Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Tutorials: Introduction to $Python$ and a quick introduction to basic concepts in machine learning

Following on from lecture 1, the following two tutorials and corresponding formative assessment aim to introduce you to the practical methodology of basic machine learning concepts.

Tutorial 1, $Python$ for Beginners, aims to build your familiarity with Python and with some Python libraries relevant to machine learning, including those used for data analytics and data visualisation.

First, we introduce basic Python syntax, which includes creating and assigning variables and applying functions. We then expand on this knowledge to discuss various data structures. We also introduce commonly used Python libraries for data analytics, such as $Pandas$ and $NumPy$. For these libraries, we demonstrate key data handling and processing techniques. Finally, we review the complementary data visualisation packages $Matplotlib$ and $Seaborn$ and demonstrate how to develop various explainable and communicable plots from your data.

Tutorial 2, Machine Learning for Beginners, aims to demonstrate the key techniques for machine learning development, including data preprocessing, model building, and model evaluation. First, we redefine the different types of machine learning approaches. We also introduce the machine learning library $scikit-learn$. In this tutorial, we focus on supervised learning (classification and regression) and unsupervised learning (clustering). We then expand on each of these three learning concepts, providing worked examples of machine learning development for each. We explain the importance of feature scaling and model evaluation in machine learning development. Finally, we consider the importance of understanding and explaining the results of our models.

A corresponding (optional) assessment, Machine Learning for Beginners Assessment, will help you evaluate your understanding of the practical methodology of basic machine learning concepts and demonstrate the skills you have learnt so far. This assessment is split into three parts: part 1 assesses your understanding of key data pre-processing techniques, part 2 assesses your knowledge of different types of machine learning approaches and model development and evaluation, and part 3 asks you to work through a problem, responding to specific questions, using one of the three learning concepts worked through in the Machine Learning for Beginners tutorial.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

1. Introduction to Machine Learning

In lecture 1, we cover some of the basic concepts of machine learning. Machine learning is all about teaching machines, usually computers, to learn from data to make decisions or predictions without being explicitly programmed to do so. Machine learning algorithms have a wide variety of applications, including in medicine and clinical research, where it can be more challenging for humans to manually develop algorithms to perform the desired tasks and/or identify patterns from large volumes of data.

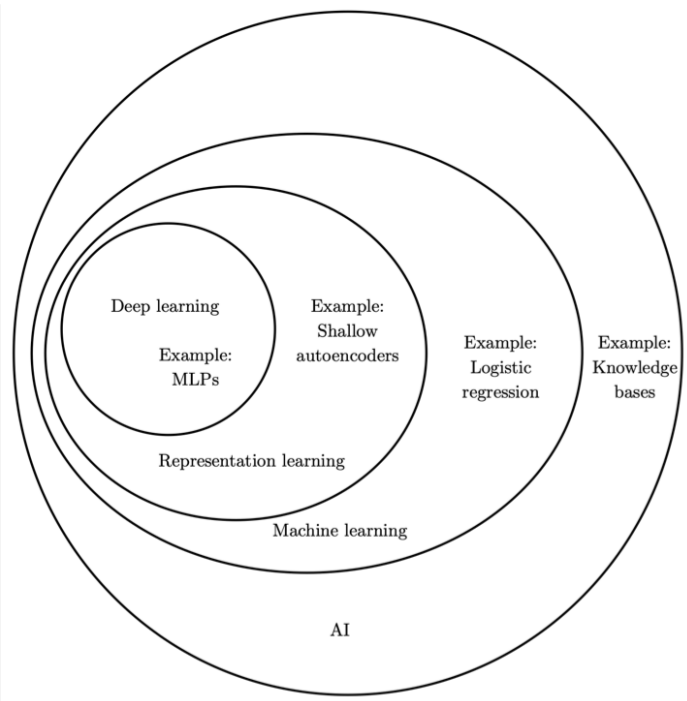

Figure 1. AI and Machine Learning
Source: Deep Learning, Ian Goodfellow et al., MIT Press.

In this first lecture, we discuss the core themes of machine learning, particularly in the context of artificial intelligence. We introduce the key components of machine learning development and summarise the three main machine learning approaches (supervised, unsupervised, and reinforcement learning). We further expand on the objectives of the different types of supervised and unsupervised learning methods, citing examples for each. We also provide a concise discussion of the advantages and limitations of each method. Finally, we consider the importance of having a way to evaluate model performance in terms of its suitability for real-world applications. Figure 1 presents key topics in AI and machine learning.

In principle, machines learn in two ways: by experience (i.e., unsupervised learning) or by examples (i.e.,

supervised) learning. In unsupervised learning, a machine learning algorithm explores a dataset, learns patterns from the data, and/or identifies correlations between different parts of the data. An example of unsupervised learning is a clustering algorithm that can learn to group a dataset into different clusters based on their inherent similarity. In supervised learning, a machine learning method is given a set of examples and the outcome labels (i.e., training data). Supervised learning algorithms use different methods to learn to predict an outcome (e.g., whether a patient has a disease) based on a set of observations/measurements, by learning from examples and training the model to make sufficiently accurate predictions. The input data usually contains several attributes (e.g., age, heart rate, blood pressure, and body temperature for a patient's data). These attributes are often referred to as features or dimensions in machine learning. Machine learning models usually accept the input data as a set of vectors in which each feature is a column (or a row if you consider a vertical vector).

Another category of machine learning models is reinforcement learning. In reinforcement learning, an intelligent agent takes a series of actions to maximise a cumulative reward. For example, an intelligent agent learns to predict treatment regimens in a patient's process, and each time it makes a correct prediction, it receives a reward. Each time it makes a mistake, a penalty is applied. The method continues to learn until it converges to a stable, sufficiently reliable solution. You can imagine an intelligent agent that learns to navigate a robot through an environment with different obstacles, with the reward for avoiding them. Reinforcement learning is seen as one of the three main machine learning categories, alongside supervised and unsupervised learning methods.

The first tutorial (Python for Beginners) aims to build your familiarity with Python and some core Python libraries relevant to machine learning, including those used for data analytics and visualisation. The second tutorial (Machine Learning for Beginners) will then demonstrate the key techniques for machine learning development, including data preprocessing, model building, and model evaluation.

