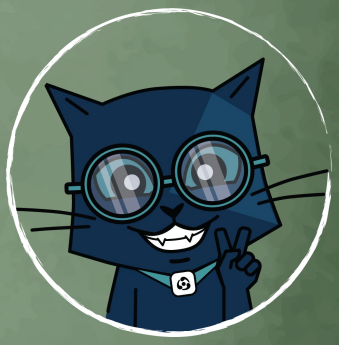
exactpro

ISTQB® AI Testing course

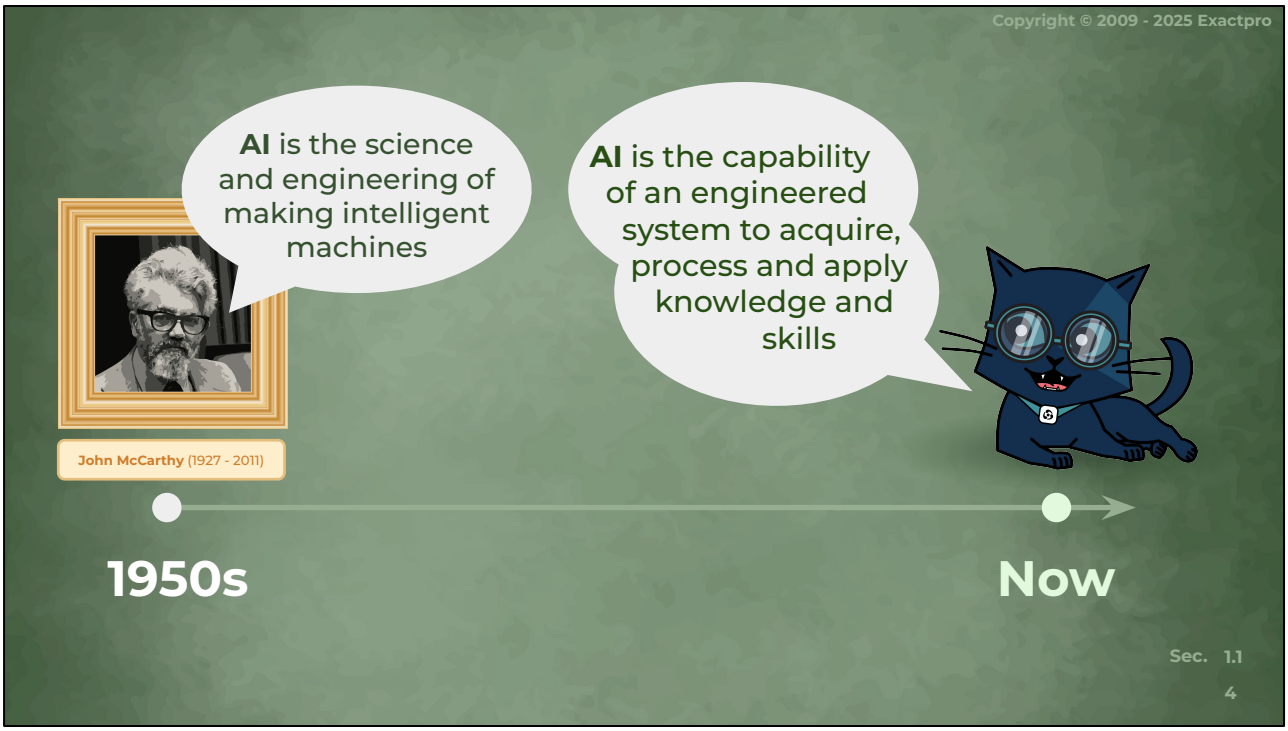
Chapter 1. Introduction to AI



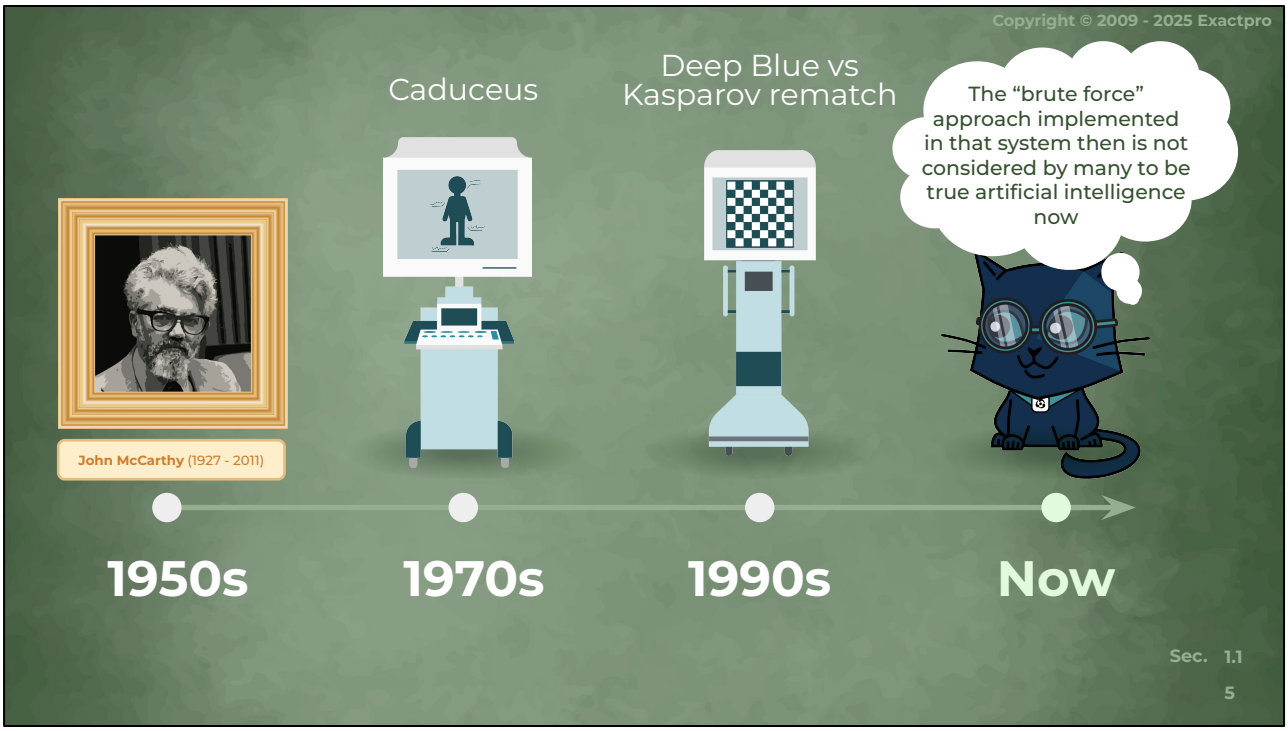
Iosif Itkin, Iuliia Emelianova,
Dmitrii Degtiarenko

Contents

1.1 Definition of AI and the AI Effect



The term artificial intelligence (AI) was coined in the 1950s by John McCarthy and refers to the objective of building and programming “intelligent” machines capable of imitating human beings. Since then, it has evolved significantly, so here's a more modern definition that captures the concept of AI: It’s the capability of an engineered system to acquire, process and apply knowledge and skills. So it’s safe to say that the way people understand the meaning of AI depends on their current world view. For example, the expert systems of the 1970s and 1980s incorporated human expertise as rules which could be run without the expert being present.



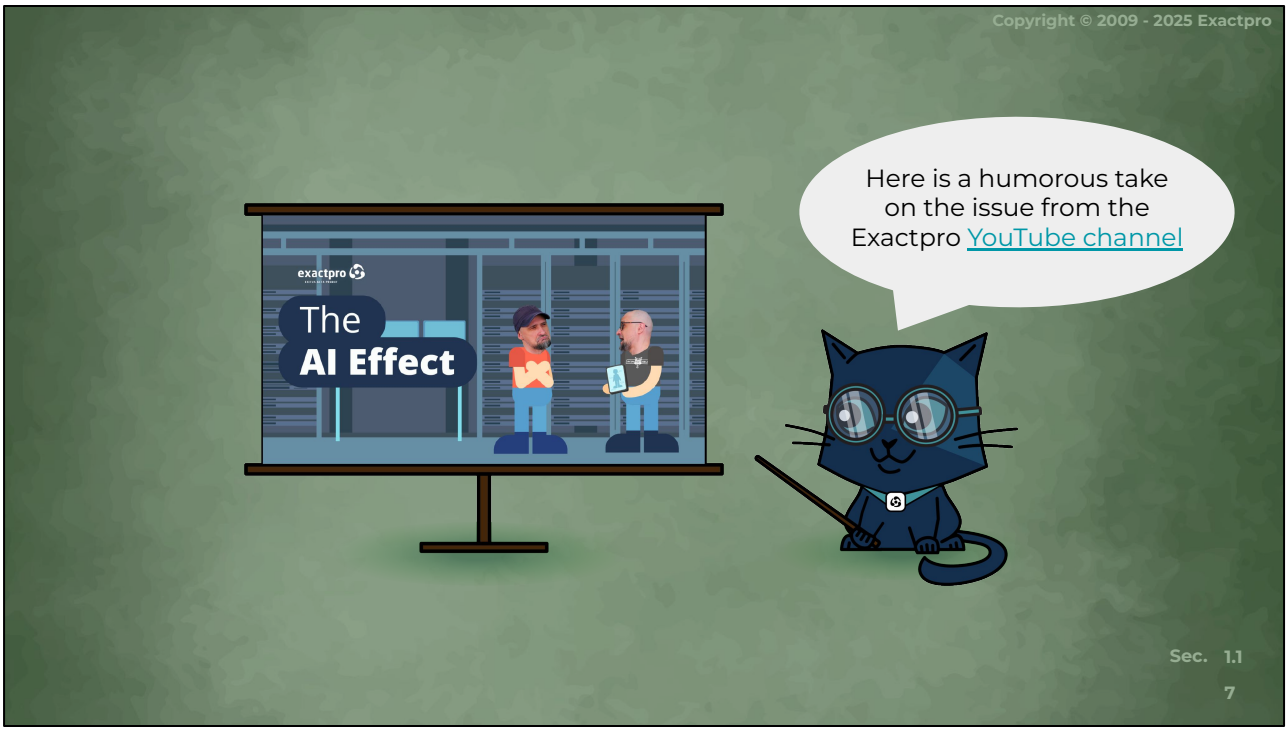
One of the most prominent examples of such machines was Caduceus, a medical machine capable of diagnosing up to 1000 different diseases. These were considered to be AI then, but are not considered as such now. Before the 1997’s Deep Blue vs Kasparov rematch, the mere idea of a computer system beating a human in a six-game match would be mind-boggling and definitely an indication that AI is a reality and then it happened. But once again the “brute force” approach implemented in that system then is not considered by many to be true artificial intelligence now. Which makes people wonder and say, ”If I beat the world’s chess champion, I’d be regarded as highly bright. But when a machine does something “intelligent”, it ceases to be regarded as intelligent”.

The changing concept of what constitutes AI is known as the **“AI Effect”**

Any definition made today is likely to change in the future and may not match those from the past

AI effect: The situation when a previously labelled AI system is no longer considered to be AI as technology advances

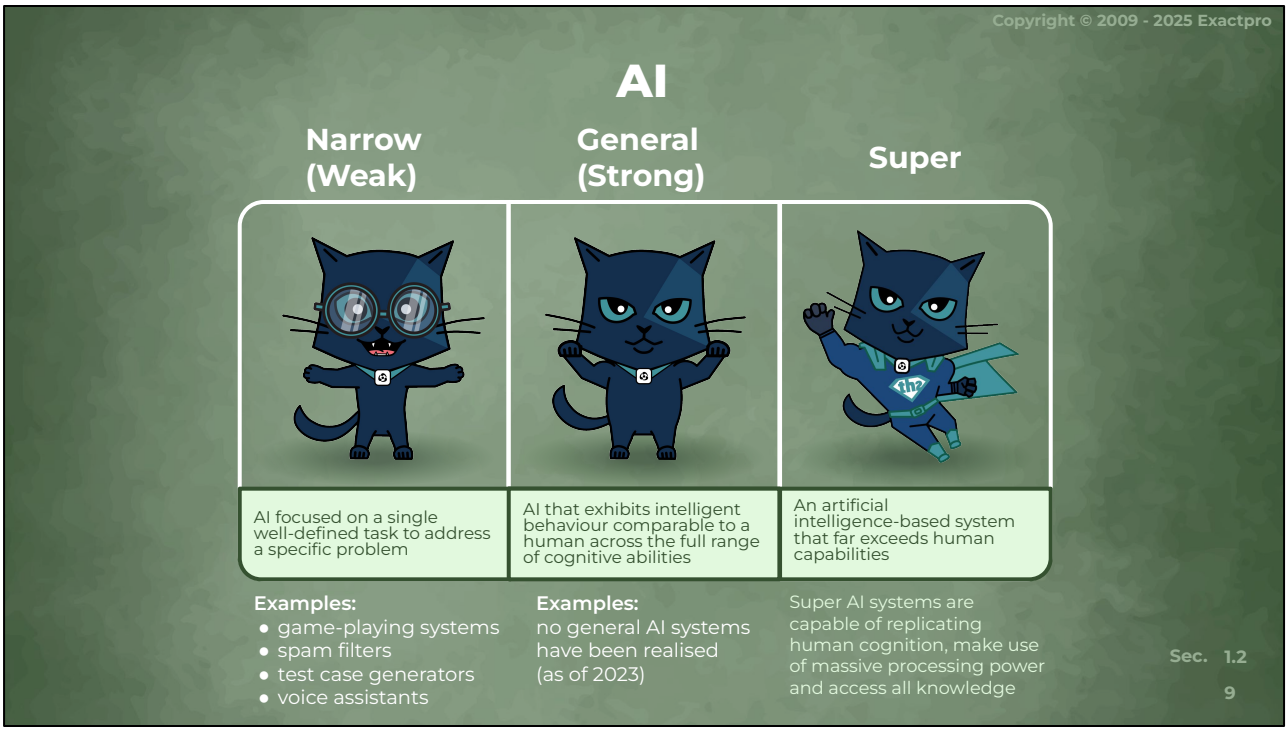
The changing concept of what constitutes AI is known as the “AI Effect”. As the perception of AI in society changes, so does its definition. It’s the same as saying that if a problem is solved using AI, the problem is no longer a part of the AI focus area. Solving an AI problem makes it lose its “mysterious” lustre and moves it from unattainable to mundane. And, as a result, any definition made today is likely to change in the future and may not match those from the past. Let’s just hope it won’t change by the time this course is over.



