Machine Learning Theory and Applications

Hands-on Use Cases with Python on Classical and Quantum Machines

Xavier Vasques

WILEY

Machine Learning Theory and Applications

Machine Learning Theory and Applications

Hands-on Use Cases with Python on Classical and Quantum Machines

Xavier Vasques
IBM Technology, Bois-Colombes, France
Laboratoire de Recherche en Neurosciences Cliniques, Montferriez sur lez, France
Ecole Nationale Supérieure de Cognitique Bordeaux, Bordeaux, France

WILEY

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

Trademarks: Wiley and the Wiley logo are trademarks or registered trademarks of John Wiley & Sons, Inc. and/or its affiliates in the United States and other countries and may not be used without written permission. All other trademarks are the property of their respective owners. John Wiley & Sons, Inc. is not associated with any product or vendor mentioned in this book.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Further, readers should be aware that websites listed in this work may have changed or disappeared between when this work was written and when it is read. Neither the publisher nor authors shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic formats. For more information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Names: Vasques, Xavier, author.
Title: Machine learning theory and applications : hands-on use cases with
Python on classical and quantum machines / Xavier Vasques.
Description: Hoboken, New Jersey : Wiley, [2024] | Includes index.
Identifiers: LCCN 2023039030 (print) | LCCN 2023039031 (ebook) | ISBN
9781394220618 (hardback) | ISBN 9781394220632 (adobe pdf) | ISBN
9781394220625 (epub)
Subjects: LCSH: Machine learning. | Quantum computing. | Python (Computer
program language)
Classification: LCC Q325.5 .V37 2024 (print) | LCC Q325.5 (ebook) | DDC
006.3/1–dc23/eng/20231023
LC record available at https://lccn.loc.gov/2023039030
LC ebook record available at https://lccn.loc.gov/2023039031

Cover image: Wiley
Cover design: © anand purohit/Getty Images

Set in 9.5/12.5pt STIXTwoText by Straive, Pondicherry, India

To my wife Laura and my daughter Elsa

Contents

Foreword

The wheels of time turn faster and faster, and as individuals and as human society we all need to adapt and follow. Progress is uncountable in all domains.

Over the last two years, the author has dedicated many hours over weekends and late evenings to provide a volume of reference to serve as a guide for all those who plan to travel through machine learning from scratch, and to use it in elaborated domains where it could make a real difference, for the good of the people, society, and our planet.

The story of the book started with some blog post series that reached many readers, visits, and interactions. This initiative was not a surprise for me, knowing the author's background and following his developments in the fields of science and technology. Almost 20 years passed since I first met Xavier Vasques. He was freshly appointed for a PhD in applied mathematics, but he was still searching for a salient and tangible application of mathematics. Therefore, he switched to a domain where mathematics was applied to neurosciences and medicine. The topic related to deep brain stimulation (DBS), a neurosurgical intervention using electric stimulation to modulate dysfunctional brain networks to alleviate disabling symptoms in neurological and psychiatric conditions. Therapeutic electric field modelling was the topic of his PhD thesis in Neurosciences. He further completed his training by a master's in computer science from The Conservatoire National des Arts et Métiers, and continued his career in IBM where all his skills combined to push technological development further and fill the gap in many domains through fruitful projects. Neurosciences remained one of his main interests. He joined the Ecole Polytechnique Fédérale de Lausanne in Switzerland as researcher and manager of both the Data Analysis and the Brain Atlasing sections for the Blue Brain Project and the Human Brain Project in the Neuroinformatics division. Back at IBM, he is currently Vice-President and CTO of IBM Technology and Research and Development in France.

Throughout his career, Xavier could contemplate the stringent need but also the lack of communication, understanding, and exchanges between mathematics, computer science, and industry to support technological development. Informatics use routines and algorithms but we do not know precisely what lies behind it from the mathematical standpoint. Mathematicians do not master codes and coding. Industry involved in production of hardware does not always master some of the concepts and knowledge from these two domains to favor progress.

The overall intention of this book is to provide a tool for both beginners and advanced users and facilitate translation from theoretical mathematics to coding, from software to hardware, by understanding and mastering machine learning.

The very personal approach with handwriting, "hand-crafted" figures, and "hands on" approach, makes the book even more accessible and friendly.

May this book be an opportunity for many people, and a guidance for understanding and bringing forth, in a constructive and meaningful way, data science solely for the good of mankind in this busy, febrile, and unsteady twenty-first century.

I am writing from the perspective of the clinician who so many times wondered what is a code, an algorithm, supervised and unsupervised machine learning, deep learning, and so forth.

I see already vocations from very young ages, where future geeks (if the term is still accepted by today's youth) will be able to structure their skills and why not push forward the large amount of work, by challenging the author, criticizing and completing the work.

It is always a joy to see somebody achieving. This is the case for this work by Xavier who spared no energy and time to leave the signature of his curriculum but especially that of his engagement and sharing for today's society.

Montpellier, July 2023

Dr. Laura Cif, MD, PhD
Service de Neurologie, Département des
Neurosciences Cliniques, Lausanne University
Hospital, Lausanne, Switzerland

Laboratoire de Recherche en Neurosciences
Cliniques, France

Acknowledgments

I would like to express my deepest gratitude to my loving wife and daughter, whose unwavering support and understanding have been invaluable throughout the journey of writing this book. Their patience, encouragement, and belief in my abilities have been a constant source of motivation.

To my wife, thank you for your endless love, understanding, and for standing by me during the countless hours spent researching, writing, and editing. Your unwavering support and belief in my work have been a guiding light.

To my dear daughter, thank you for your patience, understanding, and for being a constant source of inspiration. Your enthusiasm for learning and exploring new ideas has fueled my passion for this project.

I am truly grateful for the love, understanding, and encouragement that my wife and daughter have provided. Without them, this book would not have been possible. Their presence in my life has made every step of this journey meaningful and fulfilling.