A corresponding assessment will help you evaluate your understanding of basic machine learning concepts and demonstrate the skills you have learned so far.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 2. Linear models

In lecture 2, we cover some of the basic concepts of linear models.
In statistics, linear models describe a continuous response variable (the dependent variable) as a function of one or more predictor variables (the independent variable(s)). Such models can help us to understand and predict the behaviour of complex systems and analyse patterns in data.

In this lecture, we introduce the core theme of linear regression. We discuss the differences between the two main types of regression models (lasso and ridge) and when they might be used, citing applications for each. We also provide an overview of the different metrics that can be used to evaluate the performance of such models. We further expand on the generalisation of linear regression to the binary classification setting. Finally, we consider the importance of choosing the right model based on the training data.

Figure 2.1 demonstrates how linear regression assumes a target variable $(f(X))$ to be dependent on a feature (X). Figure 2.2 illustrates a decision boundary used in a binary classification setting. Figure 2.3 illustrates the difference between training and generalisation errors of a model with regard to the data.

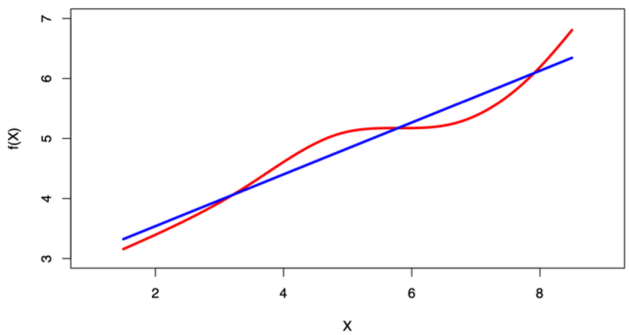

Figure 2.1. Linear Regression

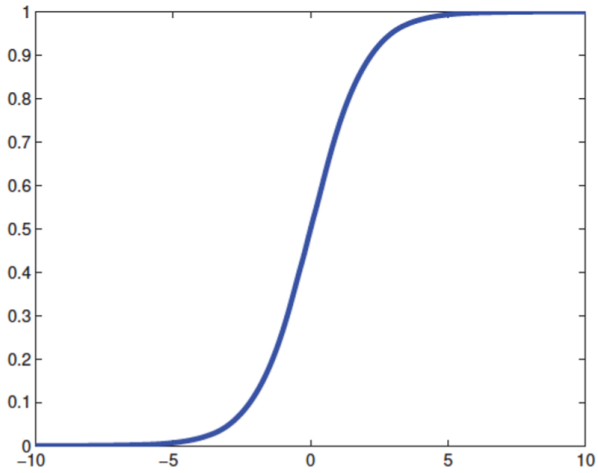

Figure 2.2. Decision Boundary for Binary Classification

In principle, supervised learning techniques are often designed in two ways: classification and regression. In linear regression, the linear regression model uses a linear model with coefficients $w = (w1, \ldots, wn)$ to minimise the residual sum of squares between the observed targets in the dataset and the targets that are predicted by the linear approximation used in the model.

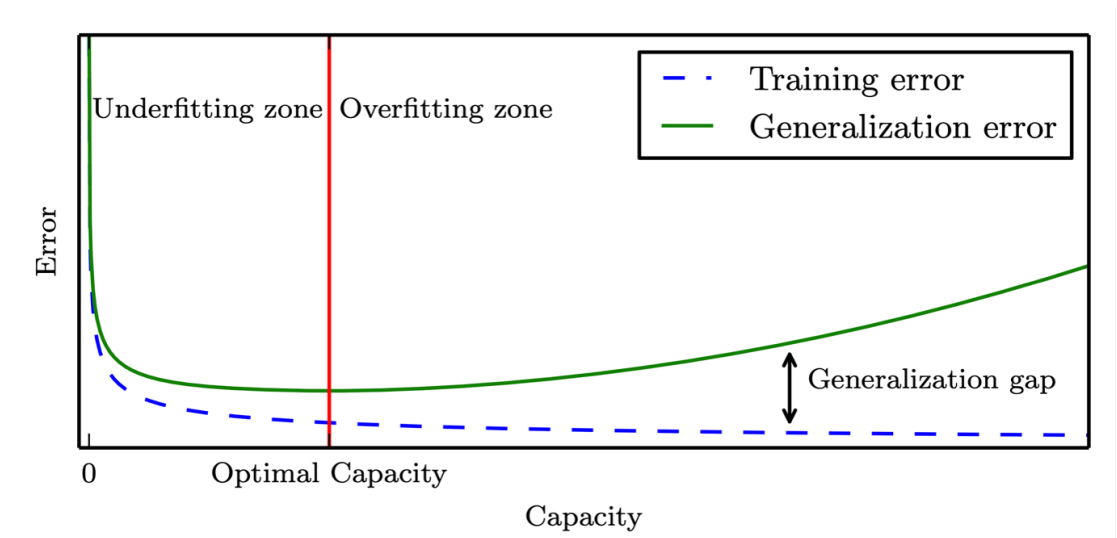

Figure 2.3. Training and Test Errors
Source: Deep Learning, Ian Goodfellow et al., MIT Press.

An example of regression in neuroscience/neurology is predicting the change in a person living with dementia's mini-mental state examination (MMSE) score as a function of the duration since diagnosis.
Using linear regression in a binary classification setting is the process of applying a linear combination of the inputs passed through a logistic function. This is otherwise known as logistic regression due to its similarity to linear regression (although it is a form of classification, not regression!). An example of how you might use a logistic regression model is to predict whether a patient has a disease based on a given set of observations/measurements.

The lab code on the GitHub page provides practical examples of developing linear regression models with Lasso and ridge regularisation. The lecture slides and the lab will also demonstrate how linear regression can be used in a binary classification setting (logistic regression).

ML4NS
Machine Learning for Neuroscience

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 3. Probability and Information Theory

In lecture 3, we cover some of the primary concepts of probability and information theory.