Here you can see a humorous take on the AI effect issue from the Exactpro YouTube channel.

1.2 Narrow, General and Super AI

It’s quite possible that a more rational way of explaining the phenomenon of an AI Effect is admitting to ourselves that what we thought could only be done with strong AI, might actually be achieved through weak AI.So let me explain what that actually means.

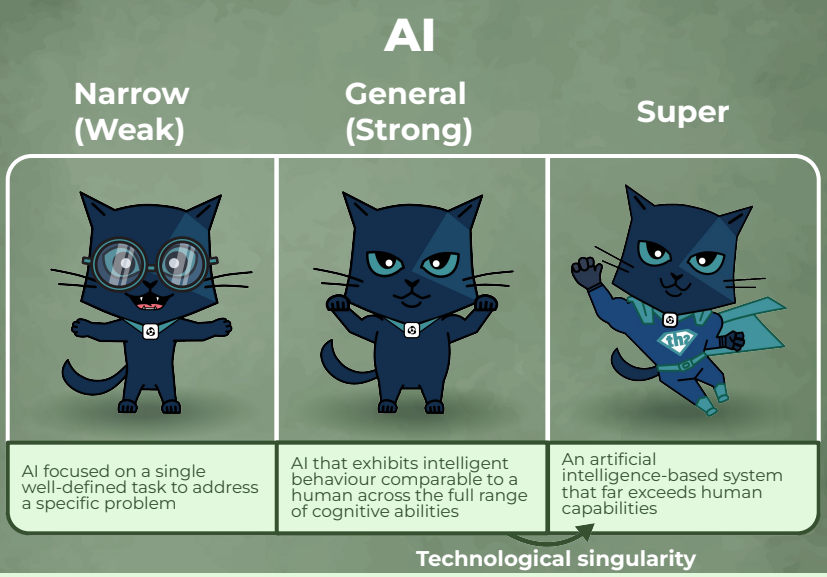


It's quite possible that a more rational way of explaining the phenomenon of an AI Effect is admitting to ourselves that what we thought could only be done with strong AI, might actually be achieved through weak AI.

So let's explain what that actually means. AI is usually broken down into three categories:

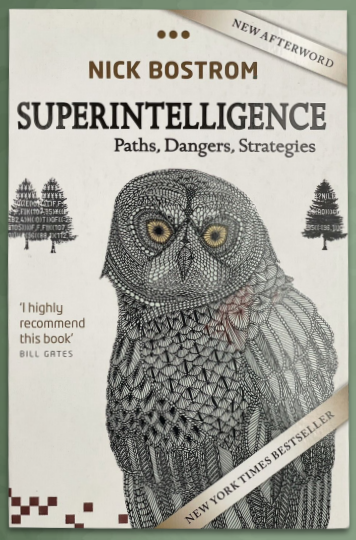
1. **Narrow AI** (or weak AI) where systems are programmed to carry out a specific task with a limited context. They can analyse and interpret data with astonishing accuracy, accomplishing it much faster than humans. They help us make better data-driven decisions, and more importantly, relieve us from monotonous tasks. Currently this form of AI is widely available. For example, game-playing systems, spam filters, self-driving cars and voice assistants. When it comes to testing, Narrow AI can assist in generating test cases, planning the overall test process as well as improving the defect report quality for QA analysts. It can also be used in intelligent capturing and text recognition without human intervention with the help of natural language processing (NLP) and sentiment analysis.
2. Next comes **General AI** (also known as strong AI or true AI), and it describes universal systems capable of performing any intellectual task

1. that humans can perform. These types of systems will also be able to reason and act based on consciousness, emotions and critical thinking. It's the holy grail for all AI researchers. But, unfortunately, no general AI systems have been realised so far, with some researchers voicing their opinion that it's not feasible at all.
2. And then there is **Super AI.** Philosopher and expert on artificial intelligence Nick Bostrom defines superintelligence as "an intelligence that vastly exceeds the cognitive capacity of human intelligence in all fields of knowledge". Here, systems are capable of replicating human cognition (or general AI), make use of massive processing power and access all knowledge, for example, via the world wide web.



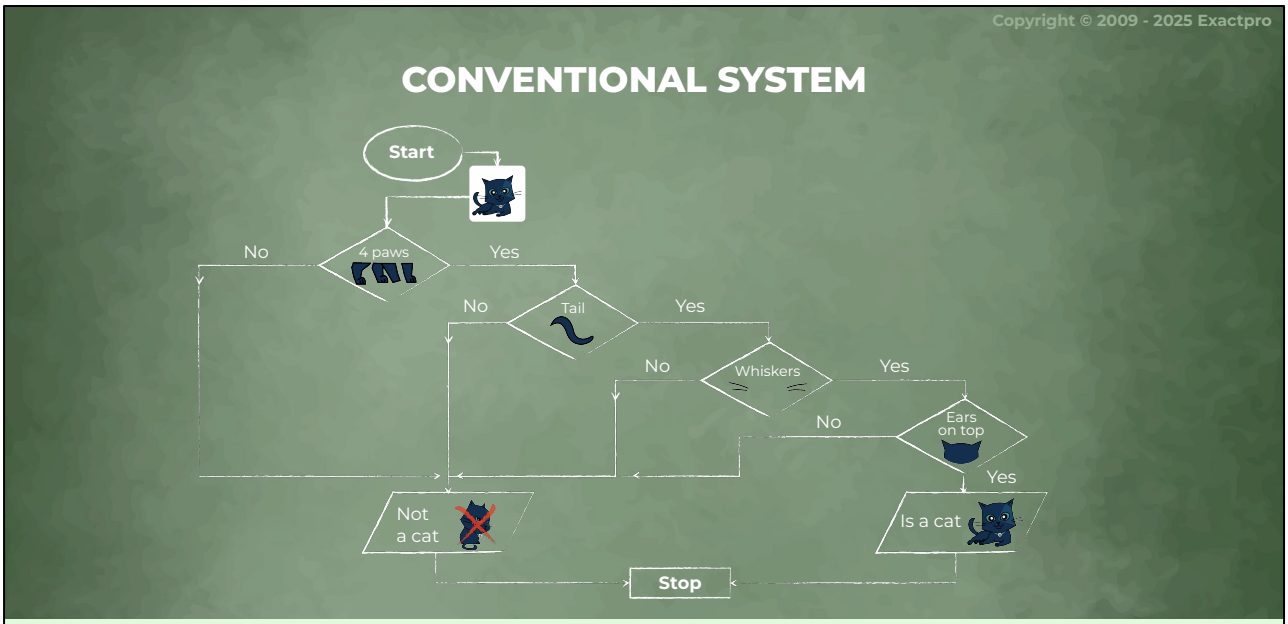
Technological singularity: A point in the future when technological advances are no longer controllable by people

The point at which AI-based systems transition from general AI to super AI is commonly known as the **technological singularity**.



If you want to learn more about the possible development of Super AI and what it might lead to, we recommend reading Bostrom's book called "Superintelligence: Paths, Dangers, Strategies".

1.3 AI-Based and Conventional System



Conventional system: In a conventional computer system, the software is programmed using constructs such as if-then-else and loops

In a conventional computer system, the software is programmed by people using constructs such as “if-then-else” and loops. It is relatively easy for humans to understand how the system transforms inputs into outputs, because everything is rule- and algorithm-based. Although this statement better reflects a trivial system rather than a complex market data juggernauts which exchange and analyse huge swaths of data.

Example

CONVENTIONAL COMPUTER SYSTEM*: In a conventional computer system, image recognition tasks are typically performed using predefined algorithms and rules. It starts with* ***Preprocessing****, where the image is prepared to enhance certain features or reduce noise using techniques like resizing, filtering, or colour adjustment. Then comes* ***Feature Extraction*** *where the system extracts predefined features from the image using techniques like edge detection, texture analysis, or colour histograms. After that we have* ***Classification****, where the extracted features are fed into a pre-trained machine learning model or an algorithm that has been manually programmed to classify or recognise specific objects. The model applies rules or comparisons to make predictions based on the extracted features. And lastly there’s* ***Decision Making****, which is based on the classification result, where the system makes a decision or takes further actions, such as identifying the object in the image or triggering a specific response.*

AI-BASED SYSTEM

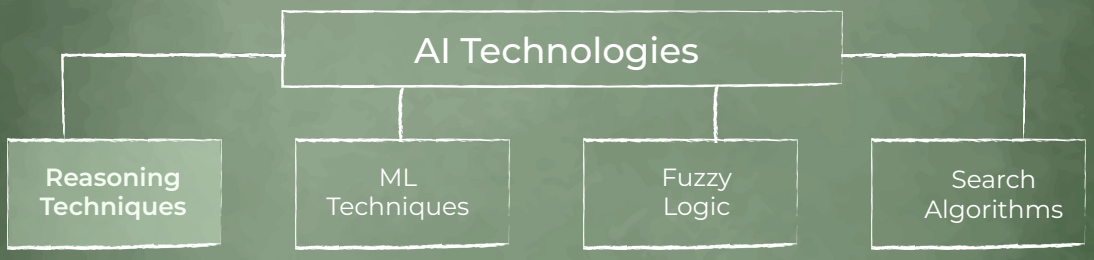
An observe-and-learn approach:
patterns in data are used to determine
how system should react to new data in the future

AI-based system: A system that integrates one or more AI components
Feature: An individual measurable attribute of the input data used for training by an ML algorithm and for prediction by an ML model

AI systems, on the other hand, have an observe-and-learn approach. Here patterns in data are used by the system to determine how it should react to new data in the future.
Take, for instance, an AI-powered image processing system specifically created to recognise images depicting cats. This system undergoes training using a curated collection of cat images. Through this training, the AI autonomously discerns the intricate patterns and distinctive features within the data that enable it to identify cats. Subsequently, these learned patterns and rules are employed to analyse new images, enabling the AI to ascertain whether or not they contain cats.
Example
AI SYSTEM*: In an AI-powered image recognition system, deep learning models and neural networks are used to automatically learn and identify patterns in images. Here it all starts with the* ***Training Phase****, where the AI system is trained on a large dataset of labelled images. Deep learning models are employed to automatically learn features and patterns from the data. Then comes* ***Feature Learning****, during which the AI system automatically learns hierarchical representations of features, detecting low-level patterns like edges and textures and gradually building up to higher-level concepts. After that there's* ***Classification and Recognition****, because once the training is complete, the AI system can classify or recognise patterns and objects in new, unseen images. It passes the image through the trained neural network, and*

the network automatically extracts relevant features and makes predictions without relying on explicitly programmed rules. Then comes the fine tuning in a form of ***Iterative Improvement*** *via increasing the size of the training dataset, or employing advanced techniques. And lastly we have* ***Adaptability and Generalisation*** *through handling variations of images with different lighting conditions, viewpoints, and object orientations.*
Compared to a conventional computer system, an AI system for image recognition utilises deep learning models and neural networks to automatically learn and identify patterns in images. It learns from labelled data, extracts features, and makes predictions without relying on explicitly defined rules. This allows the AI system to handle complex recognition tasks with improved accuracy, adaptability, and generalisation capabilities.

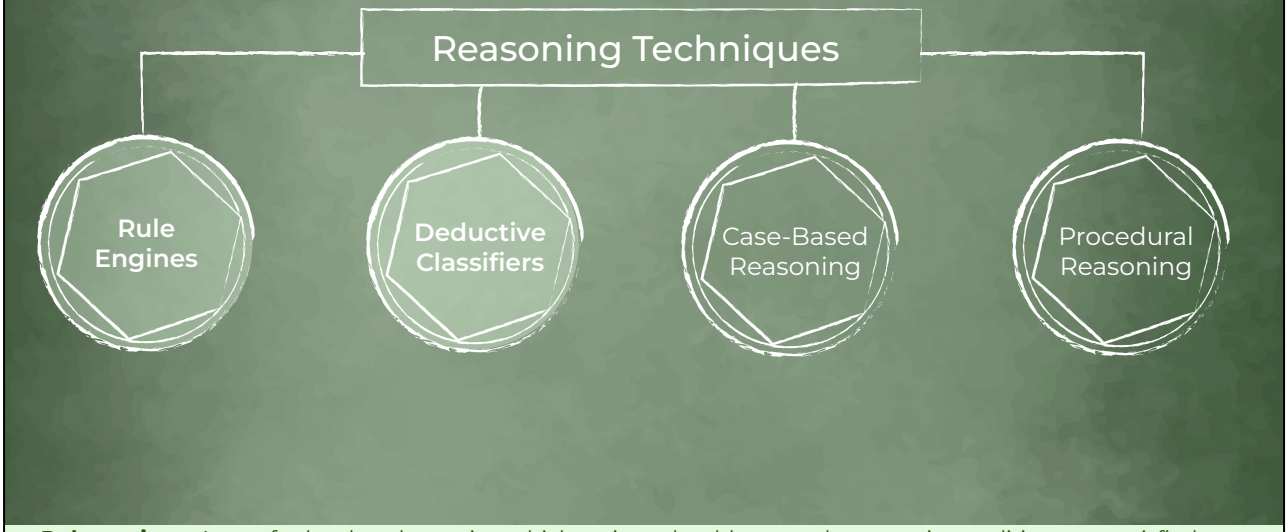
1.4 AI Technologies



Reasoning technique: AI that generates conclusions from available information using logical techniques