Thank you, from the bottom of my heart.

Montpellier, July, 2023 *Xavier Vasques*

General Introduction

The Birth of the Artificial Intelligence Concept

Thomas Hobbes begins his *Leviathan* by saying, “Reason is nothing but reckoning.” This aphorism implies that we could behave like machines. The film *The Matrix,* meanwhile, lets us imagine that we are controlled by an artificial creature in silico. This machine projects into our brains an imaginary, fictional world that we believe to be real. We are therefore deceived by calculations and an electrode piercing the back of our skull. The scenarios abound in our imagination. Fiction suggests to us that one day, it will be easy to replicate our brains, like simple machines, and far from the complexity that we currently imagine. Any mainstream conference on artificial intelligence (AI) routinely shows an image from *The Terminator* or *2001: A Space Odyssey.*

If “reason is nothing but reckoning,” we could find a mathematical equation that simulates our thinking, our consciousness, and our unconsciousness. This thought is not new. Since the dawn of time, humans have constantly sought to reproduce nature. The question of thought, as such, is one of the great questions that humanity has asked itself. What makes Odysseus able to get away with tricks, flair, imagination, and intuition? How do we reflect, reason, argue, demonstrate, predict, invent, adapt, make analogies, induce, deduce, or understand? Is there a model that allows us to approach these things? Throughout our history, we have claimed that a machine cannot calculate like humans or speak, debate, or multitask like humans. Our desire for mechanization over millennia has shown us that machines, tools, and techniques can accomplish these tasks that we had thought were purely human. Does this mean that machines have surpassed humans? We can only acquiesce to a wide range of tasks. Are machines human? No!

Since our species emerged, we have continued to create tools intended to improve our daily lives, increase our comfort, make our tasks less painful, protect us against predators, and discover our world and beyond. These same tools have also turned against us, even though they had not been endowed with any intelligence. Beyond the use as a tool of AI, the quest for the thinking machine can be viewed in a slightly different way. It can be seen as a desire to know who we are, or what we are. It can also be considered as a desire to play God. Since ancient times, philosophers and scientists have been asking these questions and trying to understand and imitate nature in the hope of giving meaning to our being and sometimes to gain control. This imitation involves the creation of simple or complex models more or less approaching reality. So it is with the history of AI. AI comes not only from the history of the evolution of human thought on the body and the nature of the mind through philosophy, myths, or science but also from the technologies that have accompanied us throughout our history, from the pebble used to break shells to the supercolliders used to investigate quantum mechanics. Some historians have found ancient evidence of human interest in artificial creatures, particularly in ancient Egypt, millennia before the coming of Jesus Christ (BCE). Articulated statues, which could be described as automatons, were used during religious ceremonies to depict a tradesperson such as a kneading baker or to represent Anubis or Qebehsenouf as a dog’s head with movable jaws. Even if they are only toys or animated statuettes using screws, levers, or pulleys, we can see a desire to artificially reproduce humans in action. These objects are not capable of moving on their own, but imagination can help. Automatons may become a symbol of our progress and emancipatory promises. These advances have also been an opportunity for humans to question their own humanity. The use of AI has been the subject of many questions and sources of concern about the future of the human species and its dehumanization.

In Greek mythology, robot servants made by the divine blacksmith Hephaestus lay the foundations for this quest for artificial creation. Despite the fact that Hephaestus is a god with deformed, twisted, crippled feet, this master of fire is considered an exceptional craftsman who has created magnificent divine works. A peculiarity of the blacksmith, recounted in the *Iliad,* is his ability to create and animate objects capable of moving on their own and imitating life. He is credited with creating golden servants who assist him in his work and many other automatons with different functions, including guard dogs to protect the palace of Alkinoos, horses for the chariot of the Cabires, or the giant Talos to guard the island of Crete. Items crafted by Hephaestus are also clever, allowing the gates of Olympus to open on their own or the bellows of the forge to work autonomously. The materials such as gold and bronze used to make these artificial creatures offer them immense resistance and even immortality. These automatons are there to serve the gods and to perform tedious, repetitive, and daunting tasks to perfection by surpassing mortals. No one can escape the dog forged by Hephaestus, and no one can circumnavigate Crete three times a day as Talos does. The human dream may have found its origins here. In the time of Kronos, humans lived with the gods and led a life without suffering, without pain, and without work, because nature produced abundantly without effort. All you had to do was stretch out your arm to pick up the fruit. The young golden servants "perfectly embody the wealth, the beauty, the strength, the vitality of this bygone golden age for humans" (J.W. Alexandre Marcinkowski). This perfect world, without slavery, without thankless tasks, where humans do not experience fatigue and can dedicate themselves to noble causes, was taken up by certain philosophers including Aristotle, who in a famous passage from *Politics* sees in artificial creatures an advantage that is certain:

> If every tool, when ordered, or even of its own accord, could do the work that benefits it... then there would be no need either of apprentices for the master workers or of slaves for the lords.
>
> *Aristotle,* Politics

We can see in this citation one of the first definitions of AI. Hephaestus does not imitate the living but rather creates it, which is different from imitation. Blacksmith automatons have intelligence, voice, and strength. His creations do not equal the gods, who are immortal and impossible to equal. This difference shows a hierarchy between the gods and those living automatons who are their subordinates. The latter are also superior to humans when considering the perfection of the tasks that are performed simply, without defects or deception. This superiority is not entirely accurate in the sense that some humans have shown themselves to be more intelligent than automatons to achieve their ends. We can cite Medea's overcoming of Talos. Hephaestus is the only deity capable of creating these wondrous creatures. But these myths lay the foundations of the relationship between humans and technology. Hephaestus is inspired by nature, living beings, and the world. He makes models that do not threaten the mortal world. These creatures are even prehumans if we think of Pandora. In the Hellenistic period, machines were created, thanks to scientists and engineers such as Philo of Byzantium or Heron of Alexandria. We have seen the appearance of automatic doors that open by themselves at the sound of a trumpet, an automatic water dispenser, and a machine using the contraction of air or its rarefaction to operate a clock. Many automatons are also described in the *Pneumatika* and *Automaton-Making* by Héron. These automatons amaze but are not considered to produce things industrially and have no economic or societal impact; these machines make shows. At that time, there was likely no doubt that we could perhaps imitate nature and provide the illusion but surely not match it, unlike Hephaestus who instead competes with nature. The works of Hephaestus are perfect, immortal, and capable of "engendering offspring." When his creatures leave Olympus and join humans, they degenerate and die. Hephaestus, unlike humans, does not imitate the living but instead manufactures it. Despite thousands of years of stories, myths, attempts, and discoveries, we are still far from Hephaestus.

