

# AI Technology Taxonomy – Distinguishing Between Artificial Intelligence, Machine Learning, and Deep Learning

## Why Taxonomy Matters

The contemporary technological discourse you, as future programmers and computer scientists, encounter daily has been dominated to an unprecedented degree by artificial intelligence terminology. This phenomenon, driven largely by corporate marketing departments and the mass media, has led to a highly damaging industry problem known in software engineering as "AI washing."<sup>1</sup> Similar to greenwashing, it involves exaggerating or even falsely claiming advanced cognitive and analytical capabilities to commercial software products.<sup>1</sup> In reality, such software often relies on simple, deterministic algorithms, basic statistical filters, or nested conditional statements, which are merely promoted as "intelligent systems" to enhance their appeal to investors and bewildered consumers.<sup>4</sup> The scale of this phenomenon is so large that global regulatory bodies, such as the U.S. Securities and Exchange Commission (SEC), have begun imposing multimillion-dollar fines on companies for misleading the market by misusing the term "artificial intelligence" in their technology prospectuses.<sup>3</sup> Such terminological misuse blurs the lines between actual innovations based on neural networks and ordinary application software implementing rigid procedures.<sup>7</sup>

For a novice programmer, a lack of conceptual precision has extremely serious engineering and architectural consequences. Failure to understand the fundamental differences between artificial intelligence, machine learning, and deep learning leads to the incorrect selection of tools for the specific business problem.<sup>9</sup> Selecting the wrong technology, inadequate for the task, can result in drastic computational cost overruns, the complete inability to deploy the system on devices with limited hardware resources, or the construction of a model whose decisions cannot be rationally explained to the end user due to its unnecessary complexity.<sup>6</sup> Understanding the precise meaning of individual concepts allows for the optimization of the development team's work.

This can be illustrated with a direct analogy, familiar to you from your basic web application development classes. In the classic web software development paradigm, you clearly distinguish between the front-end layer (responsible for the user interface and rendering views in the browser), the back-end (handling business logic, authorization, and database operations), and the full-stack competencies (the fusion of these two broad domains).<sup>12</sup> Attempting to build a complex, relational database query using Cascading Style Sheets (CSS) would be a cardinal error stemming from a fundamental misunderstanding of the tools' purpose.<sup>13</sup> In the field of artificial intelligence, the same hierarchy of tool specialization and purpose prevails. Using a

powerful, multi-billion-dollar language model (LLM) to solve a simple tabular classification problem that a classic decision tree would handle perfectly in a fraction of a second and with minimal RAM consumption is the equivalent of using an industrial excavator to plant a single flower in a pot. <sup>11</sup>

Understanding this taxonomy allows for mature optimization of the software development process. Instead of succumbing to the marketing illusion that artificial intelligence is perceived as a magical "black box" capable of solving absolutely any IT problem, a conscious engineer can decompose tasks. <sup>10</sup> Recognizing whether a given problem requires programming rigid business logic rules, implementing a model learning from structured statistical data, or creating a multilayer neural network analyzing unformatted video streams is the foundation of modern programming and system architecture. <sup>11</sup> This research report, serving as a teaching script, aims to thoroughly organize this knowledge. It will serve as a conceptual glossary, providing precise definitions and clearly delineating the concepts you will implement in code in the subsequent stages of this professional course.

## Artificial Intelligence (AI) - the umbrella term

Artificial Intelligence ( **AI** ) is the broadest of the taxonomic categories discussed and, in engineering nomenclature, functions as an umbrella term for all computer systems that demonstrate the ability to perform tasks traditionally requiring the presence of human intelligence. <sup>15</sup> This definition is inherently broad. In the classical, computer science sense, AI is a broad field of computer science concerned with the creation of machines, algorithms, and systems capable of perceiving their environment, making deductive inferences, assimilating knowledge (learning), and making autonomous decisions to maximize the chances of achieving a programmer-defined goal. <sup>11</sup> Understanding AI, however, requires realizing that the concept is by no means homogeneous. From the perspective of theoretical and practical computational capabilities and the degree of evolutionary autonomy of machines, this field is divided into three main categories: narrow artificial intelligence, general artificial intelligence, and superintelligence. <sup>11</sup>