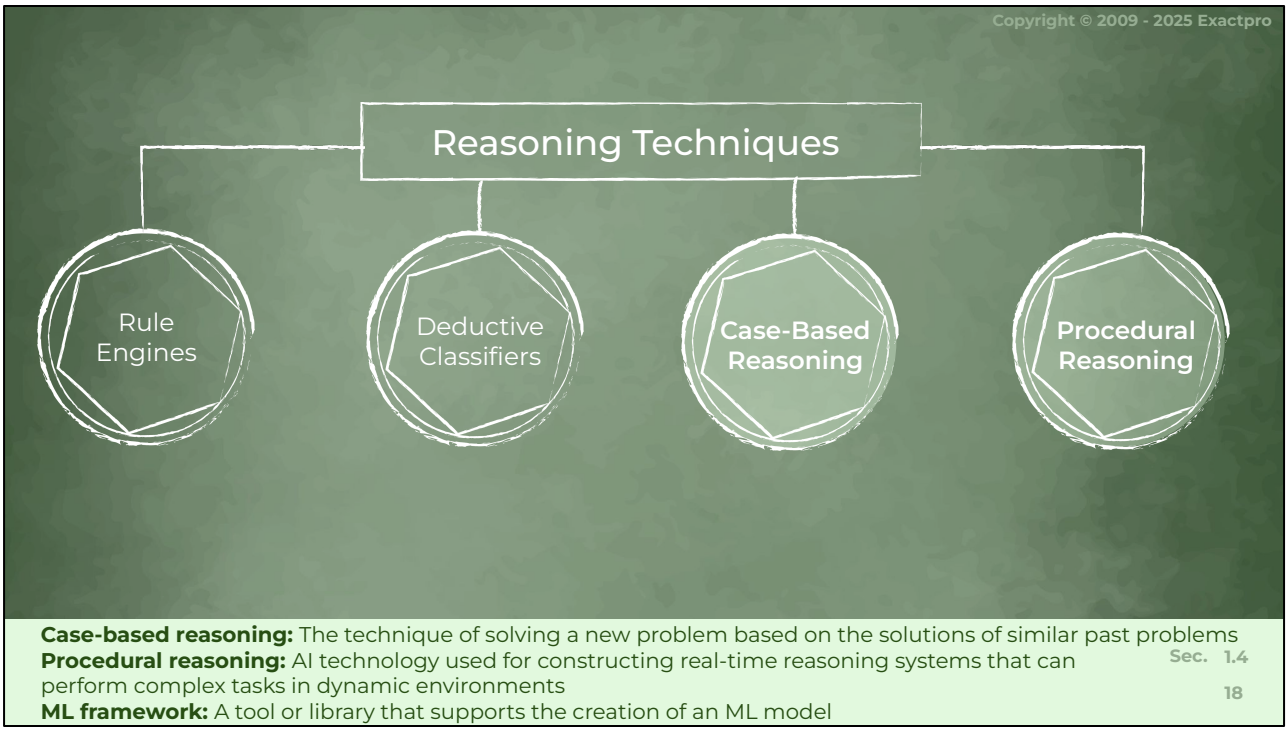
AI can be implemented using a wide range of technologies, such as:

1. **Reasoning techniques**, examples of which include:



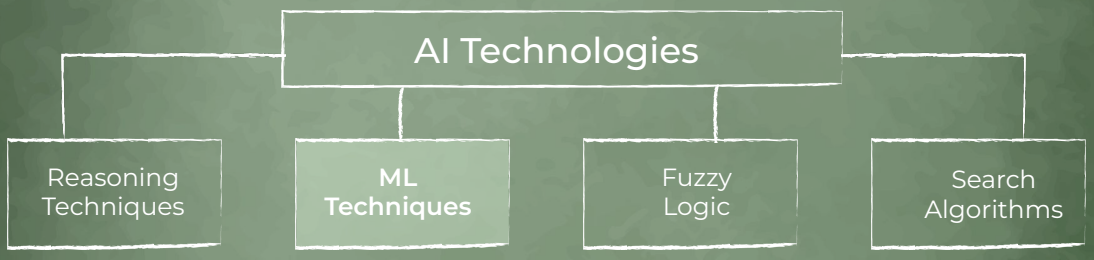
Rule engines: A set of rules that determine which actions should occur when certain conditions are satisfied
Deductive classifier: A classifier based on the application of inference and logic to input data
Classifier (classification model): An ML model used for classification
Classification: A type of ML function that predicts the output class for a given input

- **Rule engines** (or semantic reasoners) which is a set of rules that determine which actions should occur when certain conditions are satisfied. Rule engines draw logical assumptions from facts or axioms.
- **Deductive classifiers** are based on the application of inference and logic to input data. Unlike rule-based engines, which can only apply triggers like IF-THEN when a condition is not met, these classifiers seek to mimic human deductive logic to produce new information.



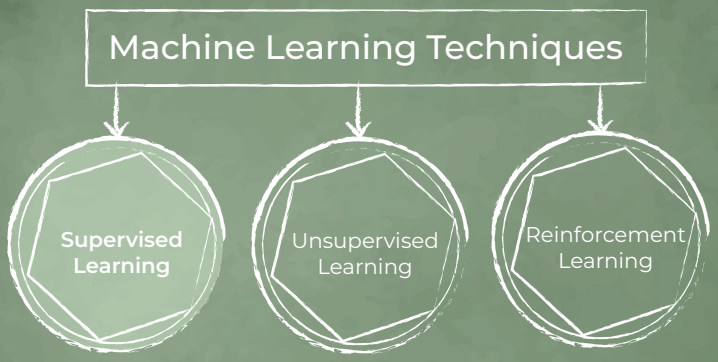
- **Case-based reasoning** which is the process of resolving new problems based on the outcomes of similar past problems.
- **Procedural Reasoning** which is used for constructing real-time reasoning systems that can perform complex tasks in dynamic environments.

They can be used to build frameworks that operate with Knowledge Areas (or predefined skills/instructions) in order to achieve a given objective



Machine learning (ML): The process using computational techniques to enable systems to learn from data or experience

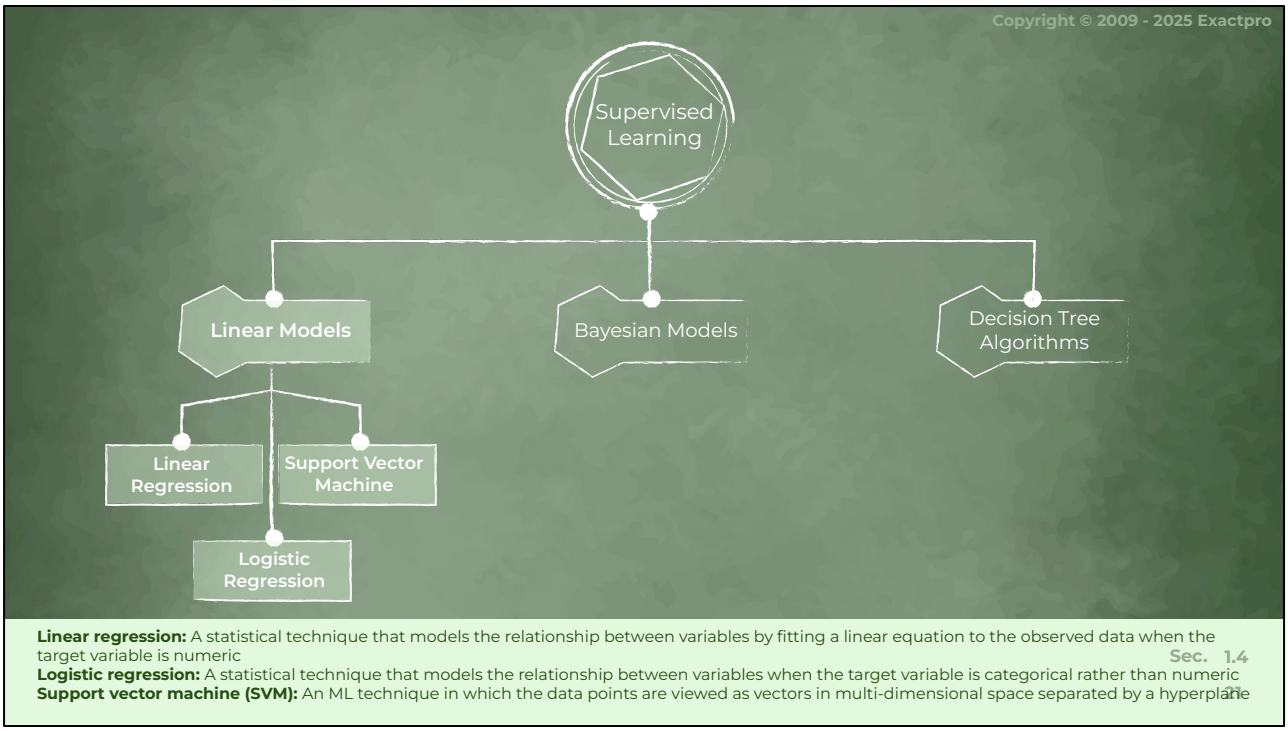
2. And then there are machine learning (ML) techniques which can be split into 3 categories: Supervised Learning, Unsupervised Learning and Reinforcement Learning, and two additional algorithm types that do not belong to the aforementioned categories. Let's look at the most common examples of these algorithms.



- **Training phase:** The labelled data is used to deduce the link between the input data and the output labels
- **Testing phase:** A new data set is applied to the trained model to predict the output. The model is deployed once the output precision level is satisfactory
- **Application domain:** Classification and regression problems

ML model: ML output of an ML algorithm trained with a training dataset that generates predictions using patterns in the input data
Supervised learning: Training an ML model from input data and its corresponding labels
ML model training: The process of applying the ML algorithm to the training dataset to create an ML model
Dataset: A collection of data used for training, evaluation, testing and prediction in ML

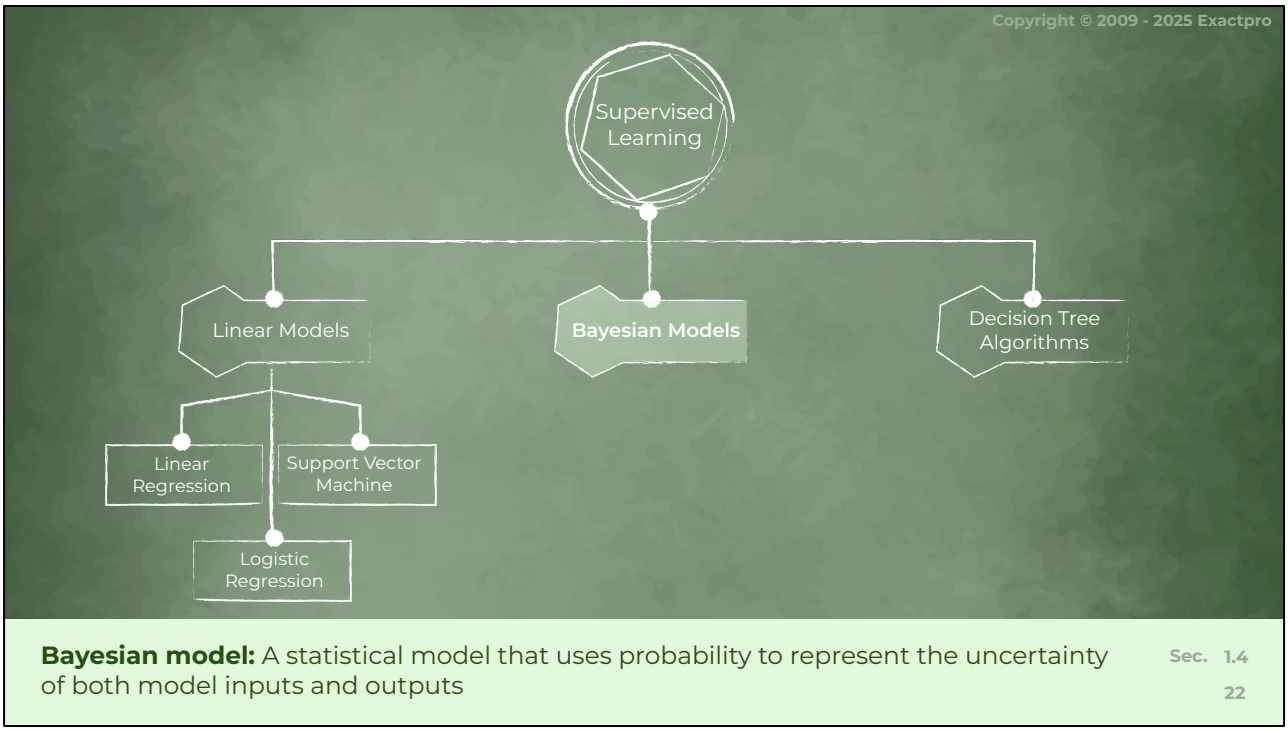
The first one is **supervised learning**. Here, the algorithm creates an ML model from labelled data during the training phase. The labelled data, which usually consists of pairs of inputs (for example, a bug report with a “bug” label vs any other text with a “not-a-bug” label) is used by the algorithm during the training to deduce the link between the input data and the output labels. During the ML model testing phase, a new data set is applied to the trained model to predict the output. The model is deployed once the output precision level is satisfactory. Supervised learning helps with classification and regression problems, such as determining what category a bug report belongs to (that describes an actual bug or just containing a user’s opinion about specific functionality) or predicting the time it takes to fix the bug.