Nevertheless, our understanding has evolved. We have known for a century that our brain runs on fuel, oxygen, and glucose. It also works with electricity since neurons transmit what they have to transmit, thanks to electrical phenomena, using what are called action potentials. Electricity is something we can model. In his time, Galileo said that "nature is a book written in mathematical language." So, can we seriously consider the creation of a human brain, thanks to mathematics? To imagine programming or simulating thought, you must first understand it, take it apart, and break it down. To encode a reasoning process, you must first be able to decode it. The analysis of this process, or the desire for analysis in any case, has existed for a very long time.

The concepts of modern computing have their origins in a time when mathematics and logic were two unrelated subjects. Logic was notably developed, thanks to two philosophers, Plato and Aristotle. We do not necessarily make the connection, but without Plato, Aristotle, or Galileo, we might not have seen IBM, Microsoft, Amazon, or Google. Mathematics and logic are the basis of computer science. When AI began to develop, it was assumed that the functioning of thought could be mechanized. The study of the mechanization of thought or reasoning has a very long history, as Chinese, Indian, and Greek philosophers had already developed formal deduction methods during the first millennium BCE. Aristotle developed the formal analysis of what is called the syllogism:

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal

This looks like an algorithm. Euclid, around 300 BCE. J.-C., subsequently wrote the *Elements,* which develops a formal model of reasoning. Al-Khwārizmi (born around 780 CE) developed algebra and gave the algorithm its name. Moving forward several centuries, in the seventeenth century the philosophers Leibniz, Hobbes, and Descartes explored the possibility that all rational thought could be systematically translated into algebra or geometry. In 1936, Alan Turing laid down the basic principles of computation. This was also a time when mathematicians and logicians worked together and gave birth to the first machines.

In 1956, the expression "artificial intelligence" was born during a conference at the Dartmouth College in the United States. Although computers at the time were being used primarily for what was called scientific computing, researchers John McCarthy and Marvin Minsky used computers for more than just computing; they had big ambitions with AI. Three years later, they opened the first AI laboratory at MIT. There was considerable investment, great ambitions, and a lot of unrealized hope at the time. Among the promises? Build a computer that can mimic the human brain. These promises have not been kept to this day despite some advances: Garry Kasparov was beaten in chess by the IBM Deep Blue machine, IBM's Watson AI system defeated the greatest players in the game *Jeopardy!,* and AlphaGo beat the greatest players in the board game Go by learning without human intervention. Demis Hassabis, whose goal was to create the best Go player, created AlphaGo. We learned that we were very bad players, contrary to what we had thought. The game of Go was considered at that time to be impregnable. In October 2015, AlphaGo became the first program to defeat a professional (the French player Fan Hui). In March 2016, AlphaGo beat Lee Sedol, one of the best players in the world (ninth dan professional). In May 2017, it defeated world champion Ke Jie.

These are accomplishments. But there are still important differences between human and machine. A machine today can perform more than 200 million billion operations per second, and it is progressing. By the time you read this book, this figure will surely have already been exceeded. On the other hand, if there is a fire in the room, Kasparov will take to his heels while the machine will continue to play chess! Machines are not aware of themselves, and this is important to mention. AI is a tool that can help us search for information, identify patterns, and process natural language. It is machine learning that allows elimination of bias or detection of weak signals. The human component involves common sense, morality, creativity, imagination, compassion, abstraction, dilemmas, dreams, generalization, relationships, friendship, and love.

Machine Learning

In this book, we are not going to philosophize but we will explore how to apply machine learning concretely. Machine learning is a subfield of AI that aims to understand the structure of data and fit that data into models that we can use for different applications. Since the optimism of the 1950s, smaller subsets of AI such as machine learning, followed by deep learning, have created much more concrete applications and larger disruptions in our economies and lives.

Machine learning is a very active field, and some considerations are important to keep in mind. This technology is used anywhere from automating tasks to providing intelligent insights. It concerns every industry, and you are almost certainly using AI applications without knowing it. We can make predictions, recognize images and speech, perform medical diagnoses, devise intelligent supply chains, and much more. In this book, we will explore the common machine learning methods. The reader is expected to understand basic Python programming and libraries such as NumPy or Pandas. We will study how to prepare data before feeding the models by showing the math and the code using well-known open-source frameworks. We will also learn how to run these models on not only classical computers (CPU- or GPU-based) but also quantum computers. We will also learn the basic mathematical concepts behind machine learning models.

From Theory to Production

One important step in our journey to AI is how we put the models we have trained into production. The best AI companies have created data hubs to simplify access to governed, curated, and high-quality data. These data are accessible to any user who is authorized, regardless of where the data or the user is located. It is a kind of self-service architecture for data consumption. The reason we need to consider the notion of a data hub is that we are in a world of companies that have multiple public clouds, on-premises environments, private clouds, hybrid clouds, distributed clouds, and other platforms.

Understanding this world is a key differentiator for a data scientist. This is what we call data governance, which is critical to an organization if they really want to benefit from AI. How much time do we spend retrieving data?