With sufficient data, statistics can enable us to calculate probabilities using real-world observations. Here, probability provides the theory, while statistics provides the tools to test that theory using data. In the end, descriptive statistics such as the mean and standard deviation of the data become proxies for theoretical. This is because real-world probabilities are often quite difficult to calculate. As such, we rely on statistics and data. With more and more data, we can become more confident that we create models that represent the true probability of these events occurring.

In this lecture, we first introduce probability regarding how likely it is that a discrete random variable (or continuous) is to take on each of its possible states, otherwise known as the probability distribution or the probability density function. We further introduce the cumulative distribution function for continuous random variables. We also describe joint probabilities and conditional probability. We further expand on the two key characteristics of a probability distribution (mean and standard deviation). Finally, we consider the different types (with examples) of probability distributions, including the Bernoulli, Binomial, Multinomial, Multinoulli, and Gaussian (normal) and standard normal distributions.

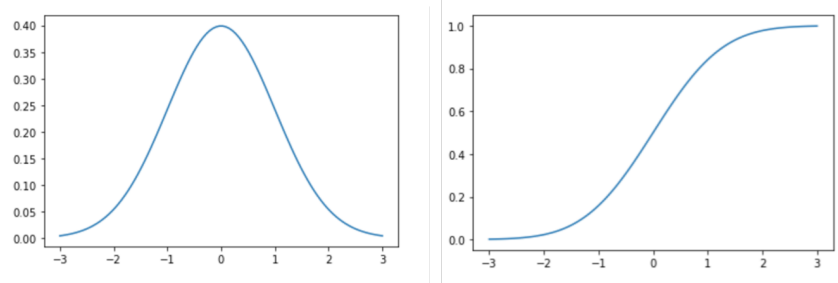

Figure 3.1. Probability Density Function vs Cumulative Density Function

Figure 3.1 illustrates the difference between a probability density function (in the case of discrete random variables) and a cumulative density function (in the case of continuous random variables).

This lecture also introduces information theory, with the basic intuition that learning that an unlikely event has occurred is more informative than learning that a likely event has occurred. Information theory has three main properties: likely events should have low information content, and in the extreme case, events that are guaranteed to happen should have no information content whatsoever; less likely events should have higher information content, and independent events should have additive information. Finally, we introduce conditional independence in the context of using Markov Chains (describing the possible sequences of events) to model behaviour. Figure 3.2. illustrates the Markov chain representation of in-home movements in a household of a person living with dementia.

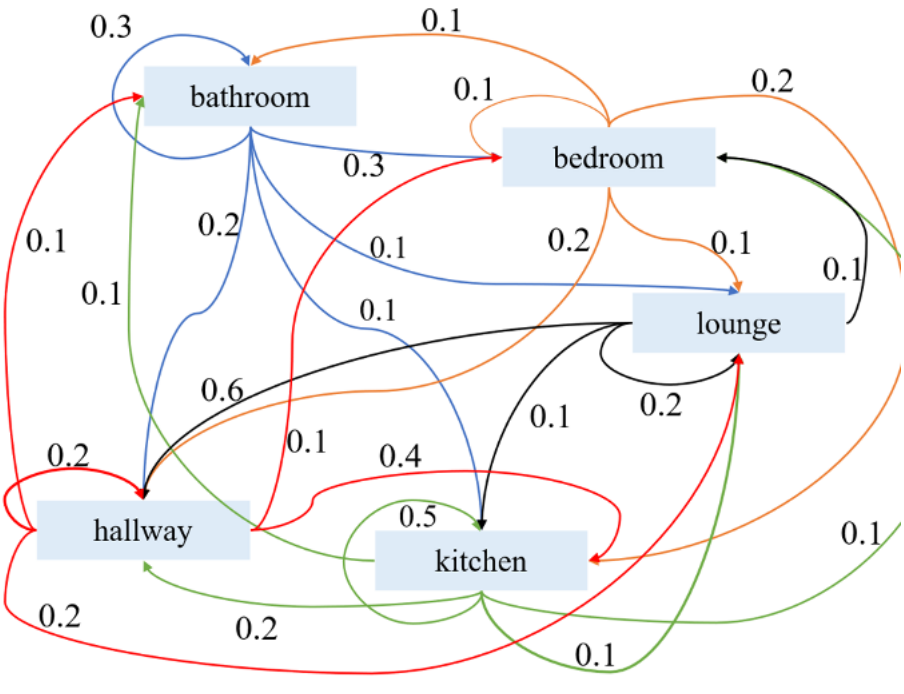

Figure 3.2. Figure 2. A Markov chain representation of in-home movements in a household of a person living with dementia.

ML4NS
Machine Learning for Neuroscience

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 3. Probability and Information Theory

In lecture 3, we cover some of the primary concepts of probability and information theory.

With sufficient data, statistics can enable us to calculate probabilities using real-world observations. Here, probability provides the theory, while statistics provides the tools to test that theory using data. In the end, descriptive statistics such as the mean and standard deviation of the data become proxies for theoretical. This is because real-world probabilities are often quite difficult to calculate. As such, we rely on statistics and data. With more and more data, we can become more confident that we create models that represent the true probability of these events occurring.

In this lecture, we first introduce probability regarding how likely it is that a discrete random variable (or continuous) is to take on each of its possible states, otherwise known as the probability distribution or the probability density function. We further introduce the cumulative distribution function for continuous random variables. We also describe joint probabilities and conditional probability. We further expand on the two key characteristics of a probability distribution (mean and standard deviation). Finally, we consider the different types (with examples) of probability distributions, including the Bernoulli, Binomial, Multinomial, Multinoulli, and Gaussian (normal) and standard normal distributions.

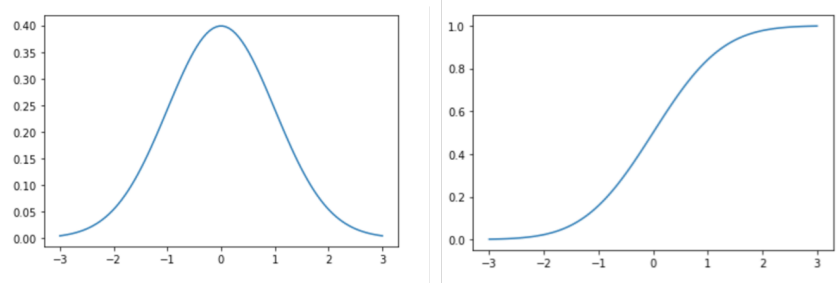

Figure 3.1. Probability Density Function vs Cumulative Density Function

Figure 3.1 illustrates the difference between a probability density function (in the case of discrete random variables) and a cumulative density function (in the case of continuous random variables).