The models presented in the syllabus can be divided into three groups: linear models, bayesian models and decision tree algorithms.
Let's look at a few examples for each group:

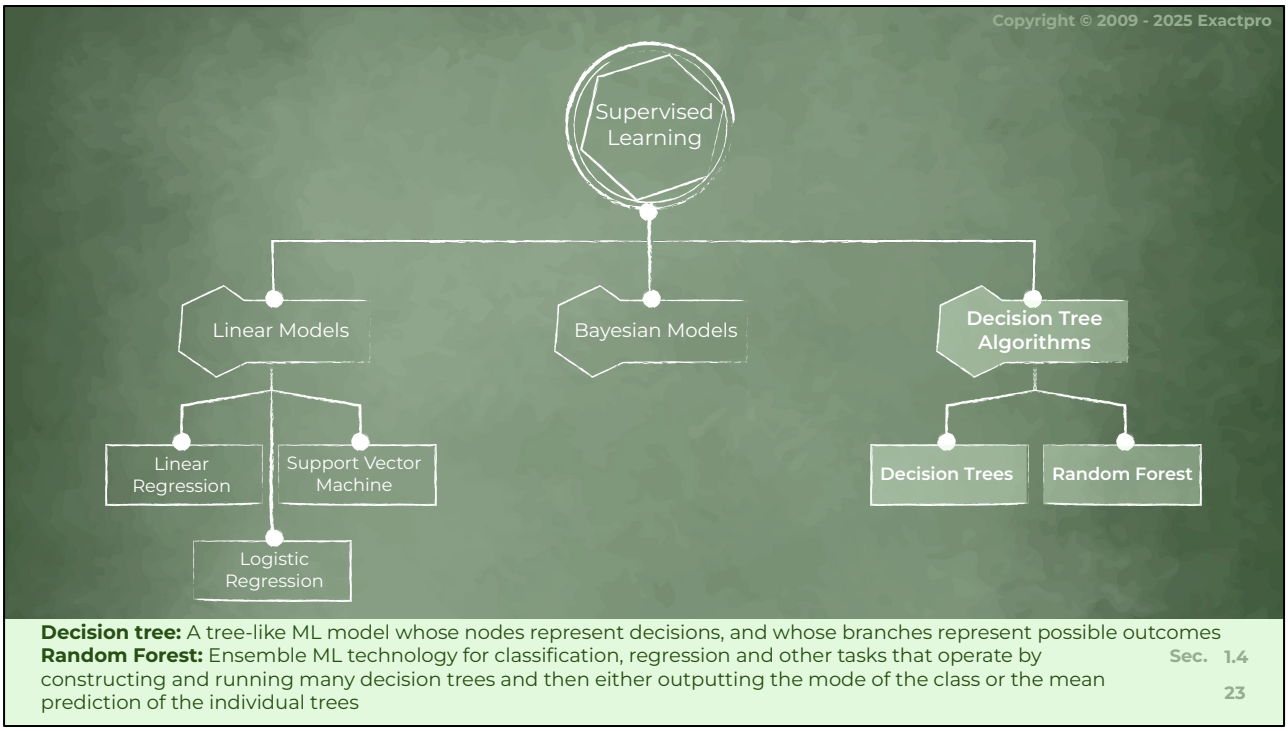
a. Linear Models

- **Linear regressio**n is a model reflecting a linear relationship between a response and one or more explanatory variables. It's used when the target variable is numeric and allows us to predict continuous values.
- **Logistic regression** is a modification of linear regression, it describes a process of modelling the probability of belonging to a certain class given an input variable. In this model, the target variable is categorical rather than numeric.
- **Support vector machine (SVM)** is a model that maps objects from training data to points in space in such a way as to increase the gap between classes. In this ML technique the data points are viewed as vectors in multi-dimensional space separated by a hyperplane.



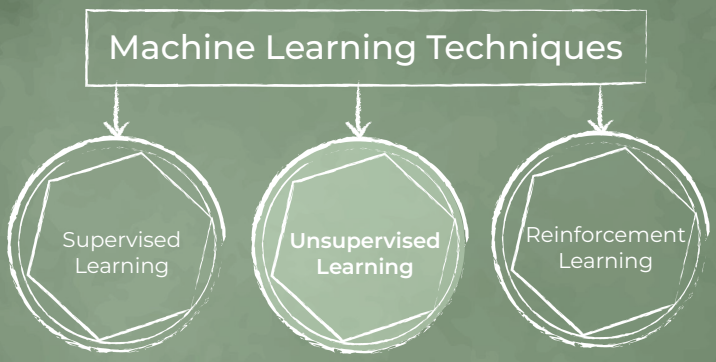
Bayesian model: A statistical model that uses probability to represent the uncertainty of both model inputs and outputs

b. **Bayesian models** are statistical models where plausibility is used to represent all uncertainty within the model in both inputs and outputs. In other words they specify probabilistic models and solve problems when less than the necessary information is available. Examples of Bayesian models include Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, etc.



c. **Decision tree algorithms**:

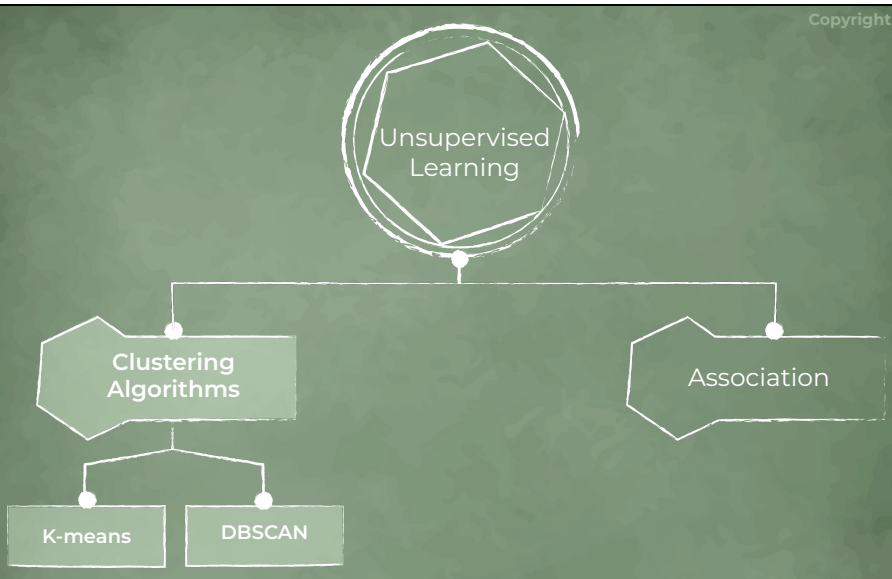
- **Decision trees** are flowchart-like structures in which each internal node contains a decision rule, each branch is a solution, and each end node represents a target value.
- **Random forest** is a learning method that operates by constructing a multitude of decision trees during training and then outputting either the mode of the class or the mean prediction of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.



- **Training phase:** An ML model is created from the unlabelled data used to deduce patterns in the input data. After that inputs are assigned to different classes
- **Testing phase:** The trained model is applied to a new data set. The model is deployed once the output precision level is satisfactory
- **Application domain:** Clustering and association problems

Unsupervised learning: Training an ML model from input data using an unlabelled dataset
Clustering: A type of ML function that groups similar data points together

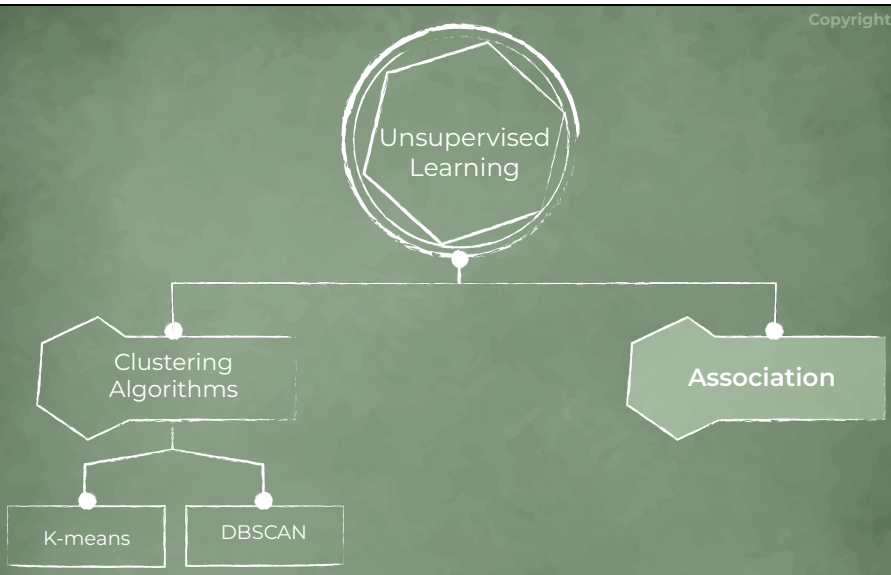
Then, there is the **unsupervised learning**. Here the algorithm creates an ML model from unlabelled data during the training phase. The unlabelled data is used to deduce patterns in the input data during the training. After that it assigns inputs to different classes. This is followed by the testing phase where the trained model is applied to a new data set. The model is deployed once the output precision level is considered to be satisfactory. For instance, this approach helps with clustering, which is when the problem requires the identification of data similarities that allow them to aggregate tens of thousands of parameters received from the exchange system into several groups, to make it easier to understand the causes of system failure.



Clustering algorithm: A type of ML algorithm used to group similar objects into clusters
K-means: An ML algorithm which attempts to cluster the data into a predetermined number of groups by minimising the distance from the cluster centre to the objects
Density-based spatial clustering of applications with noise (DBSCAN): An ML algorithm which operates on data density. The basic concept is to find areas of high density, which are separated from each other by areas of low density

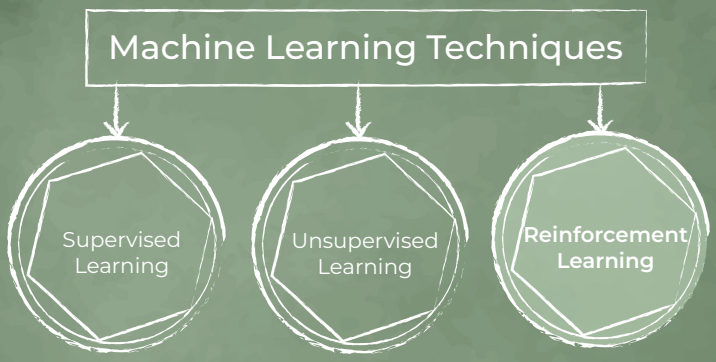
Clustering algorithms solve the task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups. It is a main task of exploratory data analysis, and a common technique for statistical data analysis.

- **K-means** which attempts to cluster the data into a predetermined number of groups by minimising the distance from the cluster centre to the objects.
- **DBSCAN (Density-based spatial clustering of applications with noise)** operates on data density. The basic concept of the algorithm is to find areas of high density, which are separated from each other by areas of low density.



Association: An unsupervised learning technique that identifies relationships and dependencies between samples

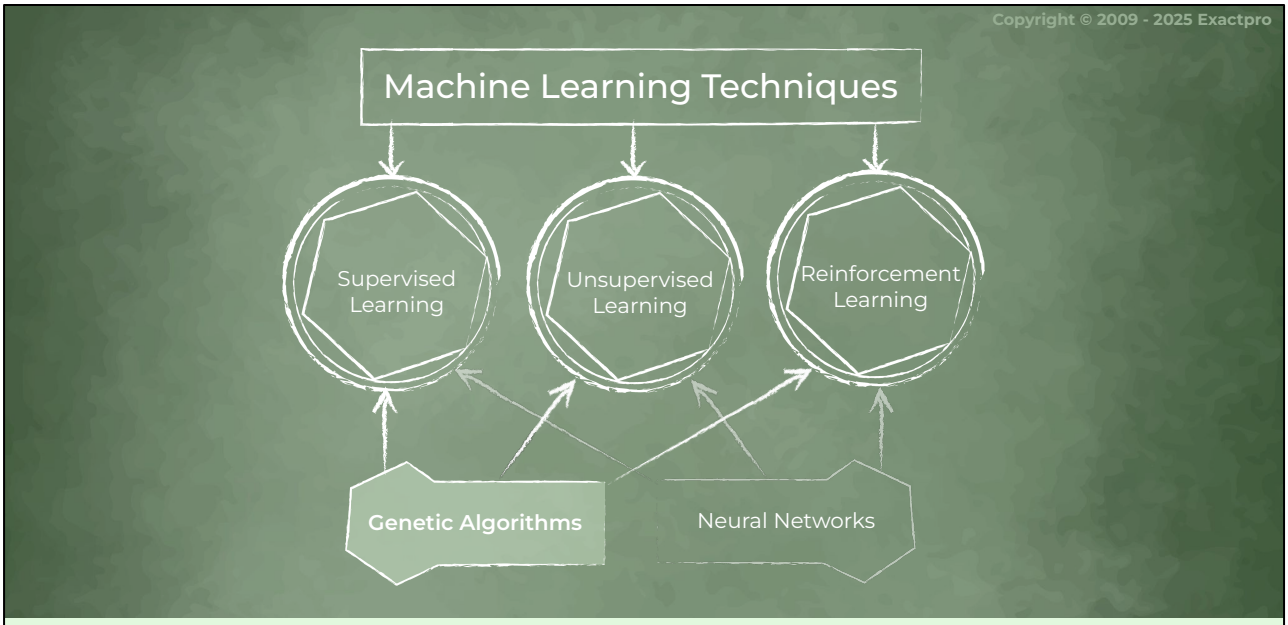
Another issue unsupervised learning aids us with is **association**. This technique identifies relationships and dependencies between samples or dependencies to be discovered among data attributes. For example, a bug label recommendation based on its description.