Another important topic regarding AI is how we ensure that the models we develop are trustworthy. As humans and AI systems are increasingly working together, it is essential that we trust the output of these systems. As scientists or engineers, we need to work on defining the dimensions of trusted AI, outlining diverse approaches to achieve the different dimensions, and determining how to integrate them throughout the entire lifecycle of an AI application.

The topic of AI ethics has garnered broad interest from the media, industry, academia, and government. An AI system itself is not biased per se but simply learns from whatever the data teaches it. As an example of apparent bias, recent research has shown significantly higher error rates in image classification for dark-skinned females than for men or other skin tones. When we write a line of code, it is our duty and our responsibility to make sure that unwanted bias in datasets and machine learning models does not appear and is anticipated before putting something into production. We cannot ignore that machine learning models are being used increasingly to inform high-stakes decisions about real people. Although machine learning, by its very nature, is always a form of statistical discrimination, the discrimination becomes objectionable when it places certain privileged groups at a systematic advantage and certain unprivileged groups at a systematic disadvantage. Bias in training data, due to either prejudice in labels or under- or over-sampling, yields models with undesirable results.

I believe that most people are increasingly interested in rights in the workplace, access to health care and education, and economic, social, and cultural rights. I am convinced that AI can provide us with the opportunity and the choice to improve these rights. It will improve the way we perform tasks and allow us to focus on what really matters, such as human relations, and give us the freedom and time to develop our creativity and somehow, as in the past, have time to reflect. AI is already improving our customer experience. When we think of customer experience, we can consider one of the most powerful concepts that Mahatma Gandhi mentioned – the concept of Antyodaya, which means focusing on the benefits for the very last person in a line or a company. When you have to make choices, you must always ask yourself what impact it has on the very last person. So, how are our decisions on AI or our lines of code going to affect a young patient in a hospital or a young girl in school? How will AI affect the end user? The main point is about how technology can improve the customer experience. AI is a technology that will improve our experiences, and I believe it will help us focus on improving humanity and give us time to develop our creativity.

AI can truly help us manage knowledge. As an example, roughly 160,000 cancer studies are published every year. The amount of information available in the world is so vast in quantity that a human cannot process this information. If we take 15 minutes to read a research paper, we will need 40,000 hours a year to read 160,000 research papers; we only have 8,760 hours in a year. Would we not want each of our doctors to be able to take advantage of this knowledge and more to learn about us and how to help us stay healthy or help us deal with illnesses? Cognitive systems can be trained by top doctors and read enormous amounts of information such as medical notes, MRIs, and scientific research in seconds and improve research and development by analyzing millions of papers not only from a specific field but also from all related areas and new ways to treat patients. We can use and share these trained systems to provide access to care for all populations.

This general introduction has aimed to clarify the potential of AI, but if you are reading these lines, it is certainly because you are already convinced. The real purpose of this book is to introduce you to the world of machine learning by explaining the main mathematical concepts and applying them to real-world data. Therefore, we will use Python and the most often-used open-source libraries. We will learn several concepts such as feature rescaling, feature extraction, and feature selection. We will explore the different ways to manipulate data such as handling missing data, analyzing categorical data, or processing time-related data. After the study of the different preprocessing strategies, we will approach the most often-used machine learning algorithms such as support vector machine or neural networks and see them run on classical (CPU- and GPU-based) or quantum computers. Finally, an important goal of this book is to apply our models into production in real life through application programming interfaces (APIs) and containerized applications by using Kubernetes and OpenShift as well as integration through machine learning operations (MLOps).

I would like to take the opportunity here to say a warm thank you to all the data scientists around the world who have openly shared their knowledge or code through blogs or data science platforms, as well as the open-source community that has allowed us to improve our knowledge in the field. We are grateful to have all these communities – there is always someone who has written about something we need or face.

This book was written with the idea to always have nearby a book that I can open when I need to refresh some machine learning concepts and reuse some code. All code is available online and the links are provided.

I hope you will enjoy reading this book, and any feedback to improve it is most welcome!

1

Concepts, Libraries, and Essential Tools in Machine Learning and Deep Learning

Photo by Annamária Borsos

In this first chapter, we will explore the different concepts in statistical learning as well as popular open-source libraries and tools. This chapter will serve as an introduction to the field of machine learning for those with a basic mathematical background and software development skills. In general, machine learning is used to understand the structure of the data we have at our disposal and fit that data into models to be used for anything from automating tasks to providing intelligent insights to predicting a behavior. As we will see, machine learning differs from traditional computational approaches, as we will train our algorithms on our data as opposed to explicitly coded instructions and we will use the output to automate decision-making processes based on the data we have provided.

Machine learning, deep learning, and neural networks are branches of artificial intelligence (AI) and computer science. Specifically, deep learning is a subfield of machine learning, and neural networks are a subfield of deep learning. Deep learning is more focused on feature extraction to enable the use of large datasets. Unlike machine learning, deep learning does not require human intervention to process data.

Exploration and iterations are necessary for machine learning, and the process is composed of different steps. A typical machine learning workflow is summarized in Figure 1.1.

In this chapter, we will see concepts that are applied in machine learning, including unsupervised methods such as clustering to group unlabeled data such as K-means or dimensionality reduction to reduce the number of features in order to better summarize and visualize the data such as principal component analysis, feature extraction to define attributes in image and text data, feature selection to identify meaningful features to create better supervised models, and cross-validation to estimate the performance of models on new data or ensemble methods to combine the predictions of multiple models.

Machine Learning Theory and Applications: Hands-on Use Cases with Python on Classical and Quantum Machines, First Edition. Xavier Vasques.

