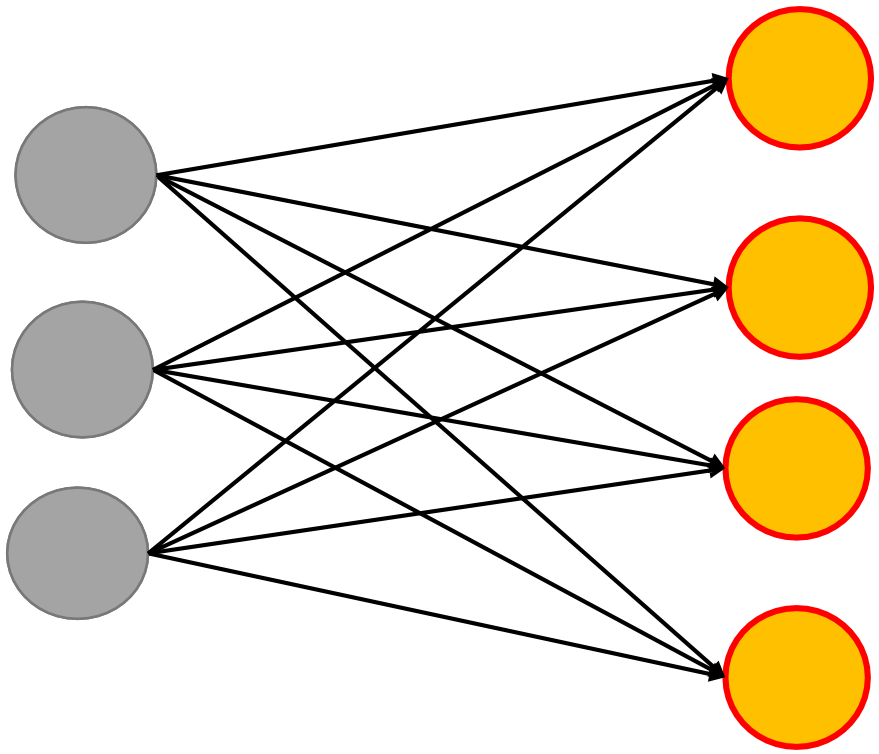
Our neural network has multiple inputs and a single output