The first, fundamental category is narrow AI, abbreviated as **ANI (Artificial Narrow Intelligence)** , often also called Weak AI. It is defined as software specialized and optimized to solve only a single, precisely defined problem within a defined domain. <sup>11</sup> From an engineering perspective, this type of software can achieve results in its microdomain that are equal to, and due to the speed of its processors, often significantly exceed, human capabilities. However, such a system is completely helpless when faced with a task beyond its programmed scope. <sup>7</sup> To provide an example that makes sense from an engineering perspective: a system perfectly trained to analyze and diagnose radiological images would be unable to win the simplest game of tic-tac-toe. <sup>7</sup>

Examples of ANI surround users in their daily lives, often operating in the background of system processes. These include advanced spam filters in email clients, GPS navigation systems that optimize routes based on real-time traffic congestion analysis, recommendation engines for streaming platforms like Netflix and Amazon, and facial recognition algorithms in modern

smartphone cameras.<sup>15</sup> Each of these systems is a master of its narrow specialization. The key lesson you must draw from this and remember in your career path is absolute: absolutely everything we currently call "artificial intelligence" in the technology and commercial industries, from simple algorithms in washing machines to the most powerful language models like ChatGPT, is in fact only narrow AI.<sup>7</sup>

The second category is general artificial intelligence, known as **AGI (Artificial General Intelligence)** or Strong AI. Currently, this is only a theoretical model of a machine that would possess the ability to fully understand, learn, adapt to any intellectual problem, and think critically at a level precisely equivalent to the human brain.<sup>11</sup> Such a system would be able to freely and abstractly transfer knowledge from one domain to another, understanding context without the need for retraining on new databases. The third category is artificial superintelligence (**ASI**), defined as a hypothetical machine whose cognitive capabilities, creativity, ability to solve complex scientific problems, and social skills would significantly exceed the intellect of the most brilliant human minds on Earth.<sup>11</sup> Both AGI and ASI currently remain the domain of research, conceptual engineering, and futurological considerations.

Architecturally, by considering the evolution of systems under the artificial intelligence umbrella, we can observe how programmers' approaches have changed over the decades. As you well remember from the previous lesson (S01) on the history of artificial intelligence, the dominant approach from the 1950s to the late 1980s was symbolic artificial intelligence ("**Symbolic AI**"), often referred to in the literature as "Good Old-Fashioned AI."<sup>22</sup> This approach is based on the fundamental assumption that human knowledge can be encoded and stored in the form of formal, logical rules and manipulations of mathematical symbols. The most striking example of this approach, which you learned in module S01, was **expert systems**. In these architectures, so-called knowledge engineers hand-programmed complex and extensive decision trees based on logical statements such as "if (condition) then (action)."<sup>15</sup> An analogy to the operation of a symbolic system is the diagnostic process conducted by a car mechanic using the service manual: "If the engine won't crank and if the battery voltage is below 10V, replace the battery." Symbolic architecture guaranteed full transparency and interpretability of the decision-making process, but it completely failed when the rules could not be clearly defined using mathematical operators.<sup>22</sup> These systems were unable to cope with the chaos of the real world, for example, image analysis, because it is impossible to write down the rule "if a pixel is brown, it is a dog."

Engineers' response to these limitations eventually became the connectionist approach (**Connectionist AI**), which completely abandoned the manual definition of if-else rules in favor of virtually simulating the operation of biological neural systems.<sup>22</sup> In this model, knowledge is not represented by logical symbols explicitly encoded by the programmer, but is distributed throughout a powerful structure of connections and mathematical masses within artificial neural networks. The model itself learns relationships based on processing vast amounts of diverse data.<sup>23</sup> Contemporary advanced systems engineering is also witnessing the development of a novel hybrid approach known as neuro-symbolic AI. This architecture strives for an unprecedented combination of the power of precise logical reasoning characteristic of classical expert systems with the phenomenal adaptive and perceptual capabilities offered by neural networks.<sup>24</sup> The goal is to create systems that are not only extremely efficient at processing

unstructured visual data, but whose deductions can be logically and textually explained to system auditors. From an educational perspective, it's like pairing an intuitive artist (neural network) with a rigorous mathematician (symbolic system), who together verify the generated output.