- **Training phase:** No training data. The system learns by interacting with the environment in an iterative manner
- **Key challenges:** Setting up the environment, selecting a strategy so that the "intelligent agent" could achieve the desired goal, designing a reward function that defines the success of learning
- **Application domain:** Robotics and autonomous vehicles

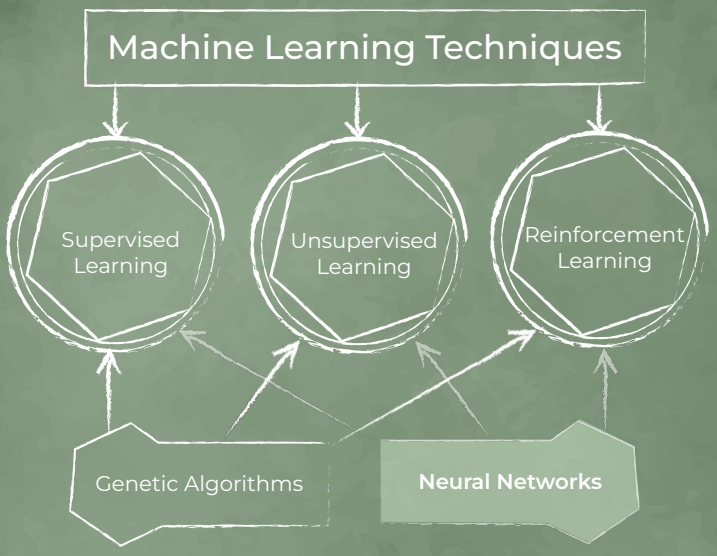
Reinforcement learning: The activity of building an ML model using a process of trial and reward to achieve an objective
Intelligent agent: An autonomous program which directs its activity towards achieving goals using observations and actions
Reward function: A function that defines the success of reinforcement learning

The last one is **reinforcement learning**, which is an approach where the system (an "intelligent agent") learns by interacting with the environment in an iterative manner. There is no training data. The agent is rewarded when it makes a correct decision and penalised when it makes an incorrect one. The key challenges when implementing reinforcement learning are setting up the environment, selecting a strategy so that the agent could achieve the desired goal, as well as designing a reward function that defines the success of reinforcement learning. Robotics and autonomous vehicles are examples of applications that use this approach. This approach can be implemented using Markov algorithms, dynamic programming and other, more sophisticated, algorithms.



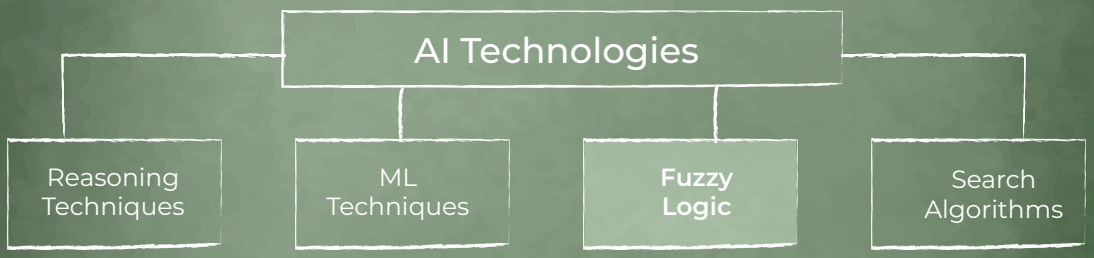
Genetic algorithm: A search-based algorithm used for solving optimisation problems in machine learning

Also, there are two other algorithm types that need to be mentioned. First is **genetic algorithms**, which are commonly used to generate high-quality solutions for optimisation and search problems by relying on bio-inspired operators such as mutation, crossover and selection. They cannot be classified as supervised learning or unsupervised learning, as they are applied over algorithms belonging to one group or another one.

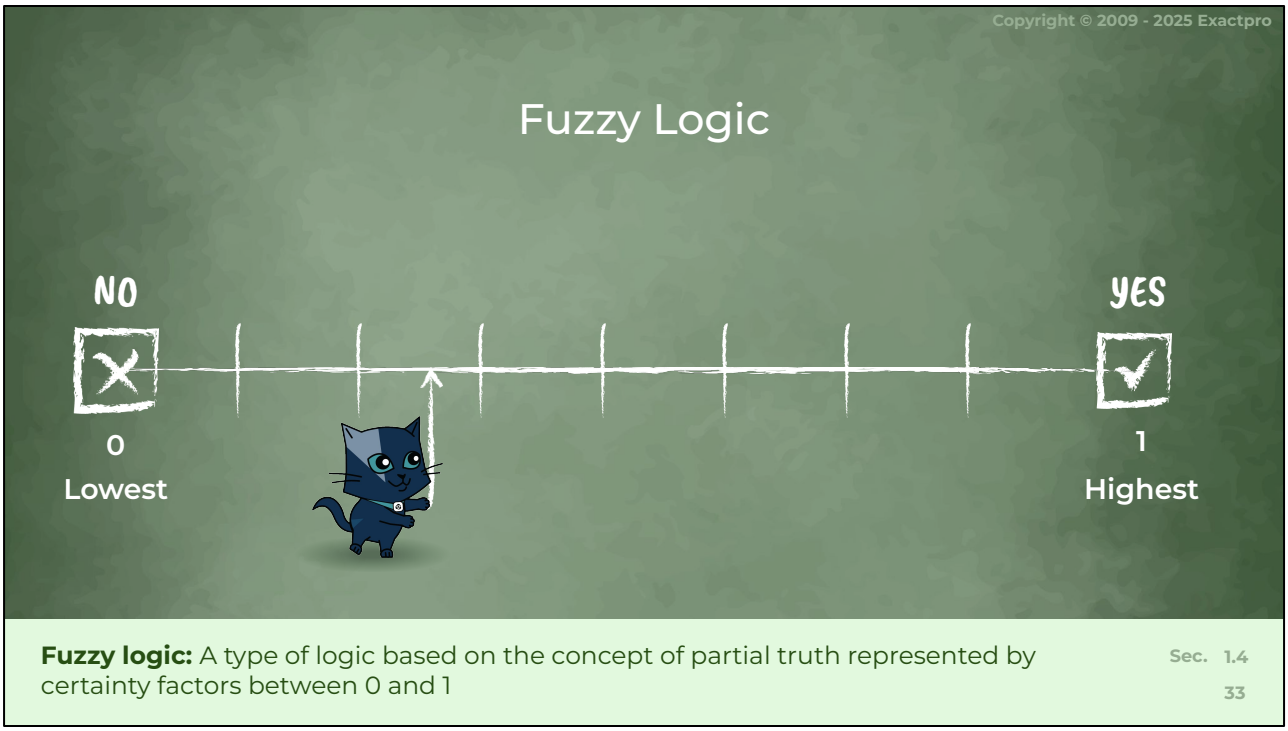


Neural network: An ML algorithm which attempts to mimic the human brain by constructing a system of connected units or nodes called artificial neurons

And secondly, **neural networks** which attempt to mimic the human brain by constructing a system of connected units or nodes called artificial neurons. They can be used in supervised, unsupervised and reinforcement learning tasks.



The next AI technology on our list is fuzzy logic.

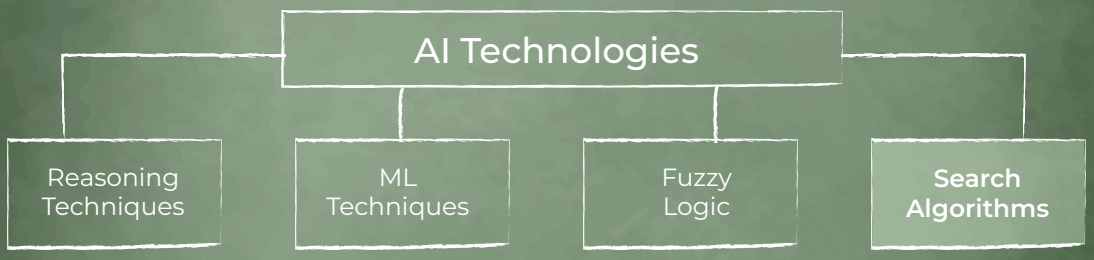


To put it simply, it operates in the grey area between human responses of yes and no, in other words, when the truth value is any real number between 0 and 1. It is used to handle the notion of partial truth. This is very similar to human reasoning. For example, when a tester assigns a priority level to a bug, he answers the question "Does this bug need to be fixed quickly?" Usually, it's not a yes-or-no question.
We set the priority in a range from the "Highest" (or "1" meaning "needs to be fixed very fast") to the "Lowest" (or "0" meaning "can be left for later") with intermediate values in between.

Example:
Let's consider a temperature control system for a room. Traditionally, we might use a rule like this. If the room temperature is greater than 25 degrees Celsius, then set the air conditioner to high power.However, this approach and doesn't account for the nuances of temperature perception. With fuzzy logic, we can introduce variable sets (Temperature and Power Level) to handle the uncertainty associated with temperature perception.
Temperature: Cold, Cool, Moderate, Warm, and Hot.
Power Level: We can define linguistic terms like Low, Medium, and High.
Fuzzy rules:

- *IF Temperature is Cold OR Temperature is Cool THEN Power Level is Low.*
- *IF Temperature is Moderate THEN Power Level is Medium.*
- *IF Temperature is Warm OR Temperature is Hot THEN Power Level is High.*

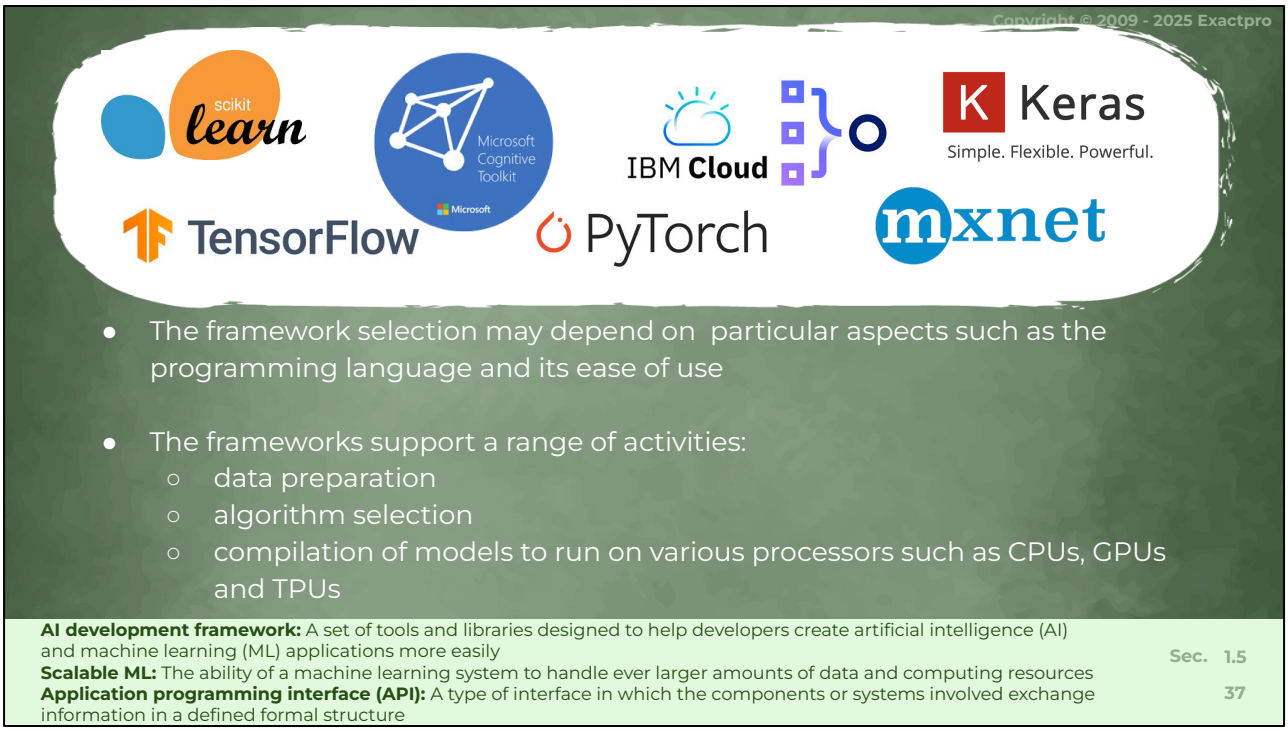
Given these rules, let's say the current temperature in the room is 23 degrees Celsius. We can evaluate the fuzzy sets for the linguistic variable "Temperature" to determine the membership values for each linguistic term. For instance, "Cold" might have a value of 0.2, "Cool" might have a value of 0.8, and so on. In this example, since the temperature is 23 degrees Celsius, the "Cool" term has the highest membership value. Using the fuzzy rules, we can infer that the power level should be "Low" or "Medium" based on the degree of membership. Finally, we can use defuzzification methods to convert the fuzzy output (e.g., "Low" and "Medium") into a crisp value that represents the air conditioner's power level, which can then be used to control the device accordingly.



Search algorithm: An algorithm that systematically visits a subset of all possible states or structures until the goal state or structure is reached

And the last AI technology to be mentioned here is search and recommendation algorithms. The most obvious way search engines use these algorithms is to rank web pages, videos, music, books, games and other content.
AI-based systems typically implement one or more of these technologies.

1.5 AI Development Frameworks



AI was once limited to a close community of leading researchers. But times have changed, and due to the development of various libraries and frameworks, it has become more user-friendly with more and more people going for it. Basically speaking, AI framework access is no longer limited to nerds and geniuses. There are many AI development frameworks available, some of which are focused on specific domains. The following frameworks could be considered as some of the most popular today:

- **Apache MxNet:** A deep learning open-source framework used for Amazon Web Services (AWS).
- **CNTK:** The Microsoft Cognitive Toolkit is an open-source deep-learning toolkit.
- **IBM Watsonx Studio:** A suite of tools that support the development of AI solutions.
- **TensorFlow:** An open-source ML framework based on data flow graphs for scalable ML, provided by Google.
- **Keras:** A high-level open-source API, written in the Python™ language, capable of running on top of TensorFlow and CNTK.
- **PyTorch:** An open-source ML library operated by Facebook and used for apps applying image processing and natural language

- processing (NLP). Support is provided for both Python™ and C++ interfaces.
- **Scikit-learn:** An open-source ML library for the Python™ programming language.