$$h(x_1, \dots, x_n) = g\left(w_0 + \sum_{i=1}^{n} w_i x_i\right)$$

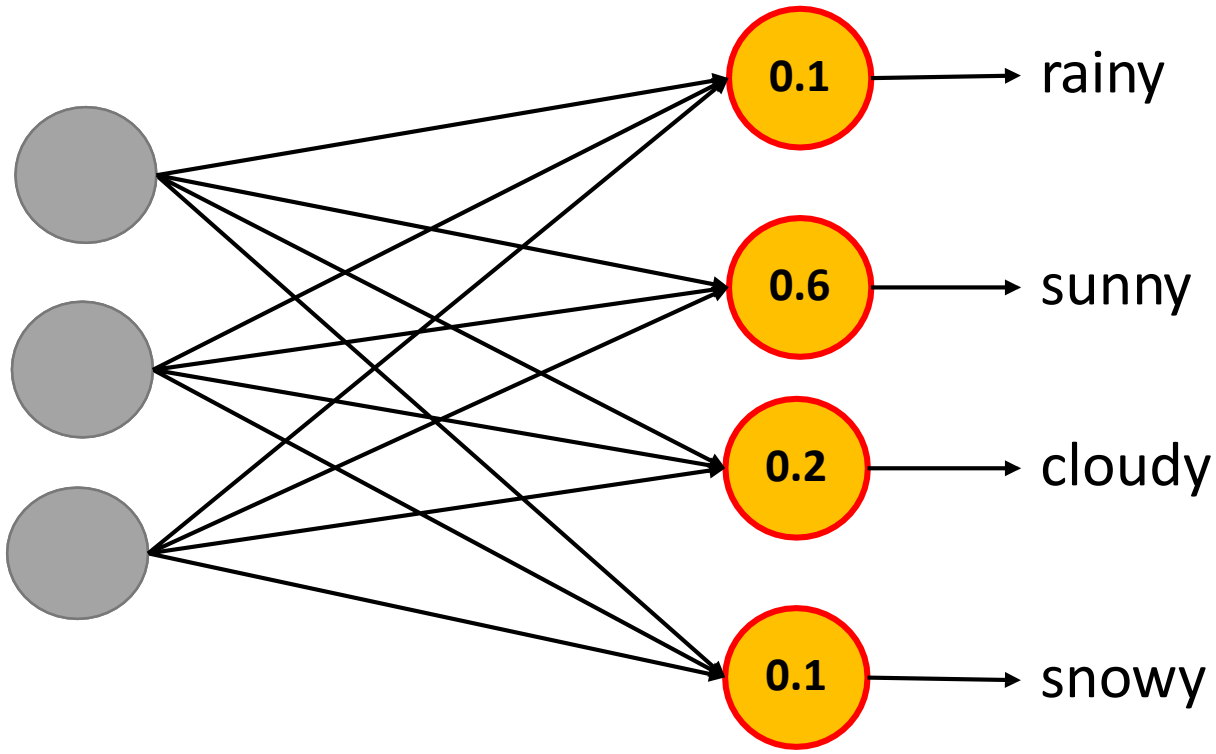
But we can also have…

…multiple outputs



But we can also have…

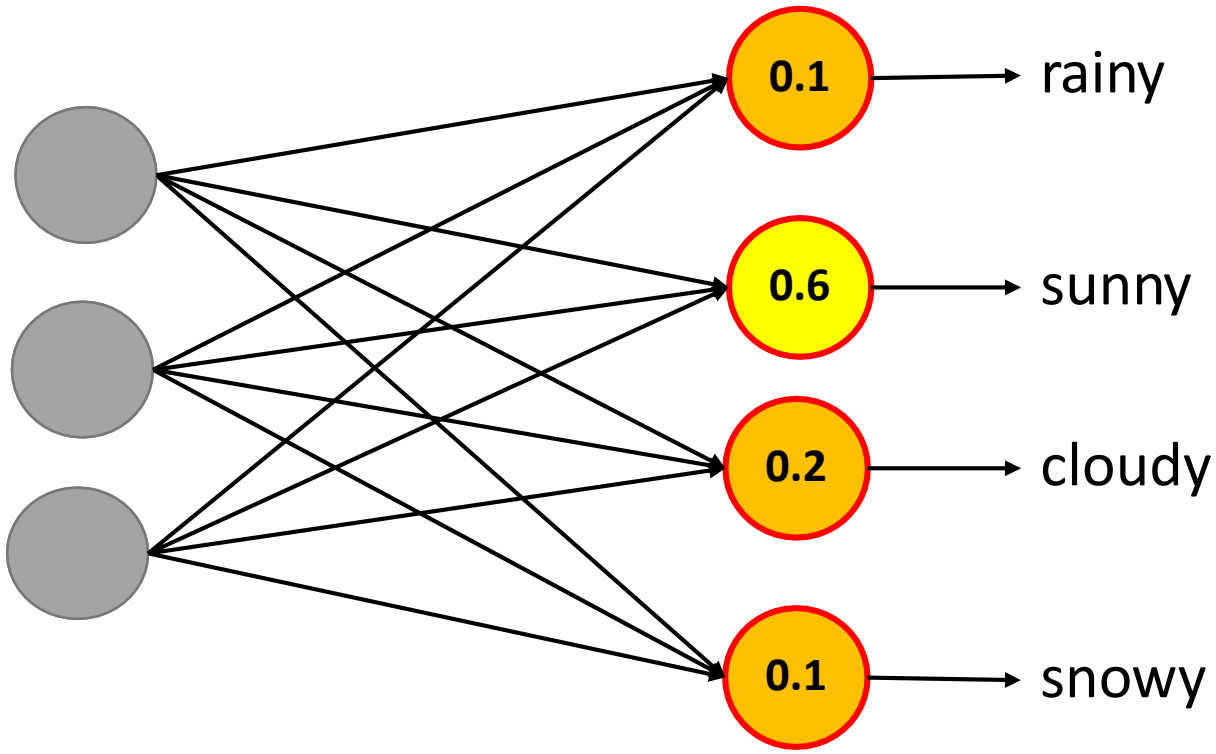
…multiple outputs



These could be a probability distribution of possible outcomes

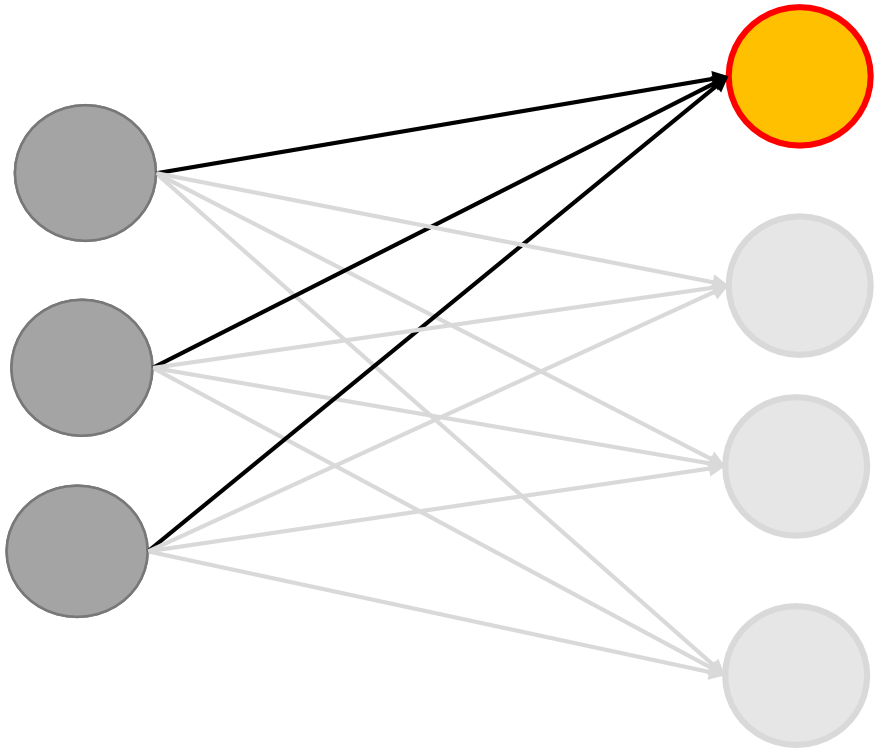
But we can also have…

…multiple outputs



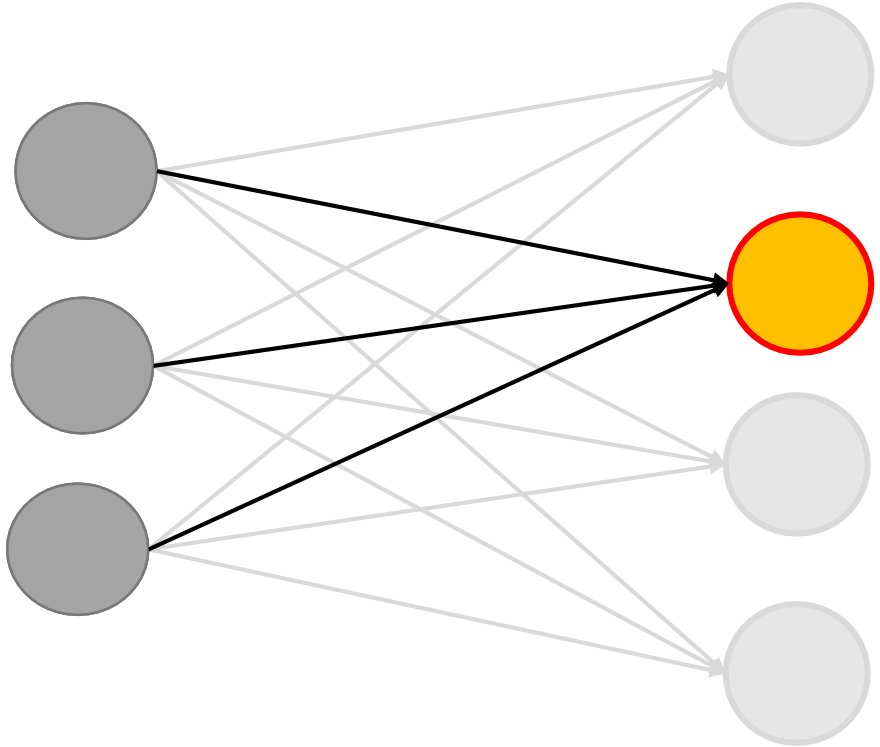
These could be a probability distribution of possible outcomes

Multiple Output Networks



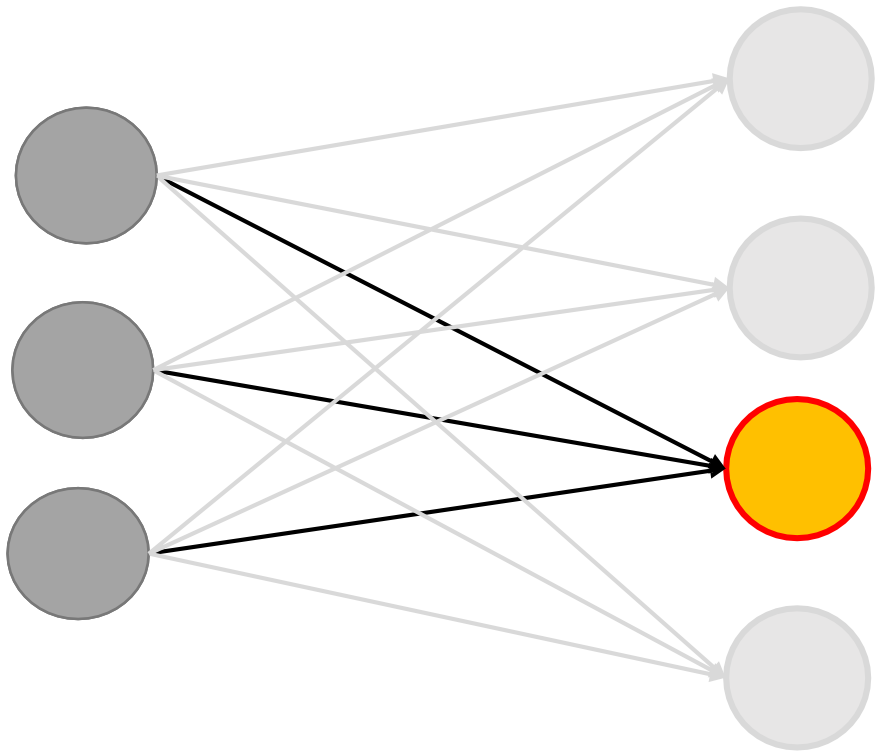
Also we can think of multiple output networks as **different networks** with the same inputs whose outputs can be converted to some form of probability distribution