This lecture also introduces information theory, with the basic intuition that learning that an unlikely event has occurred is more informative than learning that a likely event has occurred. Information theory has three main properties: likely events should have low information content, and in the extreme case, events that are guaranteed to happen should have no information content whatsoever; less likely events should have higher information content, and independent events should have additive information. Finally, we introduce conditional independence in the context of using Markov Chains (describing the possible sequences of events) to model behaviour. Figure 3.2. illustrates the Markov chain representation of in-home movements in a household of a person living with dementia.

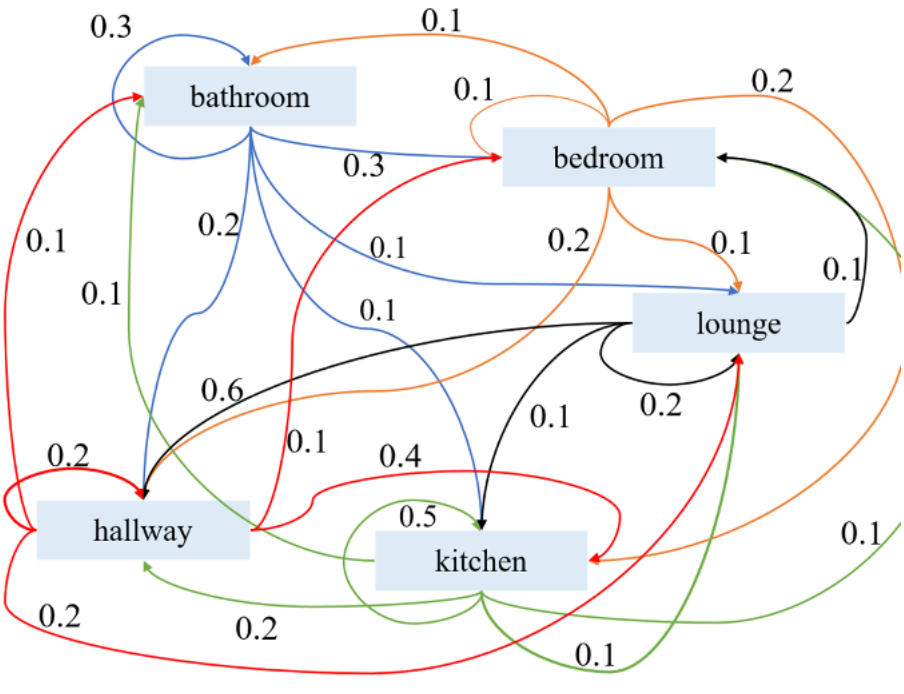

Figure 3.2. Figure 2. A Markov chain representation of in-home movements in a household of a person living with dementia.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 4. Bayesian models

In lecture 4, we cover some of the basic concepts of Bayesian models. Bayes' theorem, named after the statistician Thomas Bayes, describes the probability of an event based on prior knowledge of conditions that might be related to an event. For example, if the risk of developing a neurological disorder is known to increase with age, Bayes' theorem allows the risk to an individual of a given age to be assessed given that age. In this lecture, we introduce Bayes' rule as four key components: the posterior or $p(A|B)$, the likelihood or $p(B|A)$, the prior or $p(A)$, and the marginalisation or $p(B)$.

Figure 4.1 illustrates Bayes' rule, which combines the definition of conditional probability with the product and sum rules.

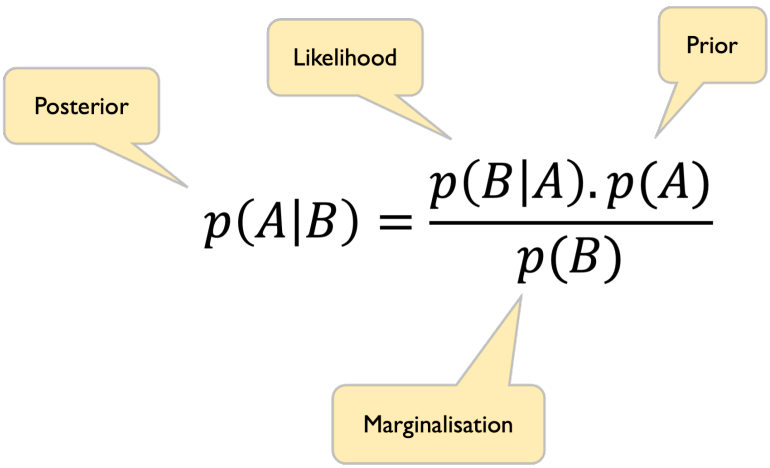

Figure 4.1. Bayes' Rule

We then discuss the use of Bayes' rule in classification (the probability of a label given some features). This is done using Naïve Bayes classifiers. Naive Bayes classifiers assume that each feature is independent of every other feature. In reality, this assumption is not often the case. However, this generalisation allows Naïve Bayes to create "Naïve" but sometimes efficient prediction and classification models. We further expand on the objectives of the different types of Naïve Bayes classifiers, citing examples for each. While the Naïve classifiers could be helpful in different applications, the derived probability scores are not reliable (i.e., the probability score in Naïve Bayes are good for the classification purpose but not for interpreting the actual probabilities of belonging to each class). Finally, we review the different metrics by which we might evaluate the performance of such models.

Figure 4.2 illustrates a confusion matrix for a classification model. From this matrix, we can derive the number of true positives, true negatives, false positives, and false negatives. From these numbers, we can further derive our measures of accuracy, recall, precision, and the F1-score.