The framework selection may also depend on particular aspects such as the programming language and its ease of use. For example, if you use Python™, then you might be also interested in Keras, Pytorch, TensorFlow for neural network implementation or Scikit-learn for classical machine learning algorithms. The frameworks support a range of activities, such as data preparation, algorithm selection, and compilation of models to run on various processors, such as central processing units (**CPUs**), graphical processing units (**GPUs**) or Cloud Tensor Processing Units (**TPUs**).

1.6 Hardware for AI-Based Systems

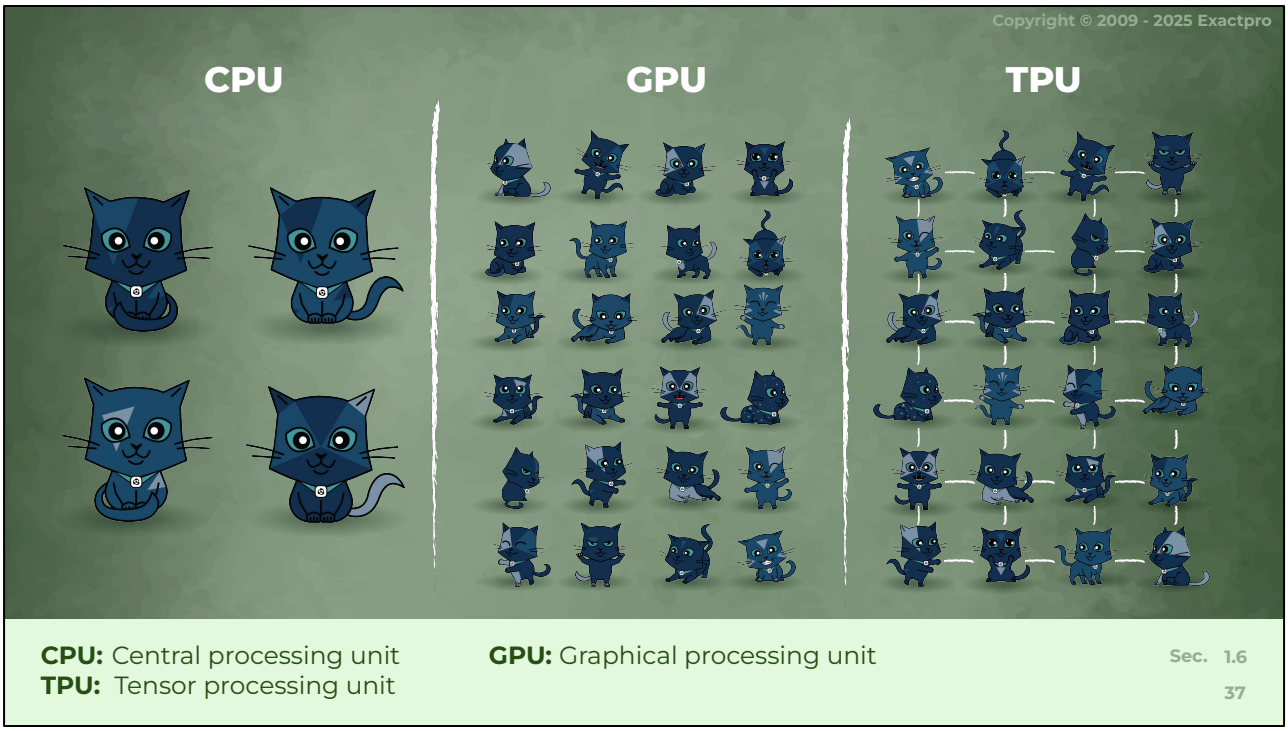
AI requires appropriate hardware. For instance, a model that performs speech recognition may run on a low-end smartphone, although access to cloud computing may be needed to train it. A common approach is to train the model in the cloud and then deploy it to the host device.

HARDWARE ATTRIBUTES that benefit ML:

- low-precision arithmetic (e.g., using 8 instead of 32 bits)
- the ability to work with large data structures
- massively parallel processing

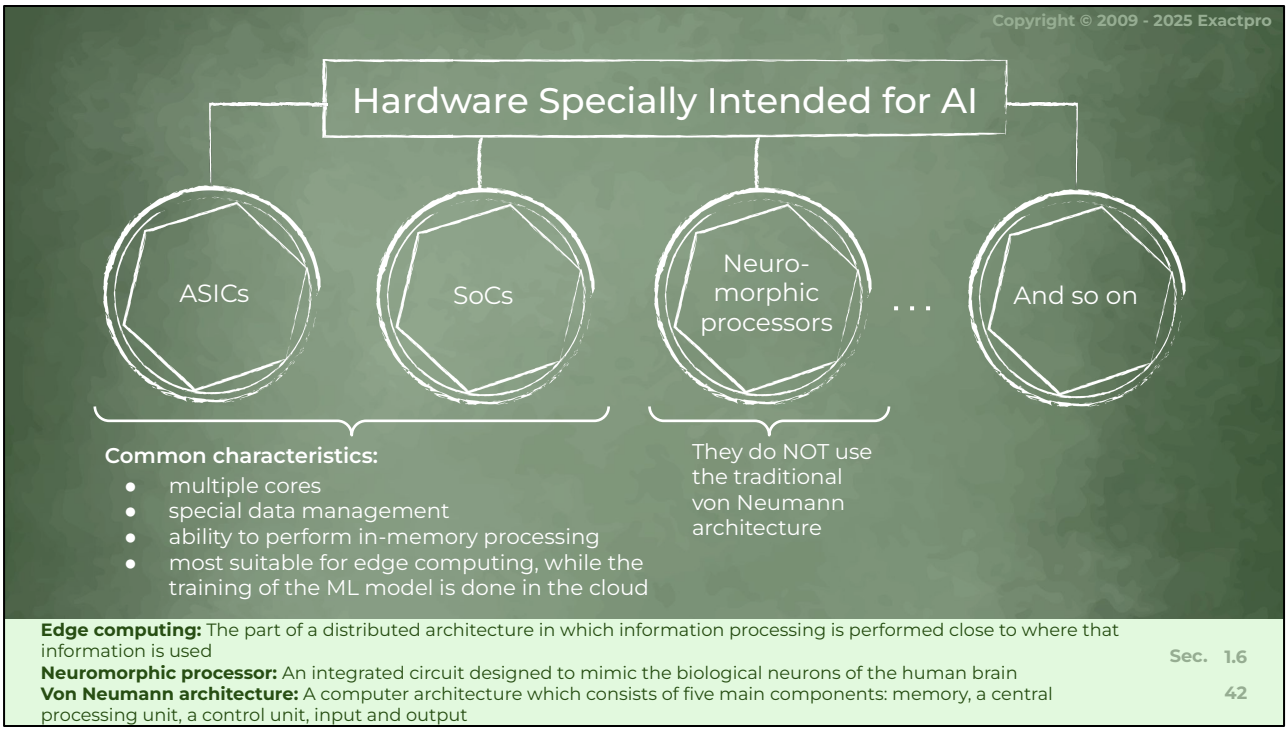
ML typically benefits from hardware that supports the following attributes:

- Low-precision arithmetic (for example, using 8 instead of 32 bits is sufficient for ML).
- The ability to work with large data structures.
- Massively parallel processing.



CPU (central processing unit) is a general-purpose processor with a small number of cores that are optimised for low-latency tasks that require single-threaded performance, such as running operating systems, web browsing or office applications. As a result, its architecture is less efficient for training and running ML models when compared to **GPU** (Graphical processing unit), which has thousands of cores and which is designed to perform massively parallel but relatively simple processing of images. Consequently, GPU typically outperforms CPU for ML applications, even though CPU typically has faster clock speeds. For small-scale ML work, GPU generally offers the best option.

TPU is a specialised processor developed by Google specifically to perform highly-optimised operations on tensors, which are multidimensional arrays used in deep learning models. They assure significant performance advantages over CPU and GPU for machine learning tasks.



Some hardware is specially intended for AI, such as purpose-built Application-Specific Integrated Circuits (**ASICs**) and System on a Chip (**SoC**). These AI-specific solutions have multiple cores, special data management and the ability to perform in-memory processing. They are most suitable for edge computing, while the training of the ML model is done in the cloud.

Hardware with specific AI architectures is currently being researched and developed. This includes neuromorphic processors, which are designed to mimic the biological neurons of the human brain and do not use the traditional von Neumann architecture.

Examples of AI hardware providers and their processors include:

- **NVIDIA:** They provide a range of GPUs and AI-specific processors, such as the Volta.
- **Google:** They have developed application-specific integrated circuits for both training and inferencing. Google TPUs can be accessed by users on the Google Cloud, whereas the Edge TPU is a purpose-built ASIC designed to run AI on individual devices.
- **Intel®:** They provide Core™ Ultra and Xeon® processors, which include neural processing units (NPUs) as a key component, designed to handle AI and ML tasks with unparalleled speed and efficiency. Core™ Ultra processors have powered the world's first AI personal computers (PCs), produced by such companies as Acer and Microsoft. Intel also produces Movidius™ Myriad™ X vision processing units (VPUs) for

- inferencing in computer vision and neural network applications.
- **Mobileye™:** They produce the EyeQ® family of SoC devices to support complex and computationally intense vision processing to be used in vehicles.
- **Apple:** They produce the Bionic chip for on-device AI in iPhones.
- **Huawei:** Their Kirin 970 chip was the world's first that has a built-in neural network processing unit (NPU) for AI.

1.7 AI as a Service (AIaaS)

WAYS TO CREATE AI COMPONENTS:

- create within an organisation
- download from a third party
- use as a service
- a hybrid approach

AIaaS BENEFITS:

- access to an ML model is provided over the web in order to perform data preparation, model training, evaluation, tuning, testing and deployment
- possibility to implement AI using cloud-based services even with insufficient resources
- training ML models on a larger, more diverse dataset available for those who have recently moved into the AI market

AI as a Service (AIaaS): A software licensing and delivery model in which AI and AI development services are centrally hosted
AI component: A component that provides AI functionality

AI components, such as ML models, can be created within an organisation, downloaded from a third party, or used as a service, although a hybrid approach is also possible. When ML is used as a service, access to an ML model is provided over the web in order to perform data preparation, model training, evaluation, tuning, testing and deployment. Third-party providers can offer specific AI services, such as facial and speech recognition. This allows individuals and organisations to implement AI using cloud-based services even with insufficient resources. In addition, these ML models are likely to have been trained on a larger, more diverse dataset available for those who have recently moved into the AI market.

1.7.1 Contracts for AI as a Service

AIaaS CONTRACTS:

- are similar to contracts for non-AI cloud-based SaaS
- usually contain SLA that defines availability and security commitments

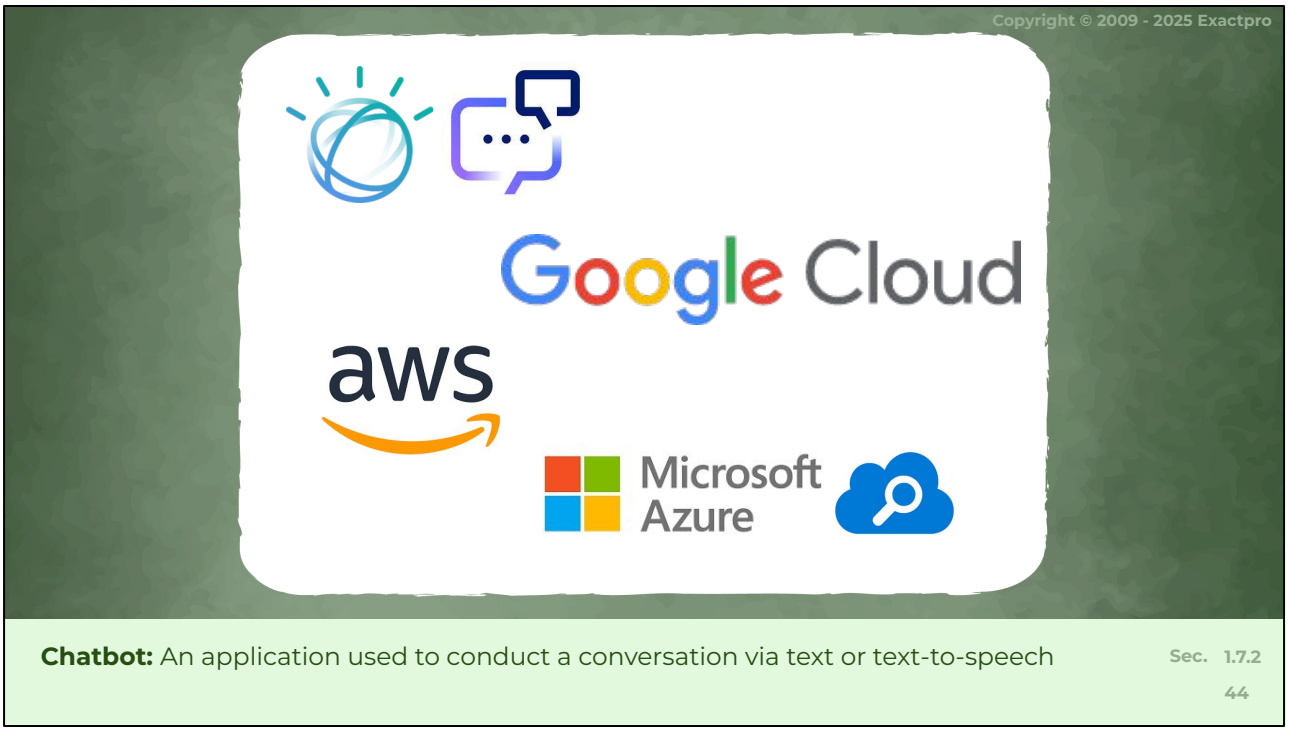
AIaaS PROS	AIaaS CONS
+ No need to create the service from scratch	– Poorly defined ML functional performance metrics
+ Availability	– Limited liability
+ Subscription basis and credits for future services	
+ Free trial	
+ No need to use your own resources	
+ Cloud-based	
+ Pre-trained on large and diverse datasets	

Software as a Service (SaaS): A software distribution model in which a cloud provider hosts applications and makes them available to end users over the internet
Service-level agreement (SLA): A contract between a service provider and its customers that documents what services the provider will furnish and defines the service standards the provider is obligated to meet

The contracts for these AI services are similar as for non-AI cloud-based Software as a Service (**SaaS**). An AIaaS contract usually contains a service-level agreement (**SLA**) that defines availability and security commitments.
It covers the service uptime and a response time to fix defects, but rarely defines ML functional performance metrics. The payment is subscription based, and if the contracted availability and/or response times are not met, then the service provider could offer credits for future services. Other than that, most AIaaS contracts provide limited liability, thus limiting it to relatively low-risk applications. But on the other hand services often come with an initial free trial period.