Multiple Output Networks



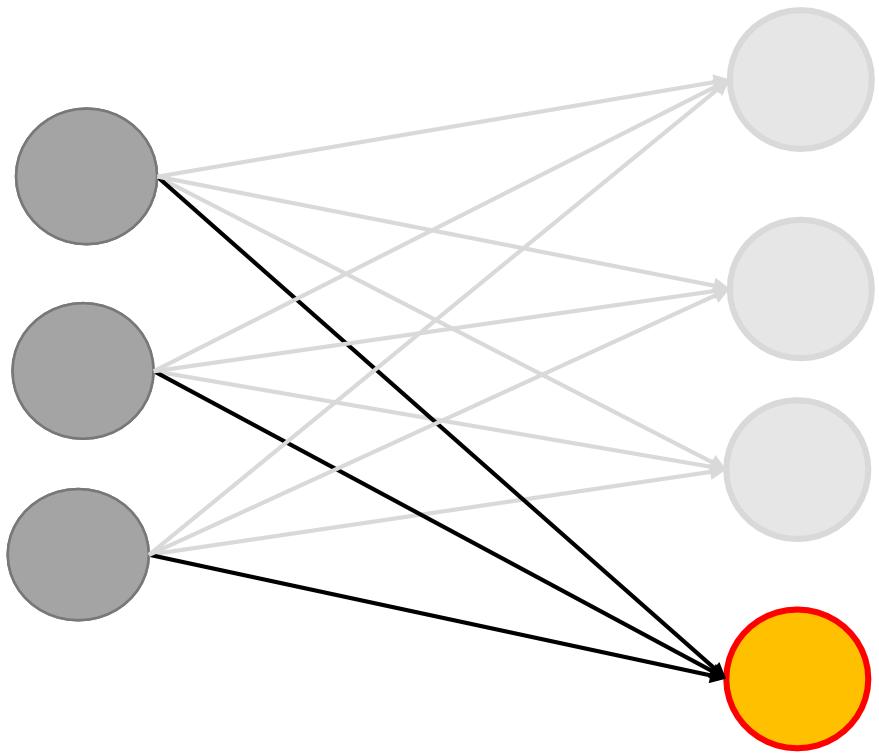
Also we can think of multiple output networks as **different networks** with the same inputs whose outputs can be converted to some form of probability distribution

Multiple Output Networks



Also we can think of multiple output networks as **different networks** with the same inputs whose outputs can be converted to some form of probability distribution

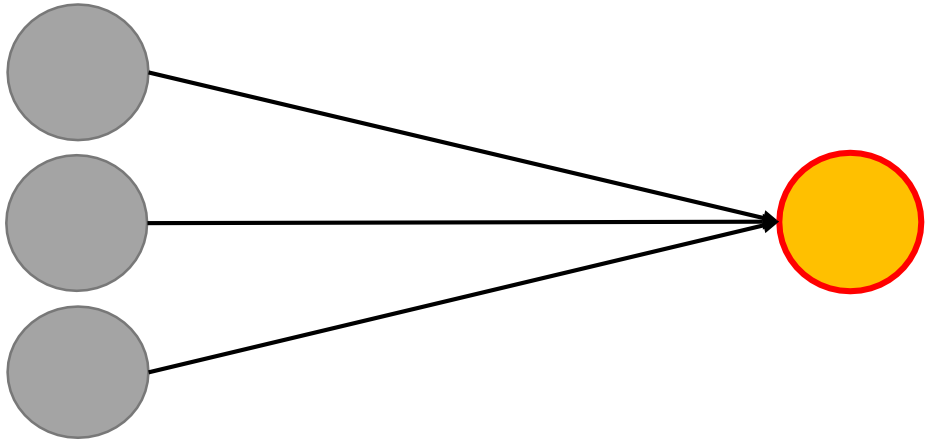
Multiple Output Networks

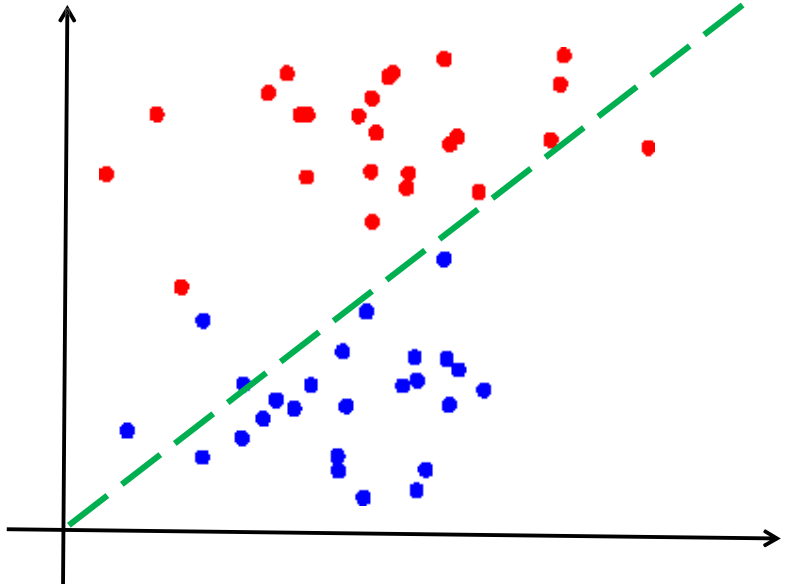


Also we can think of multiple output networks as **different networks** with the same inputs whose outputs can be converted to some form of probability distribution

The Perceptron

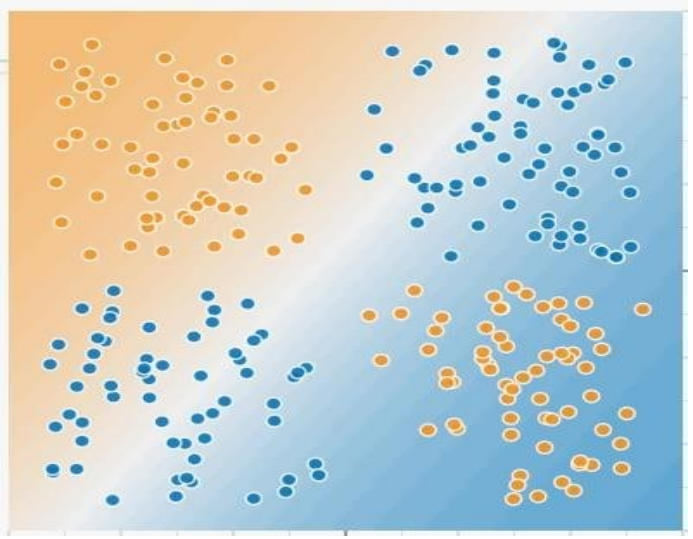
- The networks we have built so far are the simplest forms of neural networks
- They are generally referred to as **the perceptron**
- They have only a set of inputs and a set of outputs and the connections between them
- They are only capable of learning **linearly separable** decision boundaries