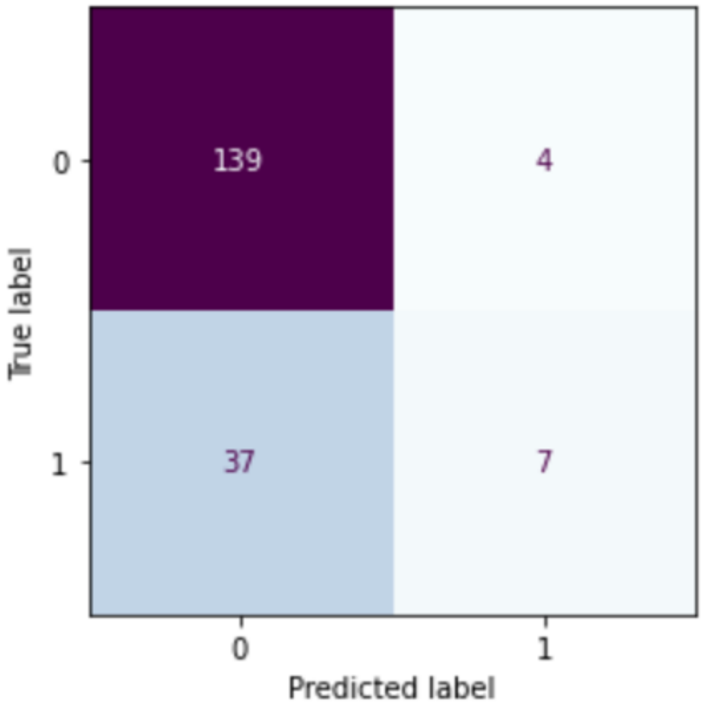

Figure 4.2. A Confusion Matrix for a Classification Model

The tutorial and assessment are combined with the previous lecture's tutorial and assessment and aim to further develop your skills in model building and evaluation (in the context of using Bayesian models). This tutorial and assessment will ask you to use scikit-learn to develop your models. To further expand your understanding of how Naïve Bayes classifiers work, you could also try developing your own Bayesian model from scratch (see: https://medium.com/@rangavamsi5/naïve-bayes-algorithm-implementation-from-scratch-in-python-7b2cc39268b9).

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 5. Ensemble models and Kernel-based models

In lecture 5, we cover the basic principles of ensemble and kernel-based models. In our previous lectures, we made two critical assumptions about the data: that a fixed-size feature vector could represent each item we were trying to process, classify, or cluster and that the items were linearly separable. In this lecture, we discuss alternative approaches for items/concepts that are not linearly separable and cannot be best represented as fixed-sized feature vectors (e.g. protein sequences, which can be of variable lengths or evolutionary trees, which have variable shape and size).

We first introduce kernel functions, a process that applies a function to transform the data into a higher-dimensional space by which the data items can then be compared/processed. We also provide an overview of the commonly used kernels, including polynomial, sigmoid, and Gaussian radial basis functions. Figure 5.1 illustrates the methodology of a kernel-based model for comparing and processing data.

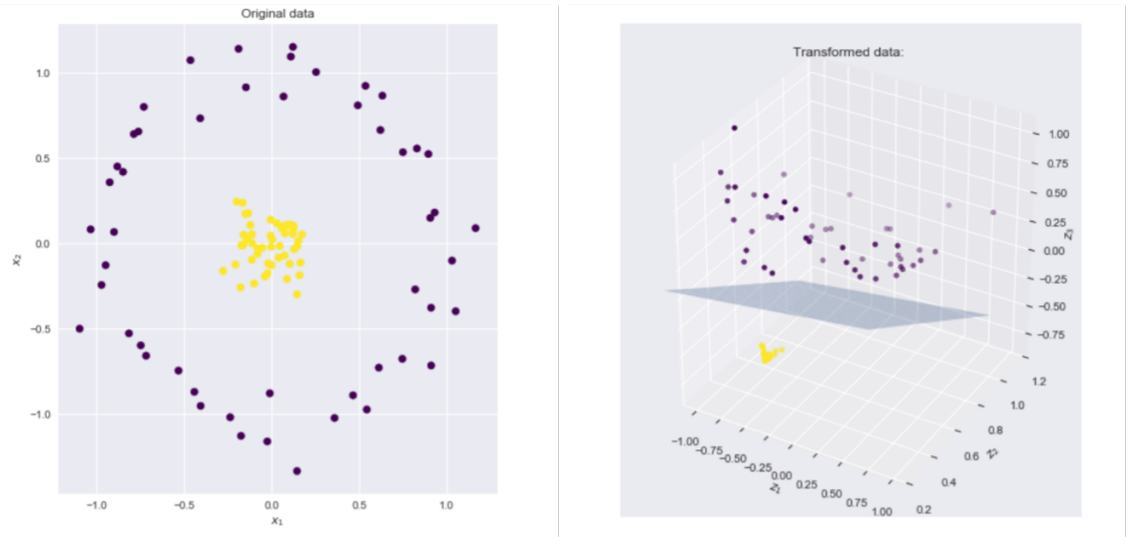

Figure 1. Example of data mapping using a kernel function
Source: https://xavierbourretsicotte.github.io/Kernel_feature_map.html

We further discuss how kernel functions can be incorporated into supervised machine learning models using support vector machines, highlighting their advantages and disadvantages.

We then introduce ensemble methods. This is the practice of learning from multiple decision trees. The goal of ensemble methods is to combine the predictions of several base estimators trained with a given learning algorithm to improve generalisability/robustness relative to a single estimator. We also discuss the two main families of ensemble methods: bagging methods in which several base estimators are built independently with their predictions then averaged, and boosting methods, in which base estimators are built sequentially and

the aim is to reduce the bias of the combined estimator.

Finally, we introduce the game theoretic approach SHAP (Shapley Additive exPlanations) to explain the output of any machine learning model. We consider how such an approach can be used to connect optimal credit allocation with local explanations regarding how a model makes its decision. Figure 5.2 illustrates a SHAP plot to demonstrate the impact (size and direction of effect) of each feature on the outcome variable.