## Machine Learning - a subset of AI

learning ( **ML** ) is currently an extremely powerful, pragmatic, and widely used subset of the broad domain of artificial intelligence in the business world.<sup>15</sup> The most precise, academic definition characterizes ML as a set of algorithms and sophisticated statistical models that enable computer systems to automatically learn patterns from provided data and continuously improve their predictive performance through accumulated experience, entirely without the need for explicit programming of logical rules by the developer.<sup>18</sup> To fully understand the true revolution that machine learning has ushered in, every software engineer must fundamentally reconstruct their current way of thinking about solving computational problems. This phenomenon is referred to in the literature as a programming paradigm shift.<sup>32</sup>

In the approach known as Traditional Programming, which you learned in your first Structured Programming lessons, the process of writing applications is highly deterministic. The programmer receives input data, manually analyzes the nature of the problem, and, based on their own intellect, formulates a set of unambiguous rules (source code lines, if/else conditions, for or while loops), which the computer applies point by point to the provided data to generate the final result.<sup>32</sup> Visually, this classic computer process can be described as a simple vector equation: **Data + Rules** → **Result** . This approach is absolutely perfect and unrivaled in transaction systems, accounting applications calculating tax, or operating system architecture, where mechanisms must be 100% repeatable and predictable.<sup>33</sup> However, traditional code fails miserably when the business problem is simply too complex for any human to describe with a finite set of mathematical instructions.<sup>33</sup> A prime example of this limitation is the attempt to write a computer program that recognizes a human face. You cannot write a series of if conditions evaluating the RGB values of millions of pixels to ensure that a photograph depicts a specific person regardless of the lighting angle.

In the field of machine learning, the aforementioned software development paradigm is undergoing a radical reversal.<sup>32</sup> During the training phase, a computer is provided with vast sets of historical input data directly linked to information about past correct outcomes. The statistical algorithm is then tasked with independently identifying correlations and generating final rules (compiling them into a mathematical model) that best connect these data with the answers.<sup>32</sup> A conceptual diagram illustrating this revolutionary paradigm shift takes the form: **Data + Results** → **Rules** . These automatically generated rules, represented in the system as a statistical model loaded into memory, can be easily used on new, completely unseen data streams during the training phase to predict future outcomes, constituting a key business value for the application.

<sup>32</sup>

The internal architecture of machine learning is divided into three main paradigms that determine the algorithmic way in which the machine acquires, interprets and assimilates

knowledge from the provided databases:

## a) Supervised Learning

This is by far the most commonly used ML paradigm in enterprises.<sup>31</sup> In this model, the statistical algorithm learns from a provided training set, which contains meticulously labeled historical data.<sup>31</sup> This means that each individual input record (referred to as a feature or attribute) has a pre-assigned, error-free answer (label), most often precisely determined in advance by an auditor or a human.<sup>15</sup> The system iteratively analyzes these pairs, minimizing the error between its predictions and the actual correct labels.

- **Example applications:** Binary classification algorithms, such as scanning emails and labeling them 1 (spam) or 0 (solicited), based on millions of historical emails flagged by users. Another group are regression problems, such as predicting the exact price of a specific property by learning from sales history, including square footage, year of construction, and location.
- **Educational analogy:** The process of supervised learning is very similar to a student preparing for a final exam using past-year tests, which have an answer key at the end of the paper. The student solves the problem and then immediately checks the answer key to see if they answered correctly. If they answered incorrectly, they analyze their steps, revising their thought process to avoid similar mistakes in the future.

## b) Unsupervised Learning

This paradigm involves providing a statistical algorithm with raw, completely unlabeled data matrices that are devoid of any cues in the form of labels, final scores, or target responses.<sup>The</sup> task of the operating system is to completely autonomously explore the environment and examine the structure of the data in search of hidden patterns, statistical correlations, or natural numerical groupings.

