

# Introduction to Machine Learning

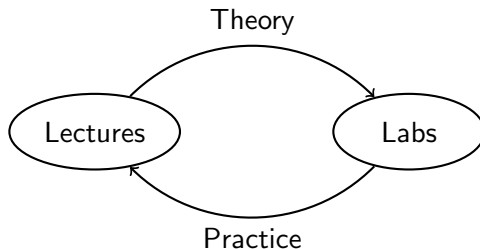
## Sample Lecture and Lab

Computer Science

# Purpose

- ▶ Understand machine learning
  - ▶ Basic concepts
  - ▶ Working with data
  - ▶  $k$  nearest neighbours
  - ▶ Neural networks/multilayer perceptrons
  - ▶ Measuring performance
- ▶ Learn about studying at university

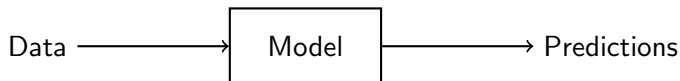
# Session Structure



- ▶ 50% lecture - exploring machine learning
- ▶ 50% lab - applying ML techniques to a dataset

# What is Machine Learning?

Getting computers to find patterns in data **without actually telling it what to look for!**



Consists of feeding some *training data* to a computer to create a *model* which can make *predictions*.

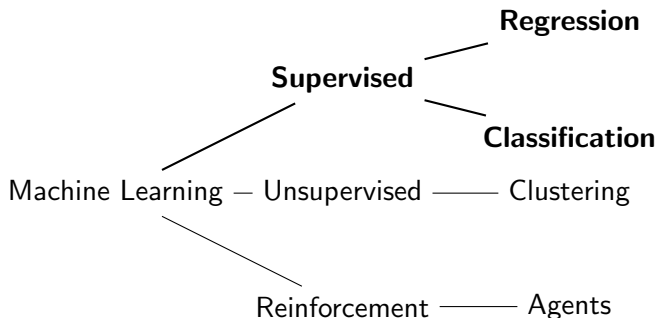
- ▶ Predictions can be on new/unseen data
- ▶ Predictions can be about data itself
- ▶ Predictions can be next action to take

# Why use Machine Learning?



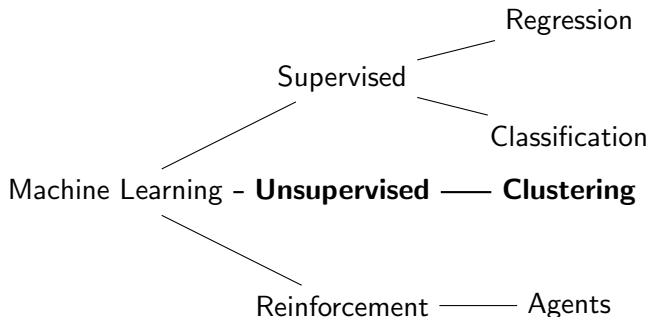
Some problems are difficult to solve using conventional algorithms. For these, it is better to let machines find their own solutions instead of manually specifying how to solve them manually.

# Types of Machine Learning Problems



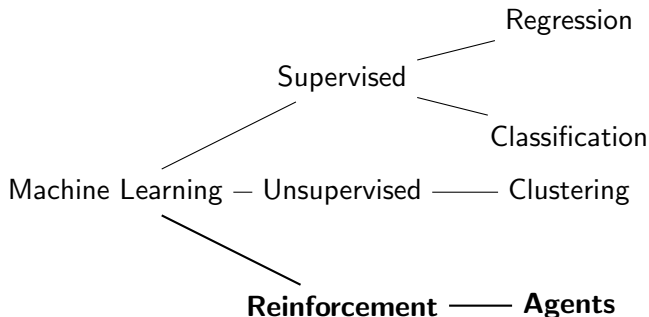
- ▶ Supervised learning: using data with labelled outcomes
  - ▶ Regression: predicting a continuous value
  - ▶ Classification: predicting a discrete value

# Types of Machine Learning Problems



- ▶ Unsupervised learning: using data without labelled outcomes
  - ▶ Clustering: finding clusters of data within dataset to discover patterns

# Types of Machine Learning Problems

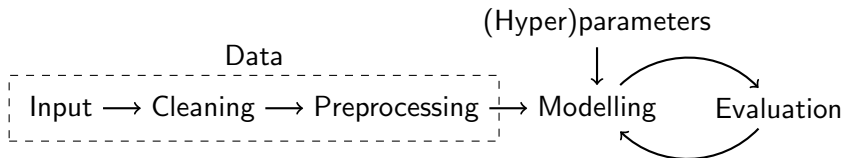


- Reinforcement learning: using feedback to learn a policy
  - Agent based modelling: using state/environment and rewards to decide next action



# Machine Learning Pipeline

Sequential process for creating machine learning models:



Significant portion dedicated to working with data – machine learning techniques rely on **high quality data**!

# Working with Data - Input

Multiple *types* of data:

Type	Description	Examples
Structured	Data which strictly adheres to a model	DBs, CSV files
Semi-structured	Data which doesn't adhere to a model	Log/JSON files
Unstructured	Data without a structure	Media files

When using different types of data, considerations need to be made as to how the data will be “fed” to ML algorithm:

- ▶ Structured data can be usually be used directly
- ▶ Semi/unstructured data may need to undergo extra processing to extract features

# Working with Data - Cleaning



Fixing or removing data which is:

- ▶ Incomplete
- ▶ Incorrect
  - ▶ Logically - invalid data
  - ▶ Syntactically - incorrect formatting
- ▶ Duplicated
- ▶ Corrupted

# Working with Data - Preprocessing



- ▶ Transforming the data to manipulate it for specific needs
  - ▶ Encoding - label or onehot for categorical values
  - ▶ Most models only understand numerical values - need to convert strings, images, etc. to a numerical representation
- ▶ Normalising data for efficiency/accuracy
- ▶ Feature extraction for selecting/extracting important features from the dataset

# Modelling - Choosing a Model



The model we choose is heavily dependent on the *problem we are trying to solve* - making a classifier is very different to learning a policy!

Other factors include:

- ▶ Speed - how long does it take to train a model?
- ▶ Behaviour - how does the model work? How does this affect speed/accuracy/precision/etc.?
- ▶ Data - is the model suited to the data we are using?

# $k$ Nearest neighbours - Introduction

*General method:* use the  $k$  nearest "neighbours" in  $n$  dimensional space to calculate the class of new data.

Questions:

- ▶ How to choose  $k$ ?
- ▶ How to calculate distance?

# $k$ Nearest Neighbours - Calculating distance

# $k$ Nearest Neighbours - Selecting $k$



# $k$ Nearest Neighbours - Evaluation