1.7.2 AIaaS Examples

The following are examples of AIaaS:



The following are examples of AIaaS:

- **IBM Watsonx Assistant:** An AI chatbot priced according to the number of monthly active users.
- **Google Cloud AI and ML Products:** These provide different products including speech-to-text and text-to-speech converters, translation AI, AI used for video analysis, AI with document-based form parser and document OCR, and others. Prices depend on the length of video and audio, the number of characters or the number of pages sent for processing per month.
- **Amazon CodeGuru:** This provides ML Java™ and Python™ code reviews supplying developers with recommendations. Prices are based on the number of lines of source code analysed.
- **Microsoft Azure AI Search**: This provides AI cloud search. Prices are based on search units defined in terms of the storage and throughput.

1.8 Pre-Trained Models

1.8.1 Introduction to Pre-Trained Models

TO TRAIN AN ML MODEL:

- can be expensive
- can require a large amount of human and computing resources

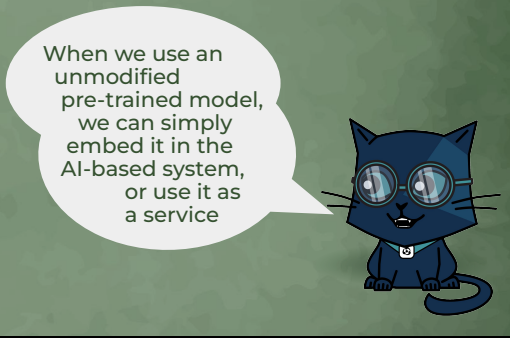
A PRE-TRAINED MODEL COULD BE A SOLUTION, BUT THIS SOLUTION HAS ITS OWN LIST OF PROS AND CONS

Pre-trained model: An ML model already trained when it was obtained

Make no mistake, ML models can be expensive, consuming large amounts of human and computing resources. So a cheaper and often more effective alternative is to use a pre-trained model.

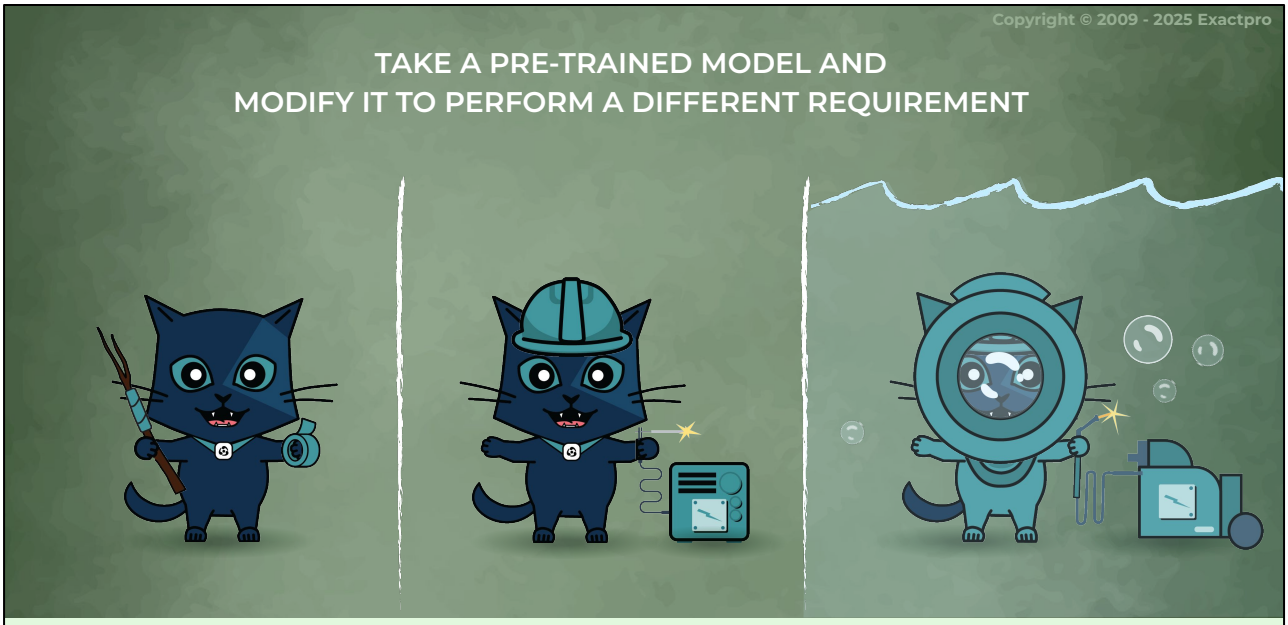
PRE-TRAINED MODEL

PROS	CONS
+ Providing similar functionality	— These models are only available for a limited number of technologies like neural networks and random forests
+ The basis existence for creating a new model that extends the functionality of the pre-trained model	
+ Reducing risk of consuming significant resources with no guarantee of success	



This provides similar functionality and is used as the basis for creating a new model that extends the functionality of the pre-trained model.
Using pre-trained models reduces the risk of consuming significant resources with no guarantee of success. But these models are only available for a limited number of technologies, such as neural networks and random forests.
When we use an unmodified pre-trained model, we can simply embed it in the AI-based system, or use it as a service.

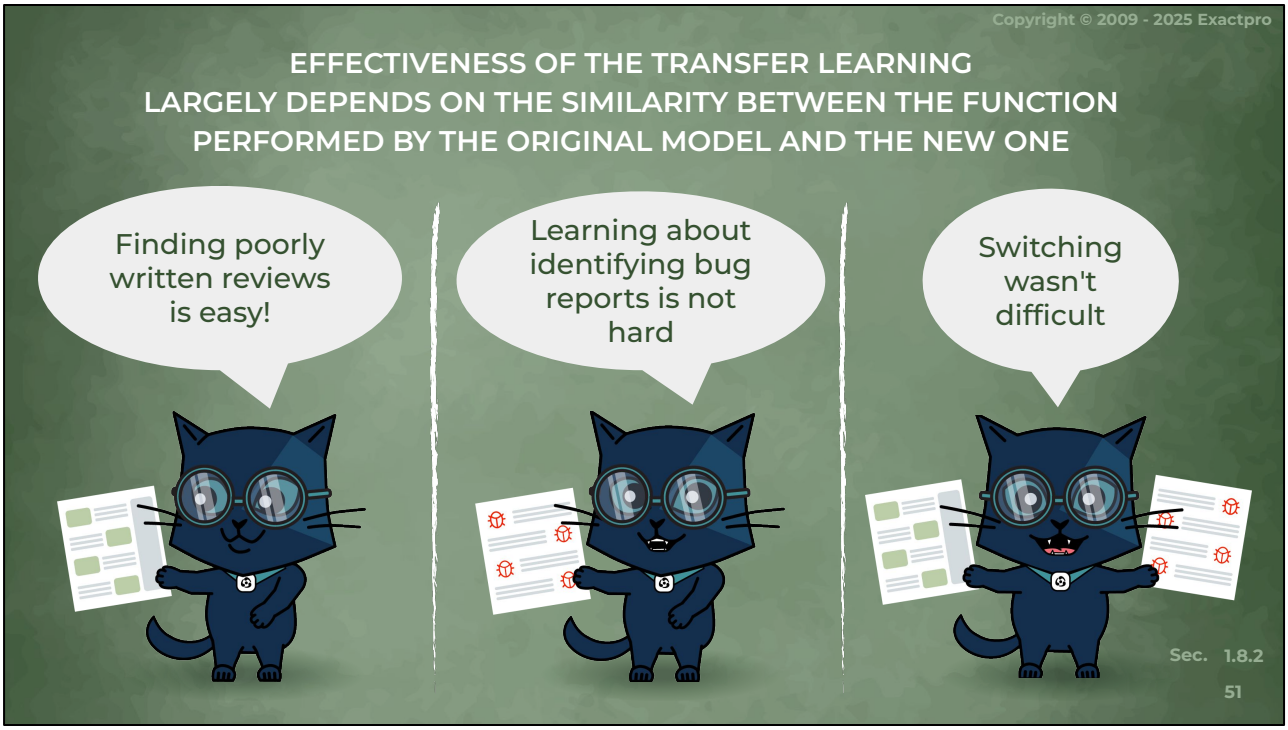
1.8.2 Transfer Learning



Transfer learning: A technique for modifying a pre-trained ML model to perform a different related task

Deep neural network (DNN): A neural network comprised of several layers of neurons

It is also possible to take a pre-trained model and modify it to perform a different requirement. This is known as transfer learning and is used on deep neural networks (DNNs) in which the early layers of the neural network typically perform basic tasks, whereas the later layers perform more specialised assignments. This eliminates the need to train the early layers. The later layers are then retrained to handle the unique requirements for a new classifier. In practice, the pre-trained model may be fine-tuned with additional training on new problem-specific data. The effectiveness of this approach largely depends on the similarity between the function performed by the original model and the new one.



For example, modifying a text search algorithm that identifies reviews on some products to identify bug reports on some issues would be far more effective than modifying it to identify computers in some images.

EXAMPLES OF PRE-TRAINED MODELS

ImageNet models for image classification:

- Inception
- VGG
- AlexNet
- MobileNet

NLP models:

- Google's BERT
- NLTK
- GPT

COMMUNITIES FOR DATA SCIENTISTS AND ML ENGINEERS OFFER PUBLIC DATA AND CLOUD-BASED PLATFORMS FOR DATA SCIENCE, AI EDUCATION AND BUILDING AI MODELS

VGG: Visual geometry group
BERT: Bidirectional encoder representations from transformers
GPT: Generative pre-trained transformer

There are many pre-trained models available, especially from academic researchers. Some examples of such pre-trained models are ImageNet models such as AlexNet and MobileNet for image classification and pre-trained NLP models like Google's BERT, NLTK and GPT.
There are also different communities for data scientists and machine learning engineers which offer public data and cloud-based platforms for data science, AI education and building AI models. They promote competitions to solve data science challenges. One of such communities is Kaggle.

1.8.3 Risks of Using Pre-Trained Models and Transfer Learning

RISKS OF USING PRE-TRAINED MODELS:

- lack of transparency
- insufficient level of similarity
- differences in the data preparation steps impact the resulting functional performance
- inherited and not documented shortcomings
- sensitivity to vulnerabilities

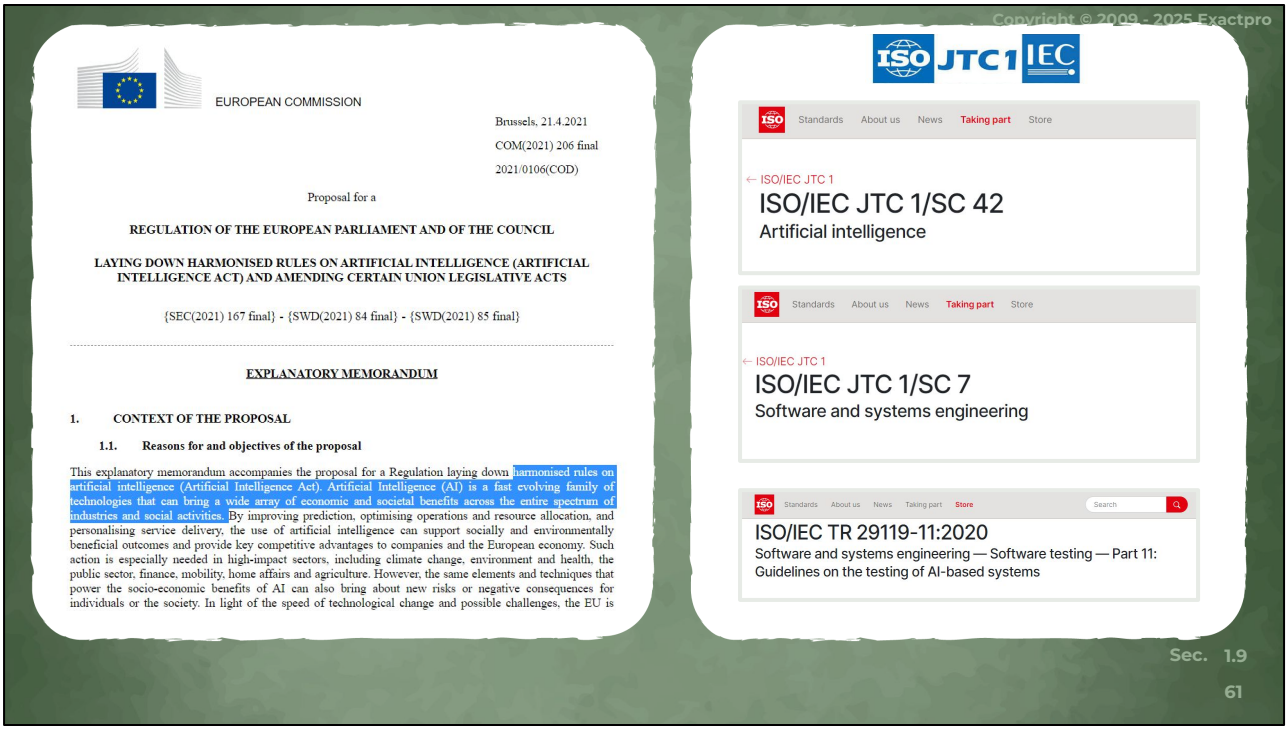
SEVERAL RISKS CAN BE MITIGATED BY HAVING THOROUGH DOCUMENTATION FOR THE PRE-TRAINED MODEL

Using pre-trained models, of course, carries some risks. These include:

- Pre-trained models may lack transparency compared to internally generated ones.
- The level of similarity between the function performed by the pre-trained model and the required functionality may be insufficient. Also, this difference may not be understood by the data scientist.
- Differences in the data preparation steps used for the pre-trained model, when originally developed, and the data preparation steps, when this model is deployed, may impact the resulting functional performance.
- The shortcomings of a pre-trained model are likely to be inherited by those who reuse it and may not be documented.
- A model created through transfer learning is highly likely to be sensitive to the same vulnerabilities as the pre-trained model on which it is based.