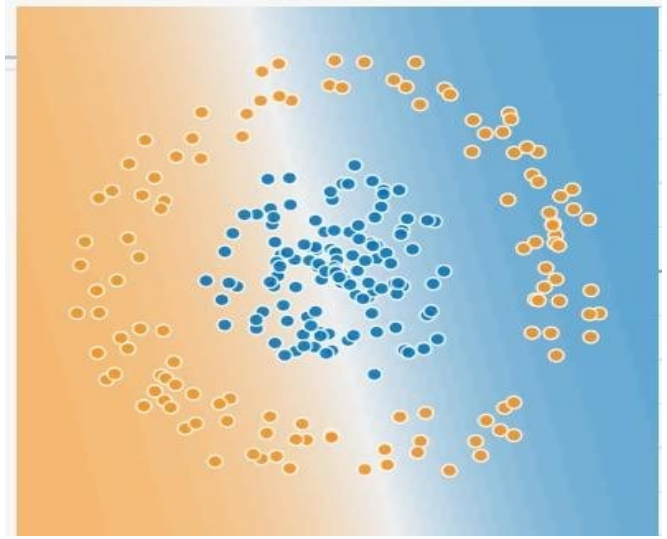


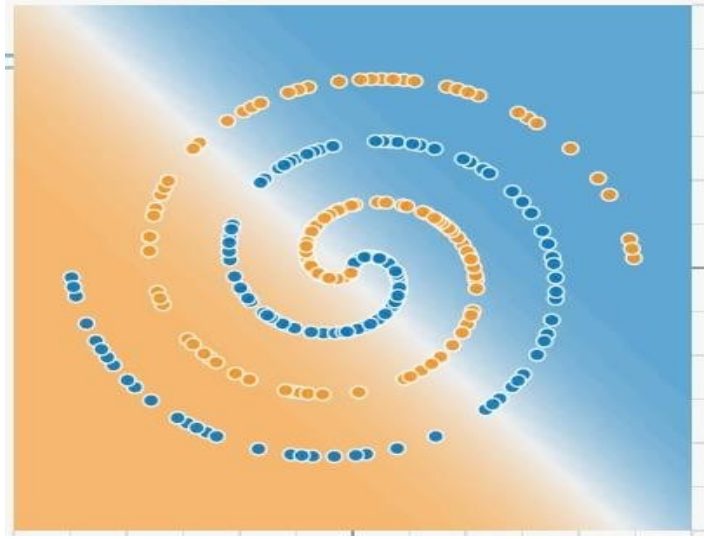


Complex Decision Boundaries

- Some example of complex decision boundaries

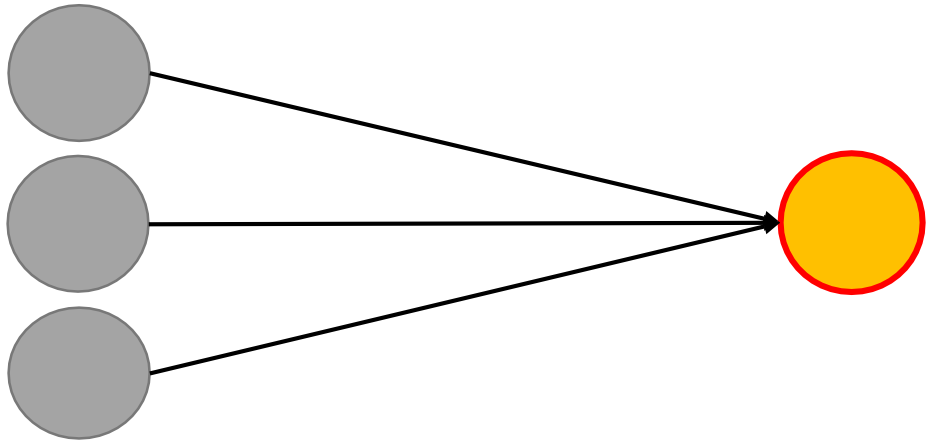






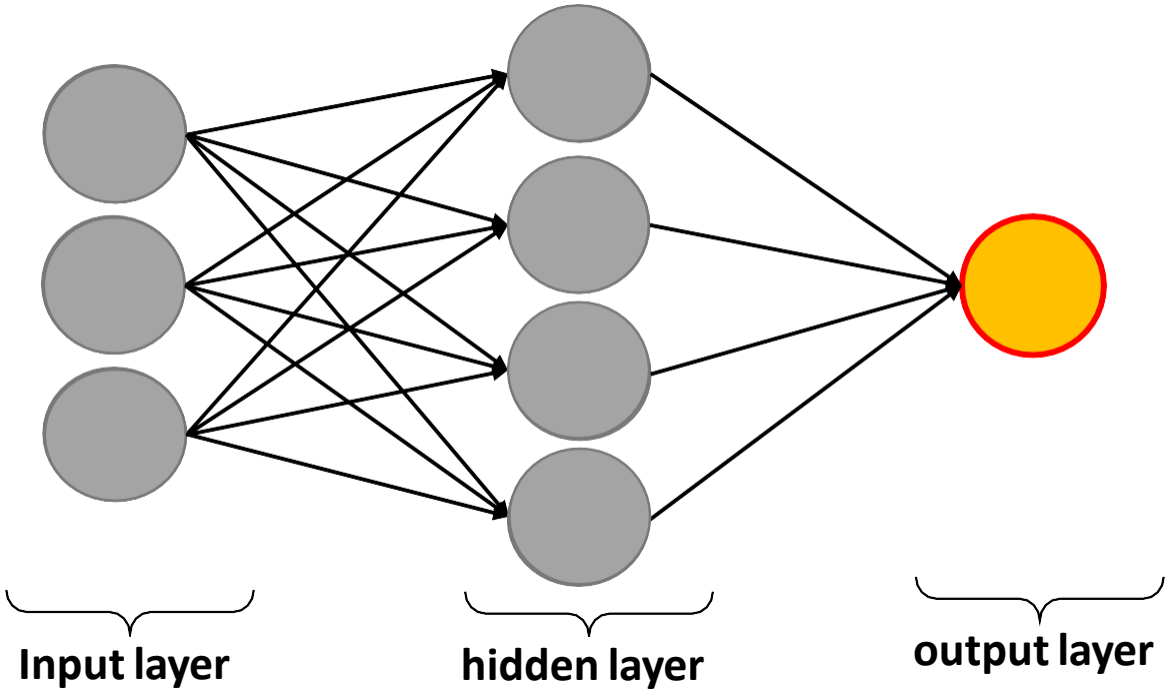
Multilayer Neural Networks

- These are artificial neural networks with an input layer, an output layer and one or more **hidden layers**



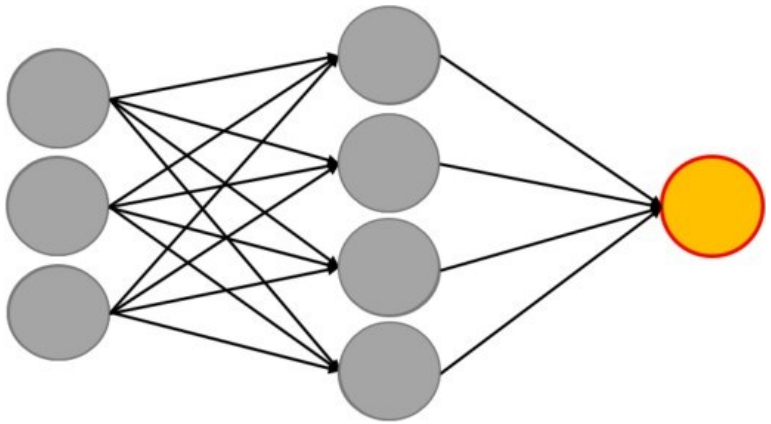
Multilayer Neural Networks

- These are artificial neural networks with an input layer, an output layer and one or more **hidden layers**



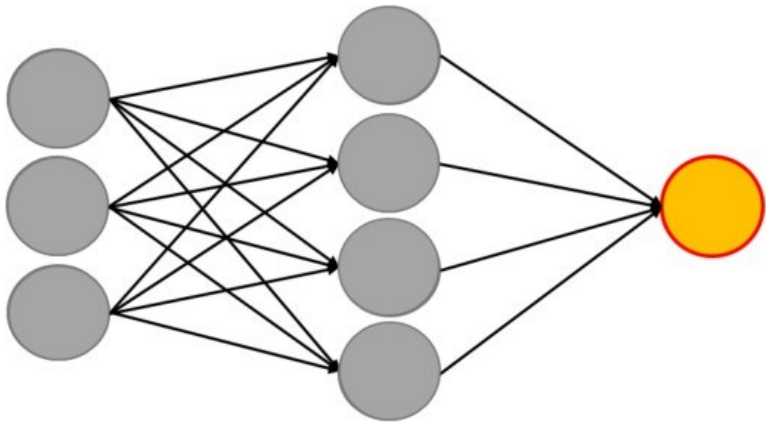
Backpropagation

- We have only the input and output layer values.
- We need a way to learn the correct hidden layer values so that we can adjust the weights appropriately
- The **backpropagation** algorithm enables us to propagate the errors from output layer through the hidden layers to adjust the weights