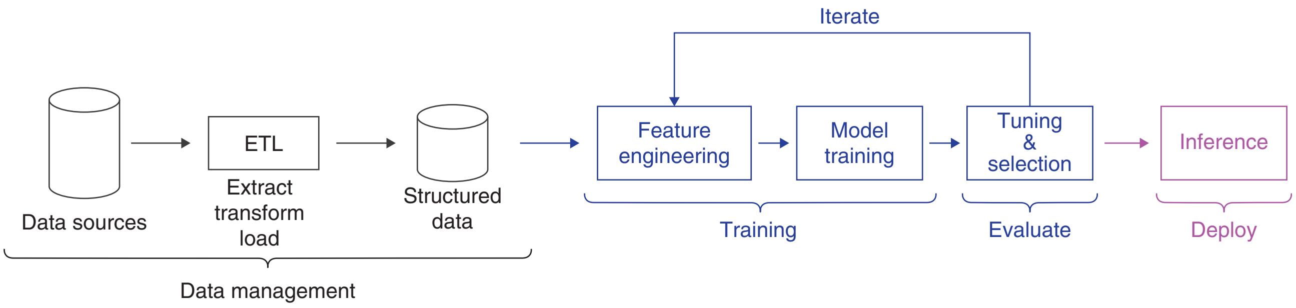


Figure 1.1 An example of a machine learning workflow.

1.1 Learning Styles for Machine Learning

The literature describes different types of machine learning algorithms classified into categories. Depending on the way we provide information to the learning system or on whether we provide feedback on the learning, these types fall into categories of supervised learning, unsupervised learning, or reinforcement learning.

1.1.1 Supervised Learning

Supervised learning algorithms are by far the most widely adopted methods. The algorithms are based on labeled and organized data for training. For example, you can feed a machine learning algorithm with images labeled as “dog” or “cat.” After training on these data, the algorithm should be able to identify unlabeled “dog” and “cat” images and recognize which image is a dog and which image is a cat. The main concept for the algorithm is to be able to “learn” by comparing its actual output with “taught” outputs. Supervised learning involves identifying patterns in data to predict label values on additional unlabeled data. In other words, a supervised machine learning algorithm suggests that the expected answer in upcoming data has already been identified in a historical dataset containing the correct answers.

There are many examples in which supervised learning can be applied, including using historical data to predict future events such as upcoming stock market fluctuations, spam detection, bioinformatics, or object recognition. Supervised learning algorithms have some challenges such as the preprocessing of data and the need for regular updates.

Mathematically, we start with some dataset:

$$D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subseteq \mathcal{R}^d \times C$$

where each x_i is a d-dimensional feature vector of the ith example, y_i the label of the ith example, n the size of the dataset, C the label space and $\mathcal{R}^d$ the d-dimensional feature space. We assume that the data points have been drawn from some unknown distribution $(x_i, y_i) \sim \wp$ with (x_i, y_i) to be independent and identically distributed (iid). The objective of supervised machine learning is to find a function $f : \mathcal{R}^d \to C$ that for every new input or output (x, y) sampled from $\wp$ we have $f(x) \approx y$.

Usually, the data that are being processed do not represent images by itself but rather metadata associated with it such as the well-known Iris dataset (https://archive.ics.uci.edu/ml/datasets/Iris), which provides different iris features (petal width and length, sepal width and length) as inputs and the classes of iris (Setosa, Versicolour, Virginica) as outputs.

When considering supervised learning methods, we need to understand the feature and label spaces that we have. Usually, supervised learning operates with the following label spaces:

- **Binary classification:** The algorithm will classify the data into two categories such as spam or not spam. The label space is $\{0, 1\}$ or $\{-1, 1\}$.
- **Multi-class classification:** The algorithm needs to choose among more than two types of answers for a target variable such as the recognition of images containing animals (e.g., dog = 1, cat = 2, fish = 3, etc.). If we have N image classes, we have $C = \{1, 2, \ldots, N\}$.
- **Regression:** Regression models predict continuous variables as opposed to classification models, which consider categorical variables. For example, if we attempt to predict measures such as temperature, net profit, or the weight of a person, this will require regression models. Here, $C = \mathcal{R}$.

In supervised learning, we can find popular classification models such as decision trees, support vector machine, naïve Bayes classifiers, random forest, or neural networks. We can also find popular regression models such as linear regression, ridge regression, ordinary least squares regression, or stepwise regression.

As mentioned above, we need to find a function $f : \mathcal{R}^d \rightarrow C$. This requires some steps such as making some assumptions regarding what the function f looks like and what space of functions we will be using (linear, decision trees, polynomials, etc.). This is what we call the hypothesis space $\mathcal{H}$. This choice is very important because it impacts how our model will generalize to completely new data that has not been used for training.

1.1.1.1 Overfitting and Underfitting

The largest challenge in supervised learning is effective generalization, which means the ability of a machine learning model to provide a suitable output by adapting to the given set of unknown input. In other words, the question is how to ensure our model will perform well on new data after training (Figure 1.2). Given our dataset D, we can define the function f as follows:

$$f(x) = \begin{cases} y_i, & \text{if there exists } (x_i, y_i) \text{ such that } x = x_i \\ 0, & \text{otherwise} \end{cases}$$

We can see that the function would perform perfectly with our training data, but if something new is introduced, the results would certainly be wrong. Underfitting and overfitting are two things we need to control to make the performance of the model stable and to determine whether the model is generalizing well. There is a trade-off that we need to accept