Note that several of the aforementioned risks can be easily mitigated by having thorough documentation available for the pre-trained model.

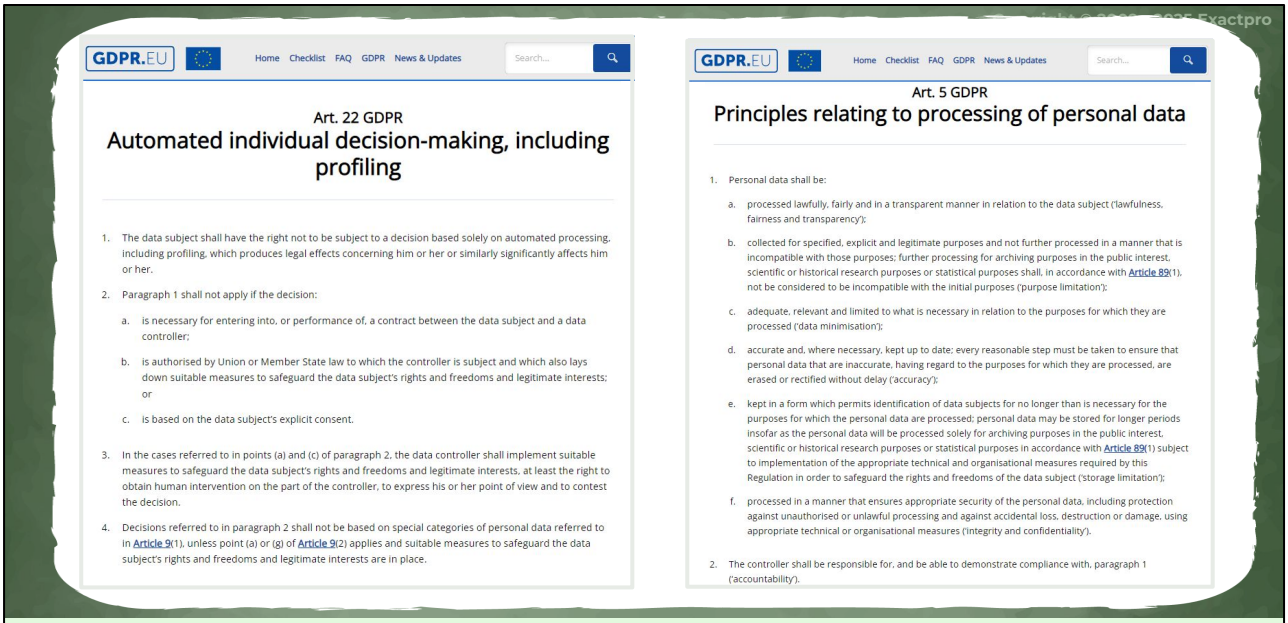
1.9 Standards, Regulations and AI



The proposal of the European Parliament and of the council focuses on harmonised rules of AI, stating that this is a "*fast evolving family of technologies that can bring a wide array of economic and societal benefits across the entire spectrum of industries and social activities*" that could provide competitive advantages to companies and the economy. In noting so, it outlined the following AI techniques and approaches:

- Machine learning approaches, including supervised, unsupervised, reinforcement and deep learning;
- Logic- and knowledge-based approaches, including knowledge representation, inductive programming, knowledge bases, inference and deductive engines, symbolic reasoning and expert systems;
- Statistical approaches, Bayesian estimation, search and optimisation methods.

The Joint Technical Committee (JTC) of IEC (International Electrotechnical Commission) and ISO (International Organisation for Standardisation) on information technology releases international standards about AI as well. For example, an AI subcommittee ISO/IEC JTC 1/SC 42 (SC – Sub Committee), was set up in 2017. In addition, ISO/IEC JTC 1/SC 7, which covers software and system engineering, has published a technical report (TR) on the "Testing of AI-based systems" (ISO/IEC TR 29119-11). Standards on AI are also published at the regional level and the national level.



General Data Protection Regulation (GDPR): The European Union (EU) regulation on data protection and privacy that applies to the data of citizens of the EU and the European Economic Area

Also General Data Protection Regulation (GDPR) came into effect in May 2018 and set EU-wide rules with regards to personal data and automated decision-making. It includes requirements to assess and improve AI system functional performance, including the mitigation of potential discrimination, and for ensuring individuals' rights to not be subjected to automated decision-making. The most important aspect of the GDPR from a testing perspective is that personal data (including predictions) should be accurate. This does not mean that every single prediction made by the system must be accurate, but that the system should be precise enough for the purposes for which it is used.

DIN

DIN SPEC 92001-1

Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 1: Quality Metamodel

DIN SPEC 92001-1 CASE STUDY

Ensuring AI quality

The background

Artificial intelligence (AI) describes the capability of an IT system to process information in a manner similar to humans. AI is based on neural networks and IT architectures that emulate those of the human brain. Instead of storing information unquestioningly like conventional storage elements, these networks and architectures process it autonomously and react to it. AI is one of the most significant topics of the future, impacting not only on industry and business but on other areas too, including our private sphere. But whatever the area, the quality aspect plays a central role. Unlike traditional software, the reactions of AI cannot be predicted with certainty because it acts autonomously. This needs to be taken into account over the entire AI life cycle and makes quality management of AI more difficult.

The DIN SPEC

With this in mind, DIN SPEC 92001 aims to ensure the quality of AI by means of a coherent concept, defining a quality metamodel that comprises and combines all key aspects of AI quality. Special focus is placed on the individual phases in the life cycle of AI modules, as certain quality aspects are only relevant at certain points in time. Particularly in the conception and development stages, for example, it is important to prevent bias in the processing of information. In addition, the metamodel identifies the three quality pillars that are the overarching goals of quality assurance: functionality and performance, robustness and comprehensibility. The metamodel according to DIN SPEC 92001-1 also takes into account those aspects relating to the development of an AI module that are relevant to quality. In this way the model creates the structural basis for the specific AI quality requirements that the second part of the specification, DIN SPEC 92001-2, "Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 2: Quality Requirements", describes in greater detail.

DIN SPEC 92001 differentiates between high-risk and low-risk AI modules: Depending on where they are used, AI modules can have safety, security, privacy or ethical relevance – in this case DIN SPEC 92001-1 classes these modules as "high risk". Deviations from defined quality requirements are either permitted or require appropriate justification. In AI applications this could be the case, for example, if human life were at risk. In contrast, the recommendations for dealing with AI modules that are classed as "low risk" and that consequently have no safety, security, privacy or ethical relevance, are less stringent. Examples of these include harmless AI applications such as the automatic classification of text and images.

The benefits

DIN SPEC 92001-1 structures the typical challenges and approaches to AI module quality, providing developers and users with a clearly organized and easily understandable basis. Reflecting the fact that AI is deployed in so many areas of life, DIN SPEC 92001-1 addresses the full range of potential uses. "With the metamodel, DIN SPEC 92001-1 creates the foundation for robust, explainable, safe, secure and trustworthy AI applications," says Stephan Hinze, Managing Director of neurocat GmbH and initiator of DIN SPEC 92001-1. "When working on the content, our main concern was to describe the require-

www.din.de/go/spec-en

DIN

DIN SPEC 92001-2

Artificial Intelligence – Life Cycle Processes and Quality Requirements – Part 2: Robustness

DIN SPEC 92001-2 CASE STUDY

Robust AI

The background

The term Artificial Intelligence (AI) covers a number of different approaches to optimize IT systems so that they can solve very specialized problems. Most AI algorithms are based on neural networks that benefit from the huge amounts of data available today. These neural networks are loosely modeled on the brains of living organisms. They can process information independently, react to it and learn to solve problems themselves. However, unlike the human brain they are susceptible to even minor disturbances in the input data. As a result, machine learning may no longer be able to solve a problem in a sufficiently robust manner, i.e. it may not be stable. However, because AI modules are increasingly used in highly sensitive industries and business areas, robustness and quality assurance throughout their entire life cycle are necessary.

The DIN SPEC

The DIN SPEC 92001-2 is based on DIN SPEC 92001-1. The latter describes a quality metamodel for AI modules, which introduces different phases in their life cycle and defines robustness, functionality and performance as well as comprehensibility as fundamental quality pillars. Part 2 now takes a closer look at the aspect of robustness under two specific angles. On the one hand, it deals with the robustness against mathematically optimized disturbances, which are used to generate malfunctions of AI, for example in the case of a targeted attack. On the other hand, robustness against naturally occurring signal disturbances or other impairments of data quality is necessary, which often occur when using AI. The DIN SPEC 92001-2 structures extensive technical requirements along a systematic risk management process. In the first step, "Scope, Context, Criteria", organizations must develop general requirements for their specific AI risks. This is followed by a comprehensive AI risk assessment, consisting of the steps "Threat Model Analysis", "Impact Analysis", and "Robustness Evaluation". Based on the empirical evaluation of the robustness, a targeted risk reduction is then carried out - for example, through measures to defend against attacks or to strengthen the robustness. Depending on the real development and deployment context of the AI, these steps should take place in digital, simulated, and/or physical AI environments. The DIN SPEC also suggests that developers and operators divide their AI modules into different risk categories in order to build up their robustness in a risk-oriented way. DIN SPEC 92001-2 comprises 53 technical requirements, which are classified in priority categories.

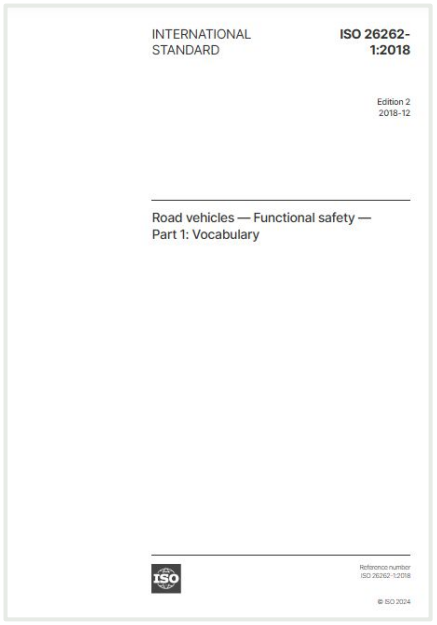
The benefit

The DIN SPEC 92001-2 structures the very lively research field of AI robustness and creates a detailed technical basis for robust and trustworthy AI applications. It enables both developers and users of AI to manage AI risks in a modern way by means of clearly formulated and pragmatically balanced guidelines and recommendations for action. To this end, the risks associated with the use of AI systems are analyzed and considered as they occur in the various processes and phases in the life cycle of an AI module. "With the help of DIN SPEC 92001-2, companies and other organizations can make their AI-based software more resistant to all conceivable disruptive processes," explains Stephan Hinze, Managing Director of neurocat GmbH and initiator of DIN-SPEC 92001-2. "It is a necessary building block in the AI strategy of every organization because it makes AI quality assurance transparent and comprehensible."

www.din.de/go/spec-en

The German national standards body (DIN – Deutsches Institut für Normung / German Institute for Standardisation) has developed the AI Quality Metamodel (DIN SPEC 92001-1, DIN SPEC 92001-2).

Standards on AI are also published by industry bodies. For example, the Institute of Electrical and Electronics Engineers (or IEEE) is working on a range of standards on ethics and AI (The IEEE Global Initiative 2.0 on Ethics of Autonomous and Intelligent Systems), although many of these standards are still in development.



INTERNATIONAL STANDARD

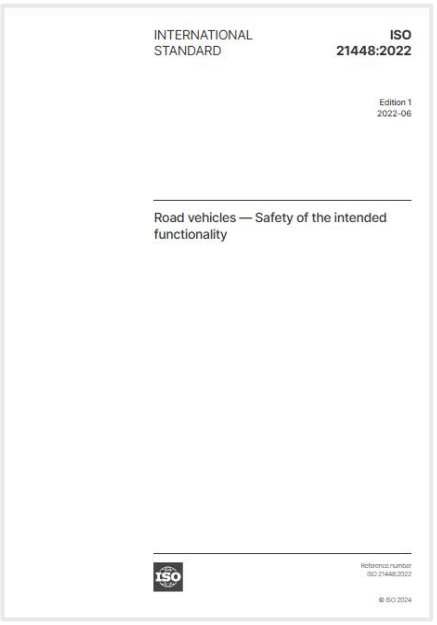
ISO 26262-1:2018

Edition 2
2018-12

Road vehicles — Functional safety — Part 1: Vocabulary

Reference number
ISO 26262-1:2018

© ISO 2024



INTERNATIONAL STANDARD

ISO 21448:2022

Edition 1
2022-06

Road vehicles — Safety of the intended functionality

Reference number
ISO 21448:2022

© ISO 2024

When it comes to safety-related systems, regulations become even more paramount, for example, regulatory standards ensure that it is illegal to sell a car in some countries if it includes software that does not comply with ISO 26262 and ISO 21448.
Standards are voluntary, and only made mandatory by legislation or contract. However, many users of these standards implement them to benefit from the expertise of the authors and to create higher quality products.

THANK YOU