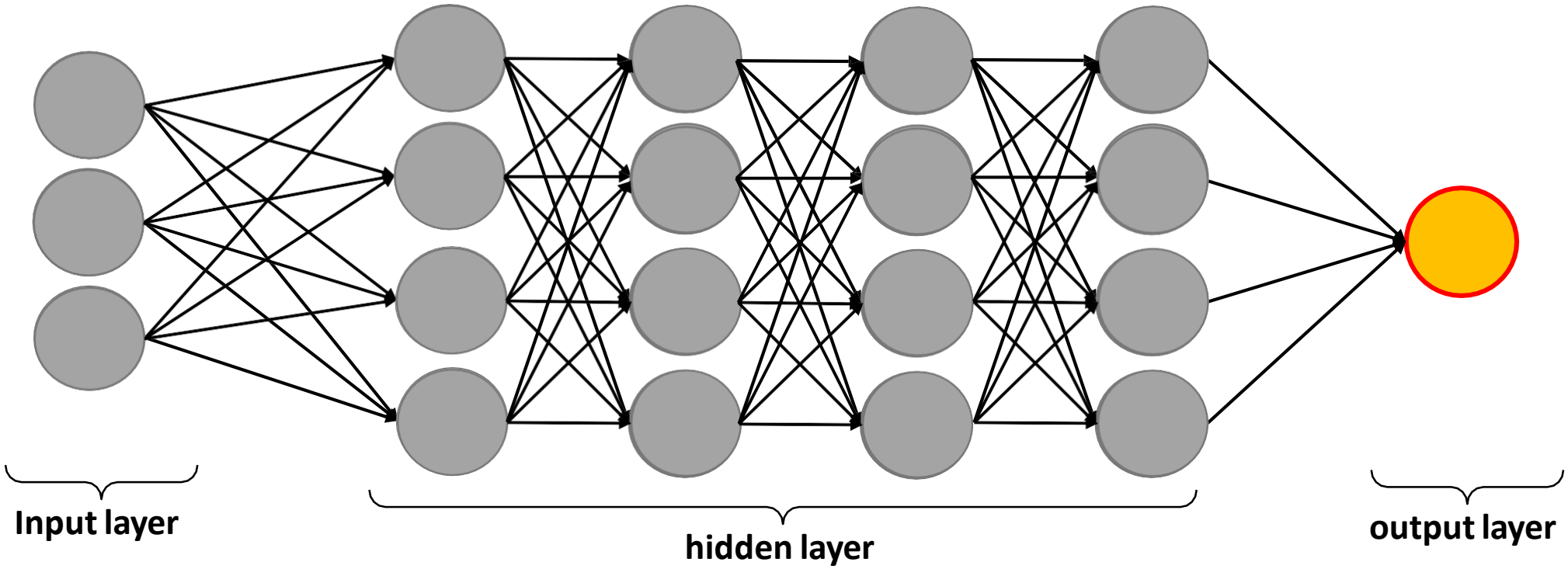
Backpropagation Algorithm

- Start with a random choice of weights
- Repeat:
  - Calculate error for the output layer
  - For each layer, starting with the output layer and moving inwards towards earliest hidden layers
    - Propagate the error back one layer
    - Update the weights



Deep Neural Networks

- Multilayer neural networks with backpropagation is the backbone of deep neural networks
- We can introduce more layers to enable the network learn higher level features



Overfitting

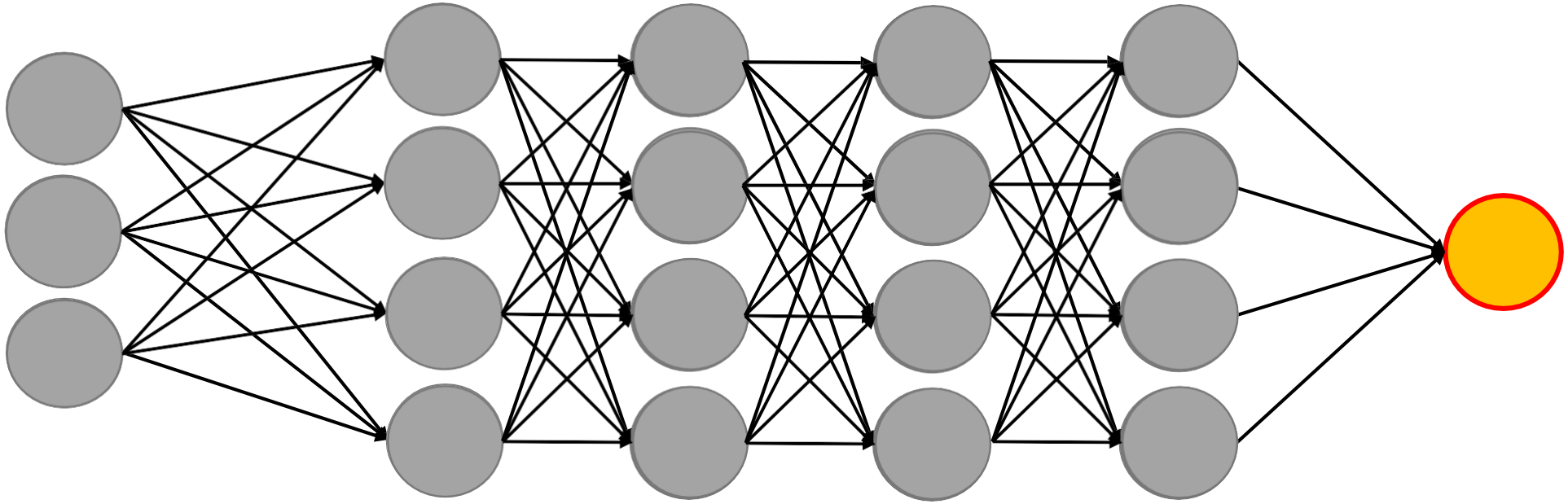
- As the network becomes more complex, there is a higher tendency of **overfitting**
  - i.e. learning so much about the training data that the model produce does not generalise very well
- Cross validation helps to minimise overfitting but there are techniques that can help neural networks reduce that
- One of such techniques is **dropout**

Dropout

- Dropout is temporarily removing nodes – selected at random – from a neural network
- The goal is to prevent over-reliance on certain nodes

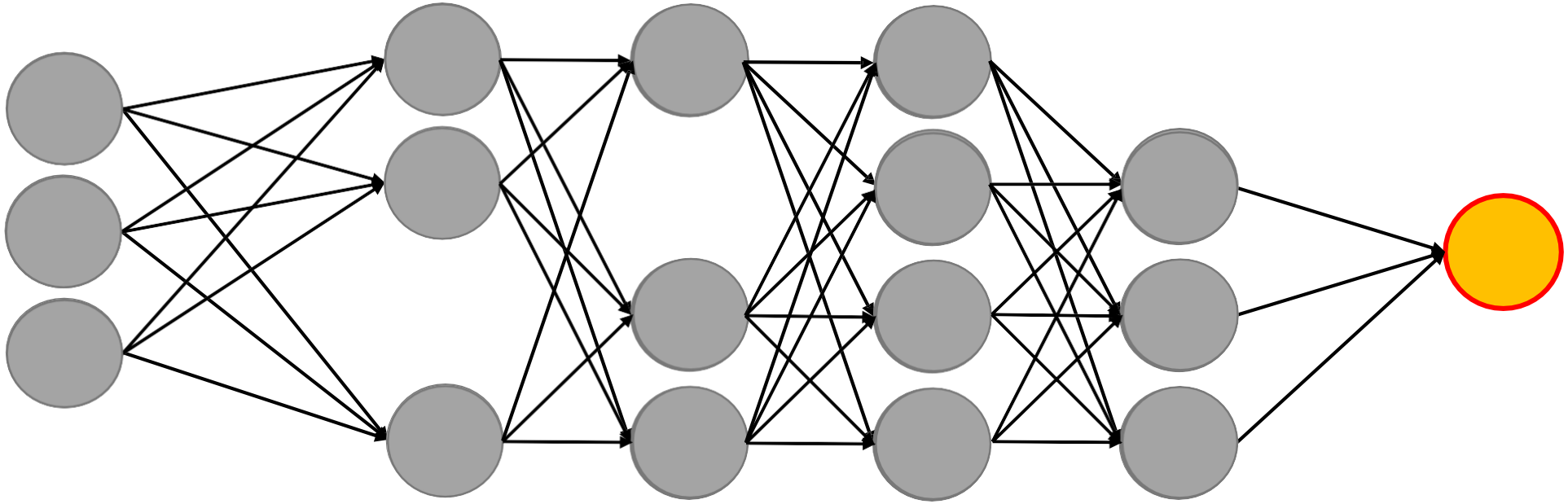
Dropout

- Dropout is temporarily removing nodes – selected at random – from a neural network
- So, given this **fully connected** network:



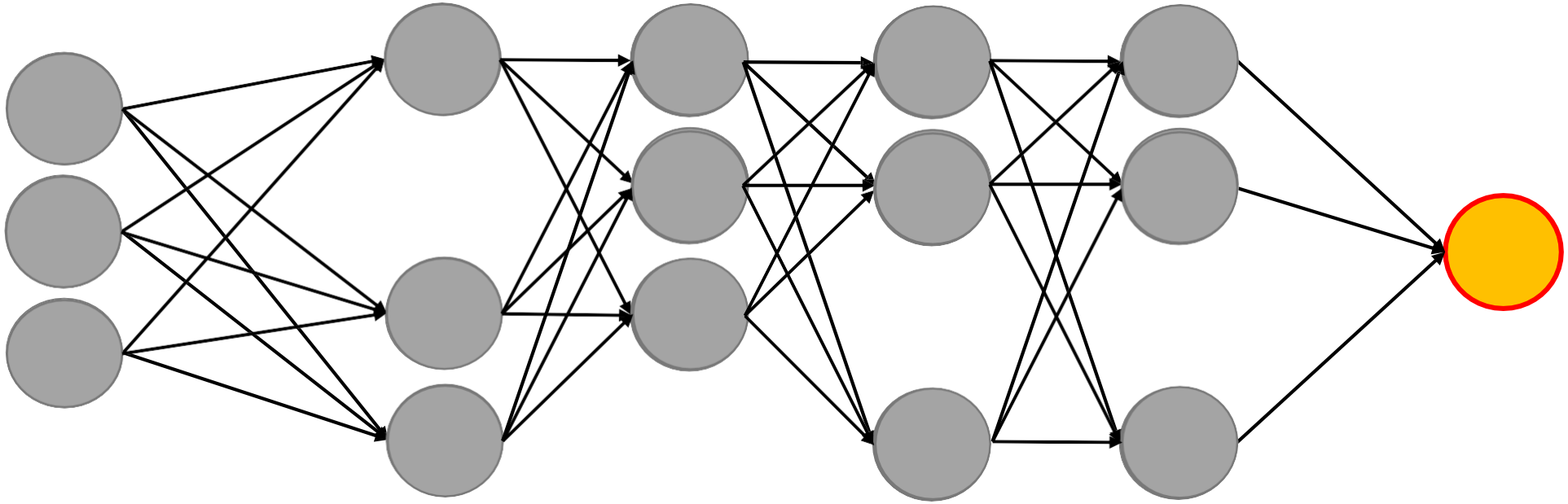
Dropout

- Dropout is temporarily removing nodes – selected at random – from a neural network
- So, given this **fully connected** network:



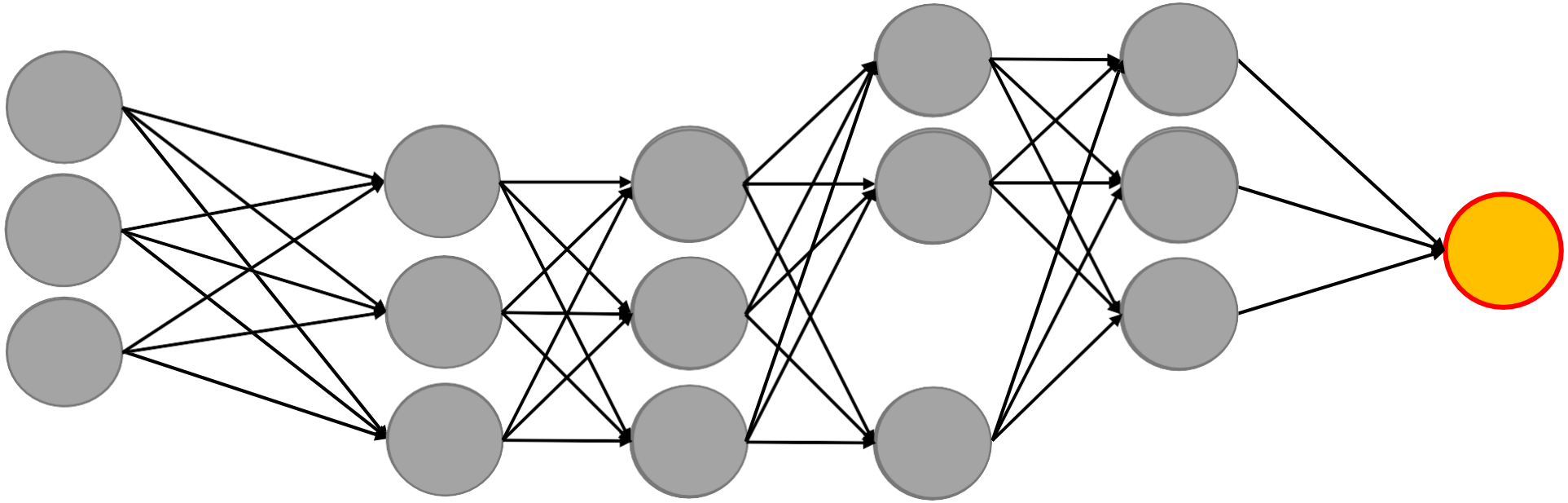
Dropout

- Dropout is temporarily removing nodes – selected at random – from a neural network
- So, given this **fully connected** network:



Dropout

- Dropout is temporarily removing nodes – selected at random – from a neural network
- So, given this **fully connected** network:



Convolutional Neural Networks

- CNNs are usually applied to computer vision tasks i.e. building models that can 'read and understand' images
- Application area includes
  - Image and video recognition
  - Medical image analysis
  - Recommender systems
  - etc
- Key components include:
  - Image convolution
  - Pooling (max pooling)