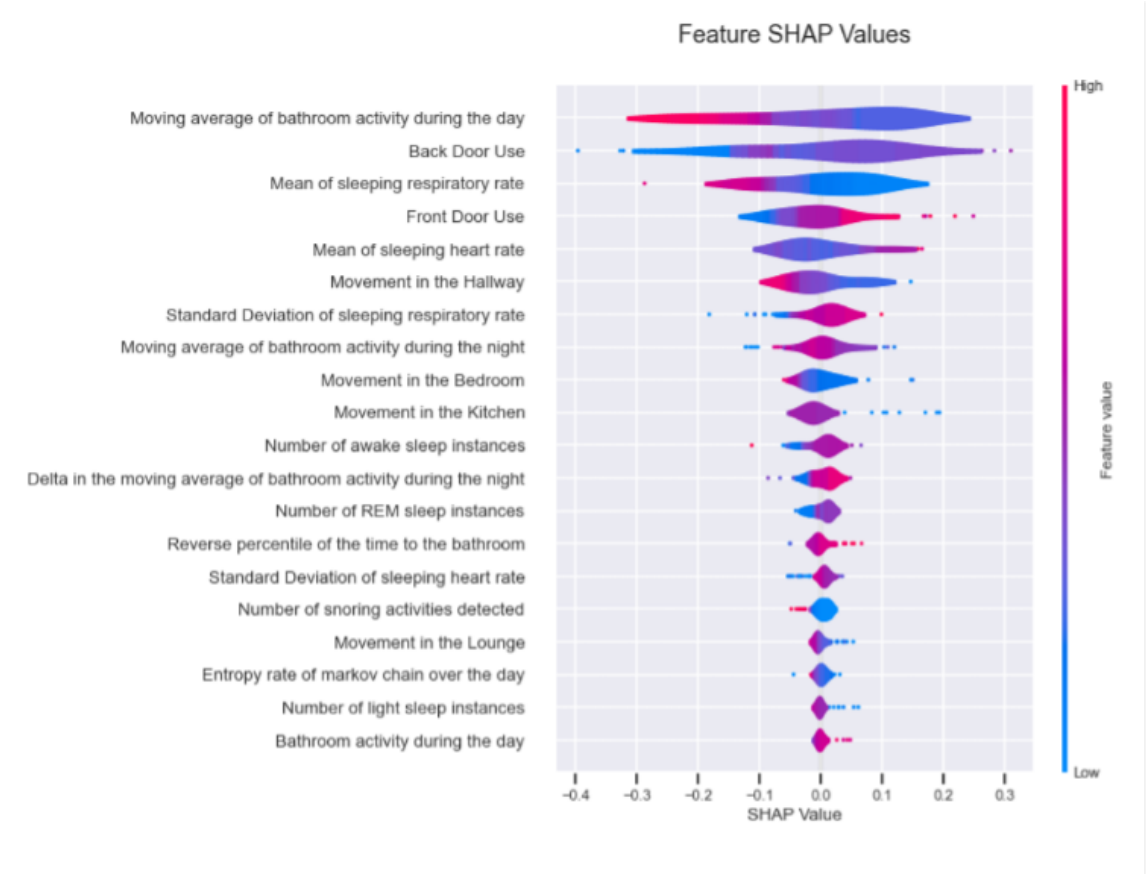

Figure 5.2. An example of a SHAP plot

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 6. Neural Networks

In lecture 6, we cover the basic concept behind the design and use of neural networks. In machine learning, we are interested in modelling biologically inspired networks known as artificial neural networks to help us solve learning and decision-making problems. Our previous lectures have focused on linear models; however, linearity implies the weaker assumption of monotonicity (i.e., that any increase in a feature will always result in an increase or decrease in the model's output). Such an assumption is not always true in real-world applications.

The simplest kind of neural network is a single-layer perceptron model, which consists of a single layer of output nodes to which the inputs are directly fed via a set of weights. In each node, the sum of the products of the inputs and their respective weights is calculated, and if the value exceeds a threshold, the neuron fires and takes the activated value. Otherwise, the neuron takes the deactivated value. Neurons with this type of activation function are also called artificial neurons or linear threshold units.

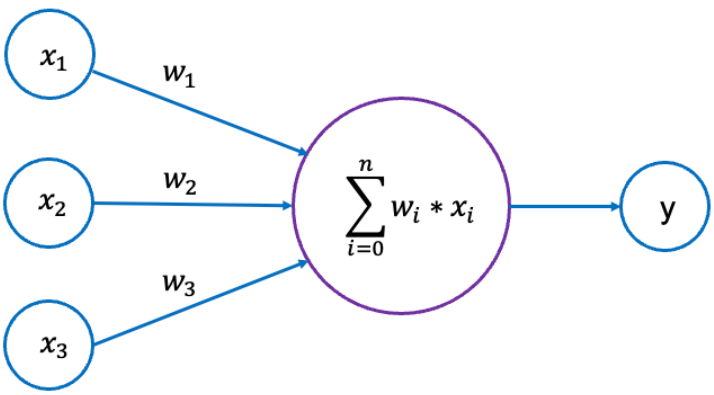

Figure 6.1. A simple illustration of an artificial neuron

Figure 6.1 illustrates a single artificial neuron (a type of single-layer perceptron that acts as a binary classifier).

In this lecture, we discuss how the limitations of linear models can be overcome by incorporating one or more hidden layers (with activation functions) into a neural network. We define network depth and width as the number of hidden layers and the number of nodes in the hidden layers, respectively. We also introduce two commonly used activation functions for adding nonlinearity to neural networks: the $sigmoid$ function and rectified linear units (ReLU). Networks with multiple layers of computational units are known as multi-layer perceptrons (MLPs).

Figure 6.2 illustrates the architecture of a fully connected, feed-forward multilayer perceptron.