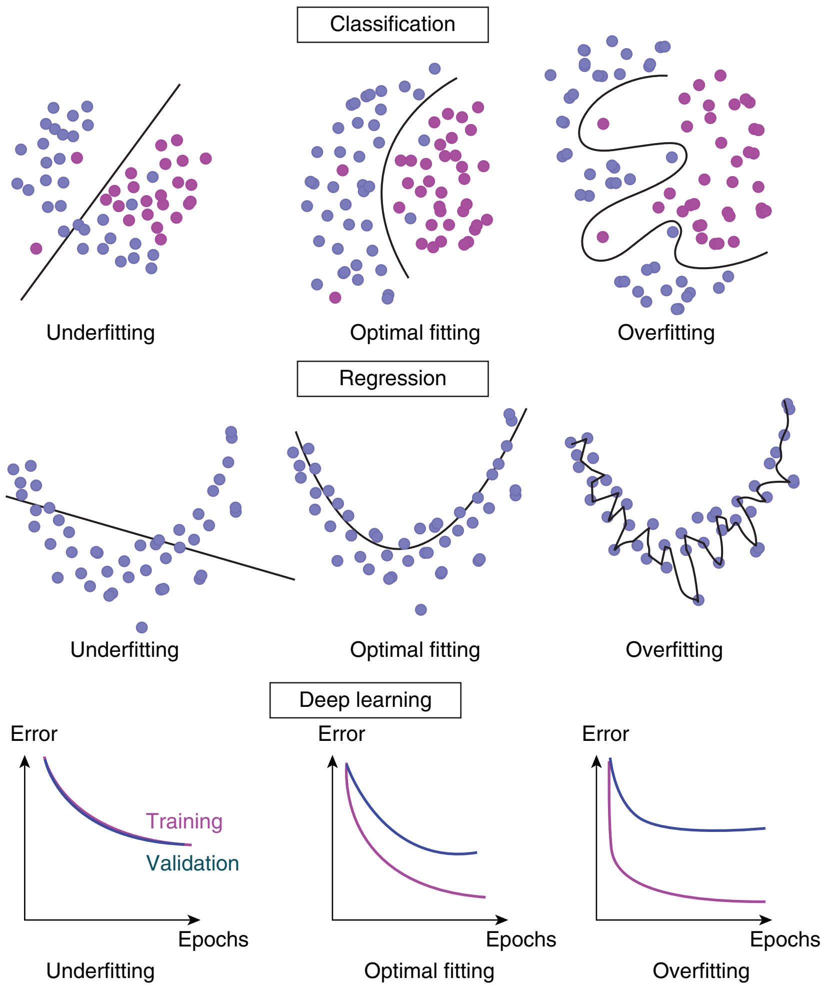


Figure 1.2 Underfitting, optimal fitting, and overfitting in classification, regression, and deep learning.

between underfitting or bias and overfitting or variance. Bias means a prediction error that is introduced in the algorithm due to oversimplification or the differences between the predicted values and the actual values. Variance occurs when the model performs well with training data but not with test data. We should also introduce two other words: signal and noise. Signal refers to the true underlying pattern of the data that allows the algorithm to learn from data, whereas noise is irrelevant and unnecessary data that reduces the performance of the algorithm.

In **overfitting,** the machine learning model will attempt to cover more than the necessary data points present in a dataset or simply all data points. It is a modeling error in statistics because the function is too closely aligned to the training dataset and will certainly translate to a reduction in efficiency and accuracy with new data because the model has considered inaccurate values in the dataset. Models that are overfitted have low bias and high variance. For example, overfitting can come when we train our model excessively. To reduce overfitting, we can perform cross-validation, regularization, or ensembling, train our model with more data, remove unnecessary features, or stop the training of the model earlier. We will see all these techniques.

On the opposite, we have **underfitting,** with high bias and low variance, which occurs when our model is not able to capture the underlying trend of the data, generating a high error rate on both the training dataset and new data. An underfitted model is not able to capture the relationships between input and output variables accurately and produces unreliable predictions. In other words, it does not generalize well to new data. To avoid underfitting, we can decrease the regularization used to reduce the variance, increase the duration of the training, and perform feature selection.

Some models are more prone to overfitting than others such as KNN or decision trees. The goal of machine learning is to achieve "goodness of fit," which is a term based on the statistics that define how closely the results or predicted values match the true values of the dataset. As we can imagine, the ideal fit of our model is between underfitting and overfitting making predictions with zero errors. As we will see during our journey to mastering machine learning, this goal is difficult to achieve. In addition, overfitting can be more difficult to identify than underfitting because the training data perform at high accuracy. We can assess the accuracy of our model by using a method called k-folds cross-validation.

1.1.1.2 K-Folds Cross-Validation

Cross-validation consists of a resampling procedure to assess machine learning models on a limited data sample. Cross-validation has a parameter called k that refers to the number of groups into which a given dataset is to be split. Therefore, in k-folds cross-validation, data are split into k equally sized subsets (folds). When we define a specific value for k, for instance, $k = 10$, k-fold cross-validation becomes tenfold cross-validation. When the data are separated into k-folds, one of the k-folds is considered as a test set (holdout set or validation set), and the rest of the folds serve to train the model. This process is repeated until each of the folds has acted as validation test. Each repetition comes with an evaluation. When all iterations have been completed, the scores are averaged, providing a general assessment of the performance of the model. It is a popular method because of its simplicity and robustness, as it generally results in a less biased or less optimistic estimate of models than other methods such as train/test split.

As mentioned above, the general procedure consists of shuffling the dataset randomly and splitting the dataset into k groups. For each group, we consider an initial group as test dataset and the rest as training dataset; we fit a model on the training set and evaluate it with the test dataset. We then retain the evaluation score, discard the model, and repeat the steps with other groups. In the end, we average the scores to evaluate the performance of the global model. The technique allows each sample to be used in the test set one time and used to train the model $(k - 1)$ times. Cross-validation can have some drawbacks, for example, when processing data evolving over a period or inconsistent data. To illustrate the method, let's say we divide a dataset into five subgroups as in Figure 1.3.

1.1.1.3 Train/Test Split

A general approach in machine learning is to split a dataset D into three sets : training D_{TR}, validation D_{VA}, and test D_{TE}. We can consider an effective split as approximately 80% of the data for training and 20% for validation (10%) and testing (10%). Of course, this split depends on the amount of data and context.