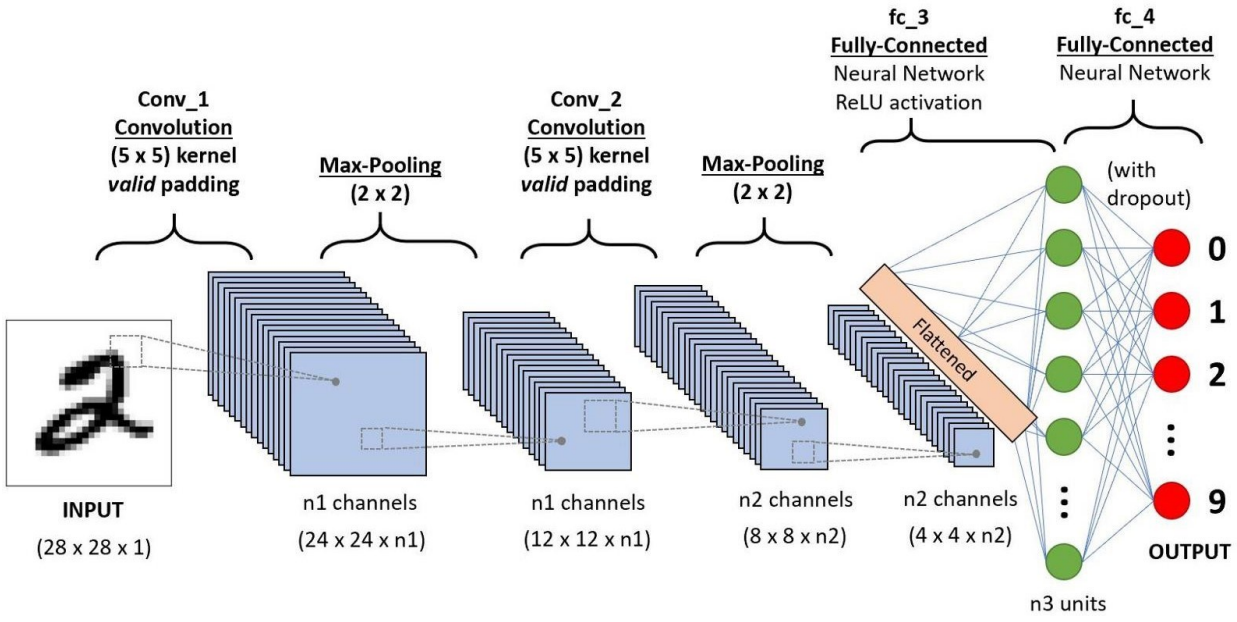


Image Convolution

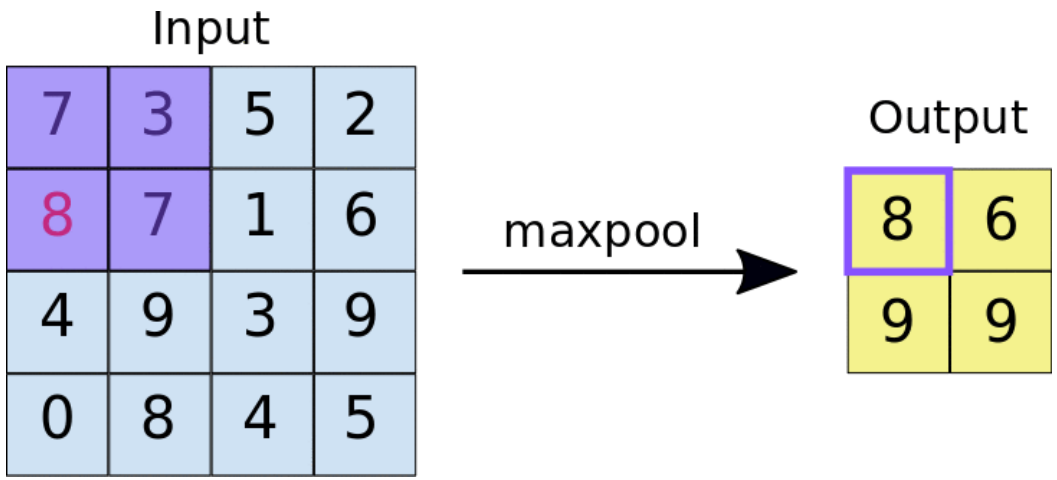
- This process involves applying a filter that adds each pixel value of an image to its neighbours, weighted according to a **kernel matrix**
- The goal is to extract **only** interesting features of the image for better model training

$$\begin{bmatrix} 7 & 2 & 3 & 3 & 8 \\ 4 & 5 & 3 & 8 & 4 \\ 3 & 3 & 2 & 8 & 4 \\ 2 & 8 & 7 & 2 & 7 \\ 5 & 4 & 4 & 5 & 4 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 6 & & \\ & & \\ & & \end{bmatrix}$$

7x1+4x1+3x1+
2x0+5x0+3x0+
3x-1+3x-1+2x-1
= 6

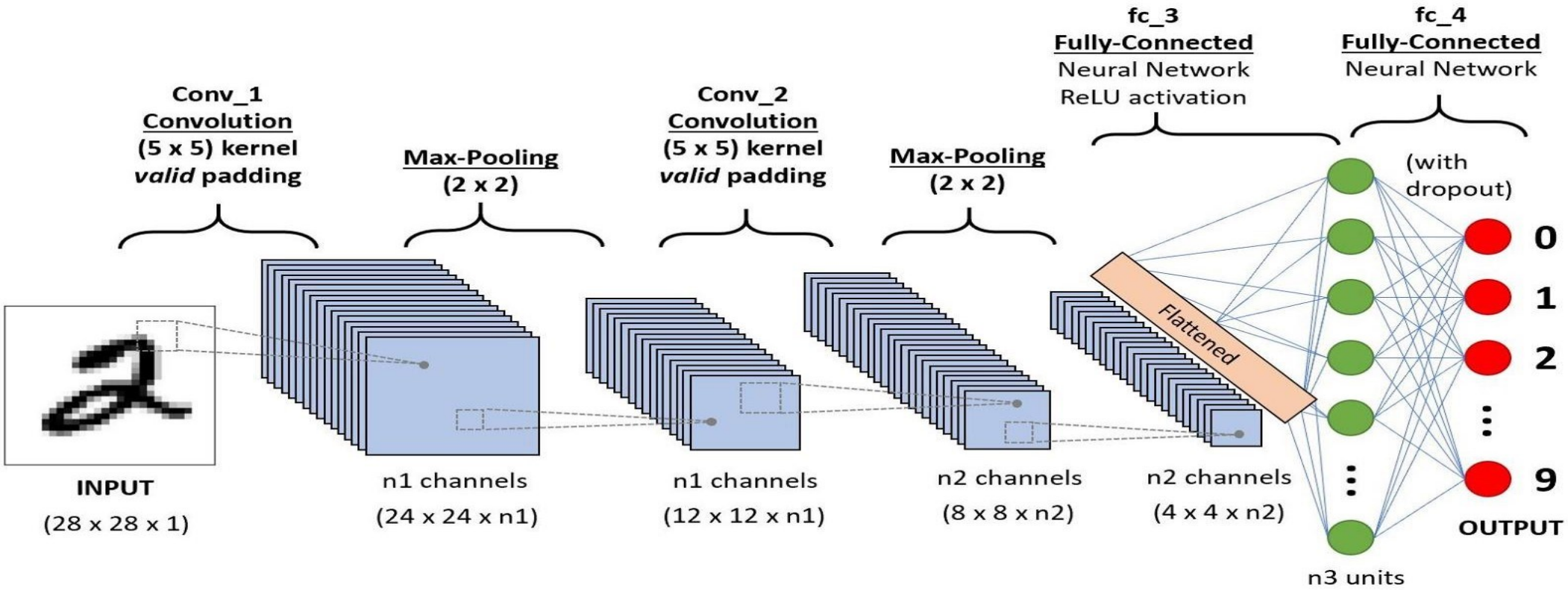
Pooling

- Pooling is a technique that helps to **reduce the size** of an input by sampling from regions in the input
- One commonly applied type of pooling is **max-pooling**
- Max-pooling chooses the maximum value in the region
- We can also do **average pooling**



CNN Design

- In the CNN design, you can apply these two key steps **image convolution** and **max-pooling**

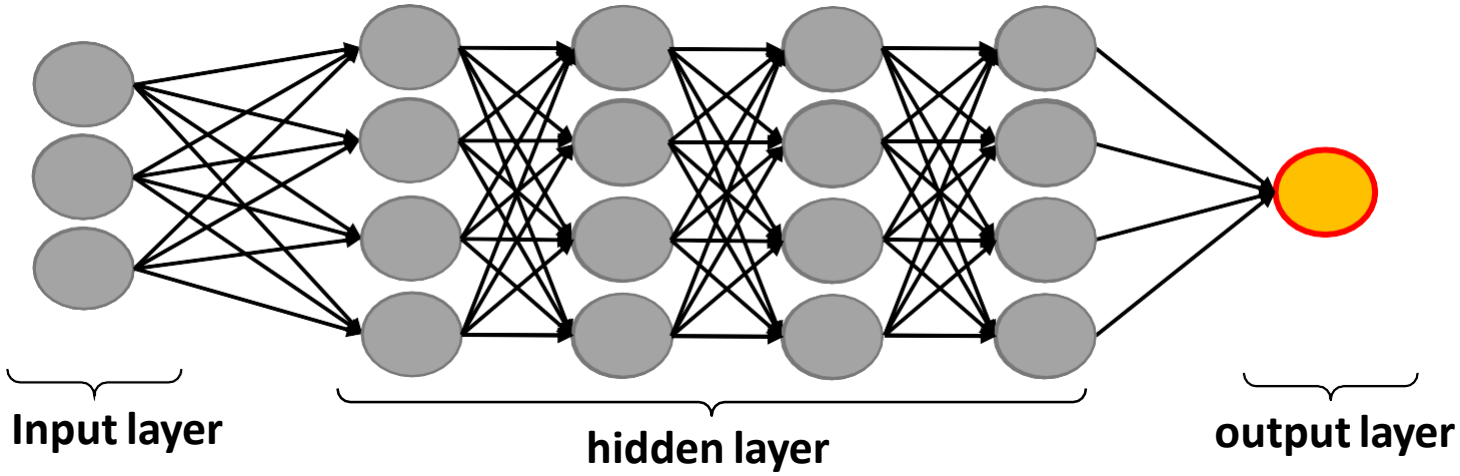


Recurrent Neural Networks

- So far, we have seen typical **feedforward neural networks**
- Feedforward networks simply pass data through the different nodes until it reaches the output node.
- Some times, we need to produce a sequence of values e.g.
  - machine translation:
    - A sequence of words in one language produces another sequence of words in
  - video subtitling
    - A sequence of image frames can produce a sequence of words
  - Image captioning
    - An image produced a sequence of words