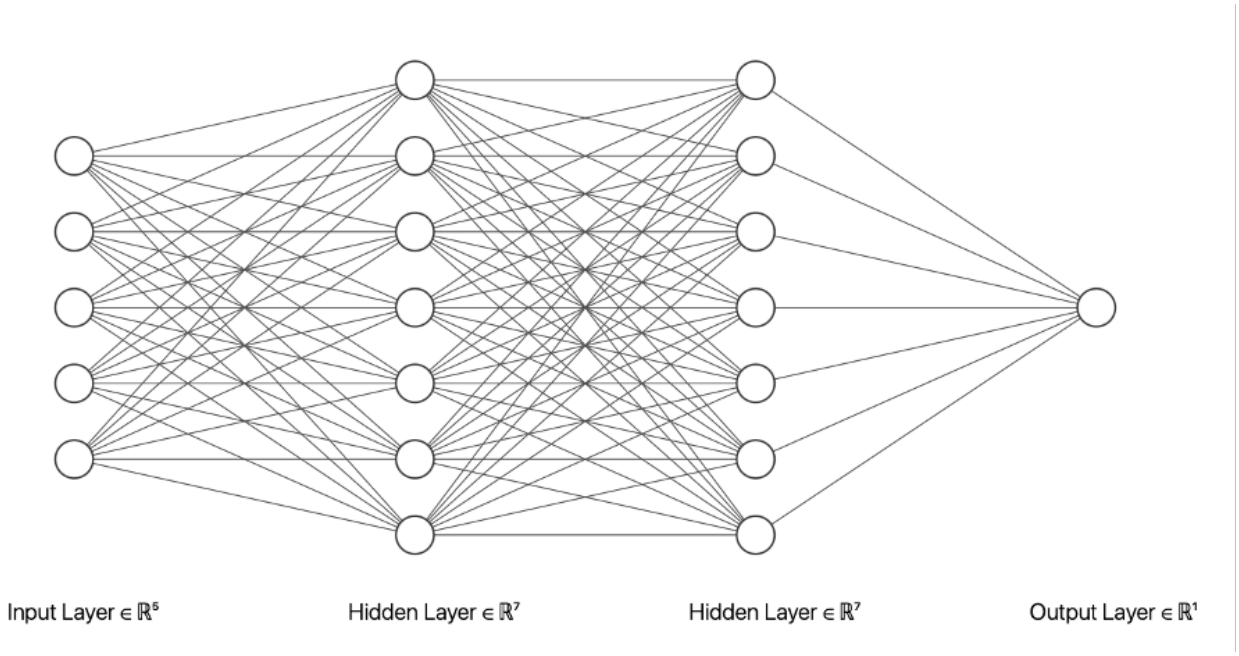

Figure 6.2. A sample multilayer perceptron architecture
Created using: https://alexlenail.me/NN-SVG/

MLPs are often interconnected in a feed-forward way, which is where the connections between nodes do not form a cycle as in recurrent neural networks (RNNs).

We further expand on how we train feed-forward neural networks using forward propagation and backpropagation. Finally, we consider how to design our neural networks, including selecting an appropriate number of hidden layers and a suitable learning rate. We also introduce and explain the objective of different adaptive learning rate optimisation algorithms.

The tutorial aims to build your familiarity with the machine learning framework $PyTorch$ and will also demonstrate the basics of building and training simple neural networks.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 7. Convolutional Neural Networks (CNNs)

In lecture 7, we cover the basic principles of Convolutional Neural Networks (CNNs). Expanding on our previous lecture on neural networks and multi-layer perceptrons, here we introduce convolutional neural networks (CNNs), a specialised kind of neural network for processing data that has a known grid-like topology, such as image data (a 2D grid of pixels) and time-series data (a 1D grid when samples are taken at regular time intervals).

We further expand on the structure of a CNN, including a description of the convolution and pooling processes. In a convolutional layer, a sliding kernel is applied to an input to produce an output. In this lecture, we consider how to design our convolutional layers, including choice of padding, choice of stride, and the number of input and output channels.

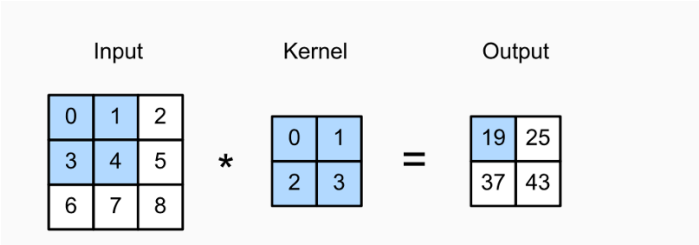

Figure 7.1. Example of a convolution
Source: Dive into Deep Learning, Aston Zhang *et al.*, https://d2l.ai/

Since we typically use small kernels in a given convolution, we risk losing pixels along the image perimeter. One solution to this problem is to augment/pad our input along the boundary, thereby retaining the original input's dimensions. This is known as the same padding. If we were to apply a kernel without padding, we would produce an output of the exact dimensions as the kernel (as in Figure 7.1), and this is known as valid padding. Figure 7.1 illustrates a convolution that might occur in a CNN by taking an input and applying a sliding kernel (convolution window).

The pooling layer of a CNN is responsible for reducing the spatial size of the previous convolved layer. This is to decrease the computational power required to process the data through dimensionality reduction. Here, we introduce two types of pooling that can occur in a pooling layer: maximum pooling returns the maximum value from the portion of the input covered by the kernel, and average pooling returns the average of all values in that portion. Figure 7.2 illustrates a 2 x 2 Maximum pooling

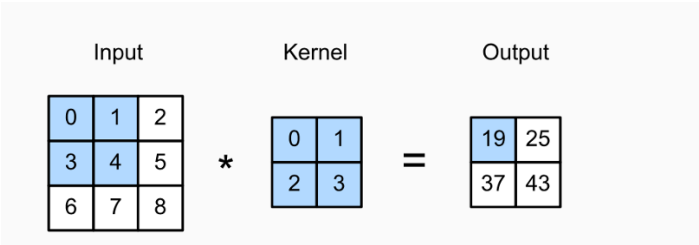

Figure 7.2. Example of maximum pooling
Source: Dive into Deep Learning, Aston Zhang *et al.*, https://d2l.ai/

Having successfully enabled the model to understand the features, we flatten the final output and feed it to a regular neural network for classification (e.g., $Softmax$). A fully connected layer is often added to learn non-linear combinations of the high-level features as represented by the output of the convolution layer.