The split is performed at the very beginning of the machine learning process when we start our search for the function $f : \mathcal{R}^d \rightarrow C$ described above. The training dataset helps to find $f : \mathcal{R}^d \rightarrow C$, and the validation set helps evaluate a given model. The split is used to fine-tune the model hyperparameters by providing an unbiased evaluation of a model fit on

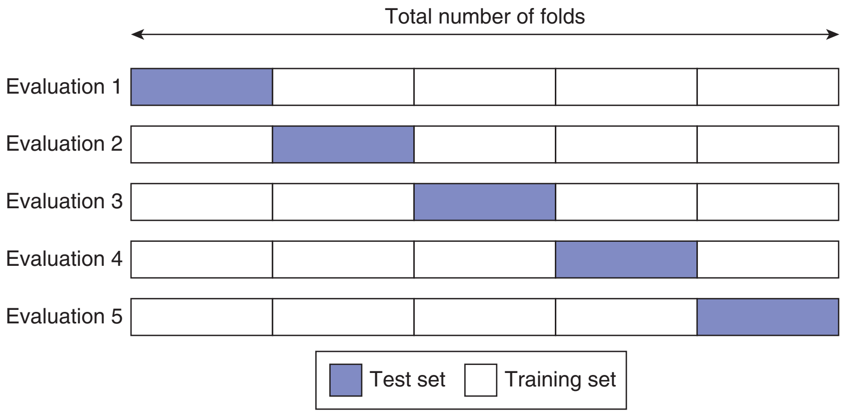


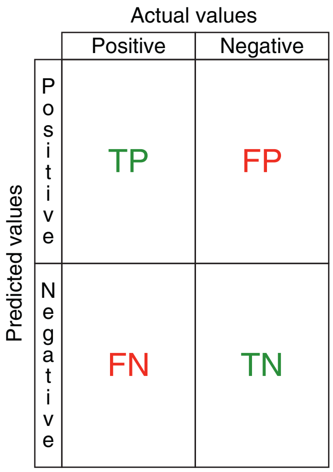
Figure 1.3 Dataset divided into five subgroups for fivefold cross-validation.

the training dataset. The more the validation dataset is incorporated into the model configuration, the more the evaluation becomes biased. The test dataset is used to provide an unbiased evaluation of a final model; it is only used when the model has been completely trained using training and validation datasets. It is important to carefully curate the test dataset in order to represent the real world.

1.1.1.4 Confusion Matrix

The confusion matrix is another important concept in machine learning used to determine the performance of a classification model for a given set of test data. It shows the errors in the form of an $N \times N$ matrix (error matrix) where N is the number of target classes. It compares the actual target values with those predicted by the machine learning algorithm. For example, for a binary classification problem, we will have a 2×2 matrix; for three classes, we will have a 3×3 table; and so on.

In Figure 1.4, the matrix contains two dimensions, representing predicted values and actual values, along with the total number of predictions. The target variable has two values (positive or negative); the columns represent the actual values and the rows the predicted values of the target variable. Inside the matrix, we have the following possibilities:

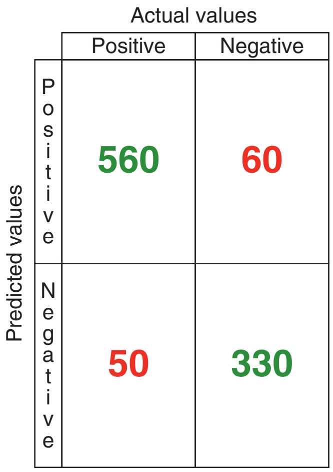


		Actual values	
		Positive	Negative
Predicted values	Positive	TP	FP
	Negative	FN	TN

Figure 1.4 Confusion matrix.

- **True positive (TP):** The actual value was positive, and the model predicted a positive value.
- **True negative (TN):** The actual value was negative, and the model predicted a negative value.
- **False positive (FP):** The actual value was negative, but the model predicted a positive value (the predicted value was falsely predicted, also known as type 1 error).
- **False negative (FN):** The actual value was positive, but the model predicted a negative value (the predicted value was falsely predicted, also known as type 2 error).

Let us take an example (Figure 1.5) with a classification dataset of 1000 data points with a fitted classifier that has provided the confusion matrix shown. In Figure 1.5, we see that 560 positive-class data points were correctly classified by the model, 330 negative-class data points were correctly classified, 60 negative-class data points were not correctly classified as belonging to the positive class, and 50 positive-class data points were not classified correctly as belonging to the negative class by the model. We can conclude that the classifier is acceptable.



		Actual values	
		Positive	Negative
Predicted values	Positive	560	60
	Negative	50	330

Figure 1.5 An example of a confusion matrix.

To make things more concrete visually, let us say we have a set of 10 people and we want to build a model to predict whether they are sick. Depending on the predicted value, the outcome can be TP, TN, FP, or FN:

ID	Actual sick	Predicted sick	Confusion matrix
1	1	1	TP
2	1	1	TP
3	1	0	FP
4	0	1	FN
5	0	1	FN
6	0	0	TN
7	1	1	TP
8	0	0	TN
9	1	0	FP
10	0	1	FN

With the help of a confusion matrix, it is possible to calculate a measure of performance such as accuracy, precision, recall, or others.

Classification Accuracy Classification accuracy is one of the most important parameters to assess the accuracy of a classification problem. It simply indicates how often a model predicts the correct output. It is calculated as the ratio of the number of correct predictions to the total number of predictions made by the classifiers:

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

Misclassification Rate The misclassification rate, also known as error rate, defines how often the model provides incorrect predictions. To calculate the error rate, we compute the number of incorrect predictions to the total number of predictions made by the classification model:

$$Error\ rate = \frac{FP + FN}{TP + TN + FP + FN}$$

Precision and Recall Precision is the number of correct outputs, or how many of the predicted positive cases were truly positive. This is a way to estimate whether our model is reliable.