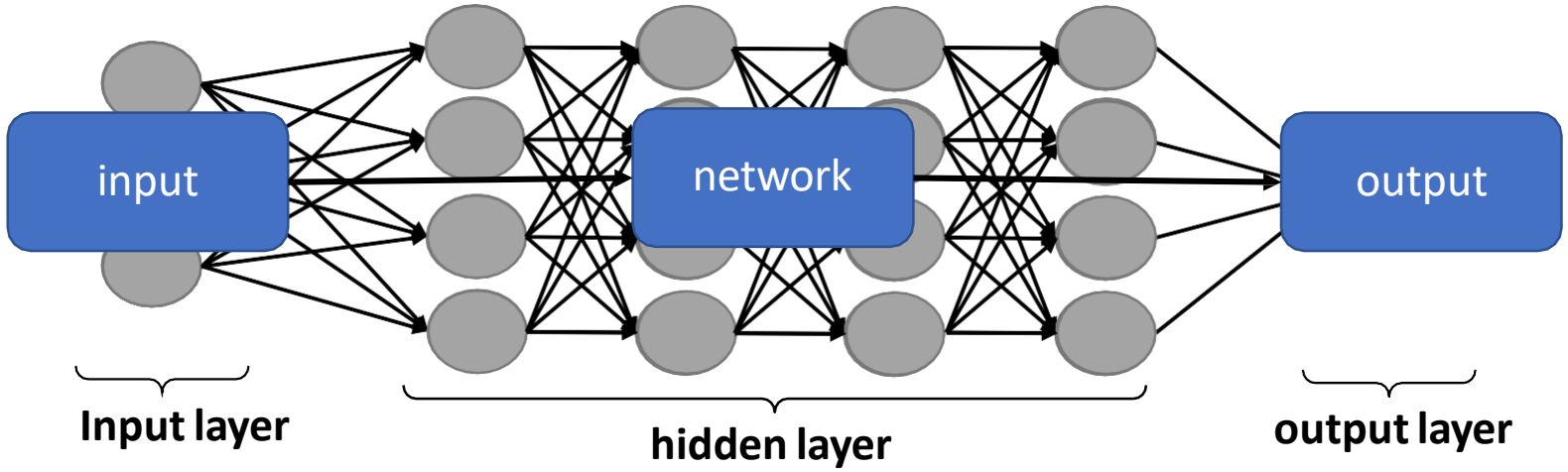
Recurrent Neural Networks

- RNNs model sequence-to-sequence learning
- It achieves this by **feeding back** its output an input to the network thereby remembering something about at each run.



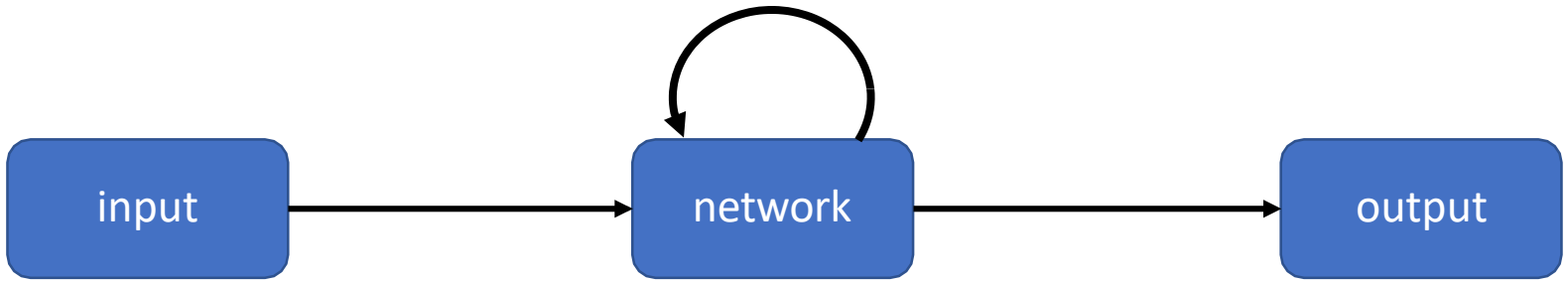
Recurrent Neural Networks

- RNNs model sequence-to-sequence learning
- It achieves this by **feeding back** its output an input to the network thereby remembering something about at each run.



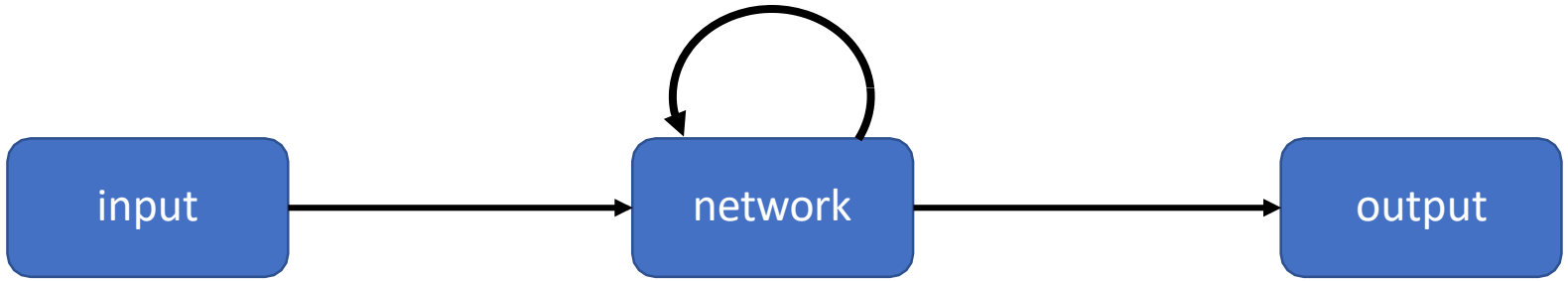
Recurrent Neural Networks

- RNNs model sequence-to-sequence learning
- It achieves this by **feeding back** its output an input to the network thereby remembering something about at each run.



Recurrent Neural Networks

- RNNs model sequence-to-sequence learning
- It achieves this by **feeding back** its output an input to the network thereby remembering something about at each run.
- A common variant of the RNN is the LSTM
- LSTM - Long-Short Term Memory helps the model to remember relevant past history for better learning



Coming up…

Lab: SVM and Artificial Neural Networks

- Materials on Moodle

Group	Day	Time	Room
SCC361/P01/01	Wednesday	11:00-13:00	FST B070
SCC361/P01/02	Thursday	16:00-18:00	FST B074
SCC361/P01/03	Wednesday	16:00-18:00	FST B070
SCC361/P01/04	Thursday	14:00-16:00	FST B070
SCC361/P01/05	Friday	14:00-16:00	FST B076

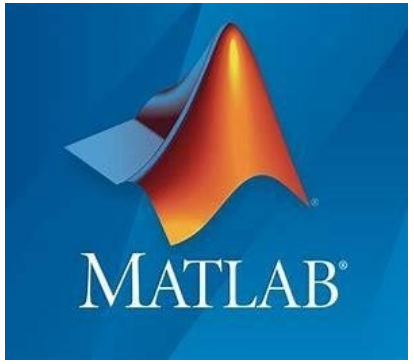
Week 6 Lectures: Genetic Algorithms

- Concept of Search Algorithms

Reminder

Coursework: Task on Moodle

- 20% of Overall Mark
- Feature Extraction and Comparison of ML approaches
- Report Template on Moodle
- Due Friday 18th November 2022 @ 4pm (Week 6)
- Submission on Moodle
- Marks in 3 weeks (Week 9)



Coursework Evaluation Results

Method	Sentiment Analysis	Traffic Sign Labelling
Discriminant Analysis	50.29	98.55
Decision Tree	43.93	86.96
Ensemble	51.50	91.30
KNN	37.63	91.30
Naïve Bayes	28.38	72.46
SVM	51.50	95.65