Finally, we discuss the various architectures of CNNs that have been key to solving current machine learning and decision-making problems. The lab notebook/practice aims to show how to work with $PyTorch$ libraries and to demonstrate the key techniques needed to build a CNN for the classification of image data.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 8. Applications of Machine Learning in Neuroscience

In lecture 8, we cover examples of real-world applications of machine learning in neuroscience. We also discuss how concepts from neuroscience have been applied to machine learning. In this lecture, we discuss how neuroscience can inspire new types of algorithms and architectures. We first review the content of previous lectures before expanding further on different sub-fields of machine learning. These include reinforcement learning, deep learning, attention-based learning, and continual learning, with citations to state-of-the-art works for each.

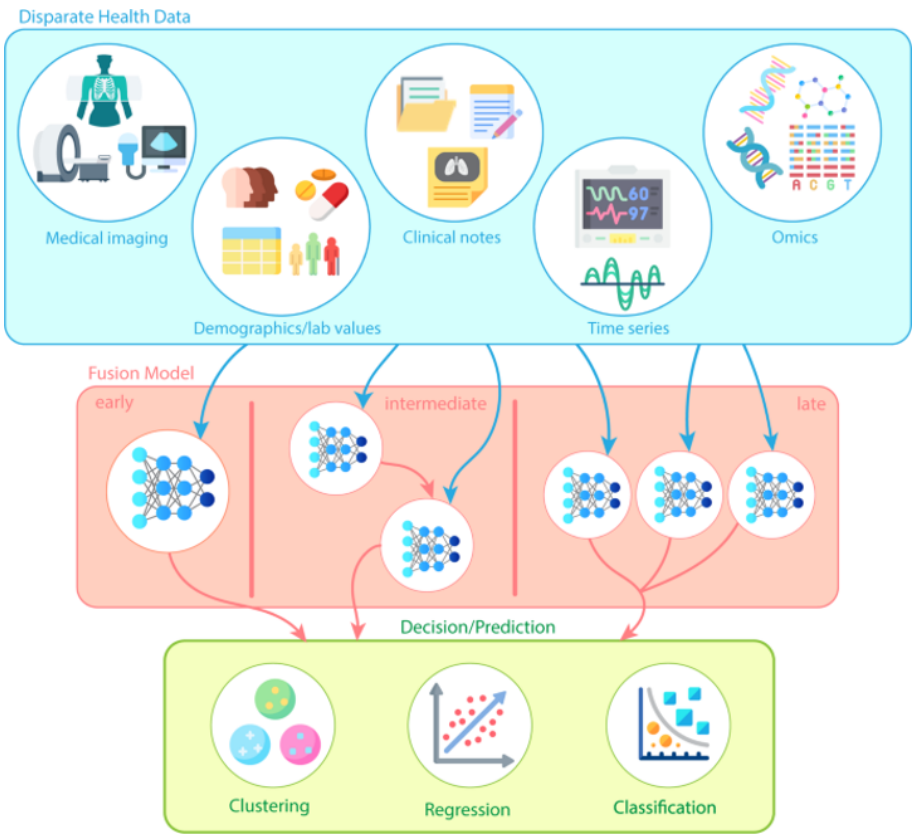

Figure 8.1. Applications of machine learning at different stages of disease
Source: Kline, A., Wang, H., Li, Y. *et al.* Multimodal machine learning in precision health: A scoping review. npj Digit. Med. 5, 171 (2022). https://doi.org/10.1038/s41746-022-00712-8

We will explore the architecture of Transformer models, a foundational component in many state-of-the-art Large Language Models (LLMs) that power today's advanced AI systems. We will break down the Transformer architecture, explain why it outperforms traditional models, and discuss how it enables machines to understand and generate human language.

We also consider applications of machine learning in precision health care, ranging from neuroimaging to electronic health record analysis to disease diagnosis and treatment to predicting disease progression. Figure 8.1 illustrates how machine learning can be applied in healthcare at different stages of a disease.

Finally, we discuss the practical methodology of machine learning approaches, including dimensionality reduction and imputation of missing values.

Slides and notebooks: https://ml4ns.github.io

Notes by: *Nan Fletcher-Lloyd, Payam Barnaghi*

Lecture 9. Ethical considerations and responsible machine learning

In lecture 9, we look at ethical considerations and the importance of responsible machine learning.

The goal of developing machine learning models and systems in healthcare is to deploy them in real-world settings to aid with patient care. These systems can be used as decision-support tools by keeping the human in the loop. Therefore, prior to deployment, it is crucial to investigate the models/systems with regard to trustworthiness, reliability and robustness. It is also important to understand user perception of these models/systems.

In this lecture, we discuss how the way in which a model is trained can affect the appropriateness of a model for a given problem, introducing the approaches of online and offline learning. We further expand on the use of continual learning methods to emulate human learning, and we also introduce the forgetting problem in such models.

We consider the importance of investigating the suitability of a dataset for its planned analysis with regard to developing clinically applicable solutions (how and when the data was collected, and the purpose of the model that such data will be used to train).

We then review ethical implications of machine learning methods with regard to algorithmic bias, in which a model is trained on data likely to be influenced by many facets of social inequality, particularly those who contribute the most data. Here, we discuss the importance of having a robust evaluation of the models backed up by qualitative analyses. We further stress the importance of transparency in the reporting of new works and introduce the notion of explainability in our models.

We also note the importance of deploying and maintaining such models in an ethical, legal and morally responsible manner, including keeping up to date on regulatory requirements. Figure 9.1 illustrates a roadmap for developing and deploying effective machine learning systems in real-world settings.

Figure 9.1. A roadmap for deploying effective machine learning systems in healthcare applications
Source: Wiens, J., Saria, S., Sendak, M. *et al.* Do no harm: a roadmap for responsible machine learning for health care. Nat Med 25, 1337–1340 (2019).

Finally, we discuss the challenges posed by data collection and analysis, particularly regarding privacy.

This topic is assessed in conjunction with Topic 8 (Real-world Applications of Machine Learning in Neuroscience) and will evaluate your understanding of the importance of responsible machine learning.

Drawing on what you have learned across this module so far, we ask you to review a given paper focusing on the different aspects of machine learning used in the presented work, particularly with regards to the limitations and potential ethical challenges posed using such methods and models, and exploring how such issues can be overcome, e.g., can other models be used that would be more applicable and why or can different metrics be used to evaluate a model that might be more appropriate?.