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

Recall defines how our model predicted correctly out of total positive classes. In other words, how many of the actual positive cases were predicted correctly:

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

F-Score The traditional F-measure or balanced F-score (F_1 score) is used to evaluate recall and precision at the same time. It is a harmonic mean of precision and recall. The F-score is highest when precision and recall are equal.

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

The F-score is widely used in the natural language processing literature (name entity recognition, word segmentation, etc.).

The metrics described above are the most widely used ones. There are a number of other important metrics that we can explore to fit our context, including the Fowlkes–Mallows index, Matthews correlation coefficient, Jaccard index, diagnostic odds ratio, and others. We can find all these metrics in the literature regarding diagnostic testing in machine learning.

To provide a simple example of code, let us say that we wish to build a supervised machine learning model called linear discriminant analysis (LDA) and print the classification accuracy, precision, recall, and F_1 score. Before doing this, we have previously split the dataset into training data (X_train and y_train) and testing data (X_test and y_test). We want to apply cross-validation ($k = 5$) and print the scores. To run this code, we need to install scikit-learn, as described later in this chapter.

Input:

```
# Importing required libraries
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.model_selection import cross_val_score

# Define a function that performs linear discriminant analysis
def lda(X_train, X_test, y_train, y_test):

 # Instantiate a LinearDiscriminantAnalysis object
 clf=LinearDiscriminantAnalysis()

 # Fit the LDA model using the training data
 clf.fit(X_train,y_train)

 # Use the trained model to make predictions on the test data
 y_pred = clf.predict(X_test)

 # Print out the results of the linear discriminant analysis
 print("\n")
 print("Linear Discriminant Analysis Metrics")
 print("Classification accuracy:", metrics.accuracy_score(y_test, y_pred))
 print("Precision:", metrics.precision_score(y_test, y_pred, average='micro'))
 print("Recall:", metrics.recall_score(y_test, y_pred, average='micro'))
 print("F1 score:", metrics.f1_score(y_test, y_pred, average='micro'))

 # Use cross-validation to estimate the performance of the model on new data
 print("Cross validation:",cross_val_score(clf, X_train, y_train, cv=5))
 print("\n")
```

1.1.1.5 Loss Functions

In supervised machine learning algorithms, the objective is to minimize the error of each training example during the learning process. The loss function $L : \mathcal{H} \to \mathbb{R}$ is a measure of how well our prediction model f can predict the expected outcome by assigning a loss to each $f \in \mathcal{H}$. Here, we simply convert the learning problem into an optimization problem with the aim of defining a loss function and then optimizing the model to minimize the loss function:

$$arg\ \min_{h\in\mathcal{H}} L(h) = arg\ \min_{h\in\mathcal{H}} \frac{1}{n}\sum_{i=1}^{n} l(x_i, y \mid h)$$

where we use the loss function L of a hypothesis h given D and l for the loss of a single data pair (x_i, y) given h. There are different types of loss functions such as squared error loss, absolute error loss, Huber loss, or hinge loss. For clarity, it is also important to mention the cost function, which is the average loss over the entire training dataset; in contrast, the loss function is computed for a single training example. The loss function will have a higher value if the predictions do not appear accurate and a lower value if the model performs fairly well. The cost function quantifies the error between predicted values and expected values and presents it in the form of a single number. The purpose of the cost function is to be either minimized (cost, loss, or error) or maximized (reward). Cost and loss refer almost to the same concept, but the cost function invokes a penalty for a number of training sets or the complete batch, whereas the loss function mainly applies to a single training set. The cost function is computed as an average of the loss functions and is calculated once; in contrast, the loss function is calculated at every instance.

For example, in linear regression, the loss function used is the squared error loss and the cost function is the mean squared error (MSE). The **squared error loss,** also known as L2 loss, is the square of the difference between the actual and the predicted values:

$$L = (y - f(x))^2$$

The corresponding cost function is the **mean of these squared** errors. The overall loss is then:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(x_i))^2$$

The MSE can be implemented as follows:

```
def MSE(y_predicted, y_actual):
    squared_error = (y_predicted - y_actual) ** 2
    sum_squared_error = np.sum(squared_error)
    mse = sum_squared_error / y_actual.size
    return mse
```

Another regression loss function is the **absolute error loss,** also known as the L1 loss, which is the difference between the predicted and the actual values, irrespective of the sign:

$$L = |y - f(x)|$$

The cost is the mean of the absolute errors (MAE), which is more robust than MSE regarding outliers.

Another example is the Huber loss, which combines the MSE and MAE by taking a quadratic form for smaller errors and a linear form otherwise:

$$Huber = \begin{cases} \frac{1}{2}(y - f(x))^2, & \text{if } |y - f(x)| \le \delta \\ \delta|y - f(x)| - \frac{1}{2}\delta^2, & \text{otherwise} \end{cases}$$

In the formula above, Huber loss is defined by a δ parameter; it is usually used in robust regression or m-estimation and is more robust to outliers than MSE.

For binary classification, we can use the **binary cross-entropy loss** (also called log loss), which is measured for a random variable X with probability distribution $p(X)$:

$$L = \begin{cases} -\int p(x) \cdot \log p(x) \cdot dx, & \text{if } x \text{ is continuous} \\ -\sum_{x} p(x) \cdot \log p(x), & \text{if } x \text{ is discrete} \end{cases}$$

If there is greater uncertainty in the distribution, the entropy for a probability distribution will be greater.

The **hinge loss** function is very popular in support vector machines (SVMs) with class labels 1 and −1:

$$L = \max(0, 1 - y * f(x))$$

Finally, for multi-class classification, we can use the **categorical cross-entropy loss,** also called softmax loss. It is a softmax activation plus a cross-entropy loss. In a multi-label classification problem, the target represents multiple classes at once. In this case, we calculate the binary cross-entropy loss for each class separately and then sum them up for the complete loss.

We can find numerous methodologies that we can explore to better optimize our models.