- **Application examples:** The most common implementation is clustering (multidimensional segmentation), which is widely used on e-commerce platforms to group a database of millions of customers with similar, unnamed purchasing preferences, without management having to rigidly define an ideal buyer profile.<sup>Another</sup> industry domain for unsupervised ML is the automatic detection of numerical anomalies, identifying unpredictable market events, which is a key line of defense in banking departments dealing with complex credit card fraud detection.<sup>17</sup>
- **Educational analogy:** Imagine a situation where you receive thousands of different, unlabeled artifacts from another planet scattered across your desk. You have no idea what they're for or what they're called, and no one has given you instructions. So you begin sorting them into piles, placing items made of the same metal, with a similar cylindrical shape, or with the same weight next to each other. In effect, you've categorized (clustered) the data, despite the lack of any initial labels or supervised knowledge.

## c) Reinforcement Learning

This is the most dynamic type of machine learning, derived directly from concepts in animal

behavioral psychology and based on the continuous interaction of an autonomous, isolated program (called an agent) with a simulated virtual environment.<sup>31</sup> Instead of relying on loading pre-made historical databases, the algorithm executes actions through millions of trials and errors over time. For actions that bring the program closer to its goal, the system awards the program a virtual reward point, and for undesirable actions, a negative penalty (a mathematical function of reward and punishment).<sup>31</sup> With each run, the model updates its policy to maximize the total value of rewards earned in the environment.<sup>31</sup>

- **Application examples: The absolute foundation of decision-making logic for advanced, autonomous walking robotics, dynamic power grid capacity control systems, and incredible achievements in gaming. A classic and widely cited example is DeepMind's AlphaGo system algorithm, which defeated world grandmasters in the** abstractly difficult logic game Go. In more recent generations, this agent trained solely by playing millions of virtual games against itself, without any analysis of past human strategies.
- **Educational analogy:** The operation of this architecture is similar to training a young dog. The owner does not explain the physical intricacies of the sitting process. When the dog accidentally performs a desired action (e.g., sit), it immediately receives positive reinforcement in the form of a treat (reward). When it ignores the command, the reward is withdrawn. Over time, the dog develops an optimal behavior pattern focused entirely on obtaining continuous rewards for desired actions.

In their daily work, a Python programmer solving classical machine learning problems utilizes specific statistical tools. There is a rich set of typical classical ML algorithms, which will be discussed in great detail, along with their mathematical implementation, in modules S16–S19 of this professional course. However, to ensure adequate terminology, it is necessary to mention the core methods, which include: linear regression algorithms, the construction of multilayer decision trees (Decision Trees), noise-tolerant Random Forest algorithms, advanced Support Vector Machines (SVMs) that determine numerical separation margins, and the K-Means algorithm for clustering based on Euclidean distances.

It's worth emphasizing at this early stage of learning that an engineer designing an ML model doesn't write thousands of lines of complex code implementing vector transformations from scratch. In the world of programming, and especially in the Python ecosystem dominated by data science tasks, the absolute industry standard for facilitating the process of creating classic ML models is the open-source library called **scikit-learn**.<sup>11</sup> This tool provides all engineers with a consistent, standardized, and reliable API (software interface). Using just a few abstract commands, such as `model.fit()` to trigger the training process or `model.predict()` to run the prediction engine, it allows for the rapid initialization, training, and deployment of highly effective analytical algorithms in production within an enterprise without the programmer having to bother with detailed implementation of mathematical formulas at the operating kernel level.

## Deep Learning - a subset of ML

learning ( **DL** ) is the most architecturally evolved, highly specialized subset of machine learning.

<sup>11</sup> In this subfield, which is extremely complex to implement, operations and predictive analytics are implemented exclusively through the use of powerful, multi-layer artificial neural networks (ANNs). These networks are architecturally designed with mathematical and logical inspiration drawn from the structure and functioning of organic neural networks in the human cerebral cortex. <sup>11</sup> Although the theoretical and mathematical foundations of the artificial neuron (perceptron) concept were developed decades ago, the prefix "deep" added to the modern nomenclature refers strictly to the engineering structure of contemporary algorithms. <sup>19</sup> This term means that between the software input layer (accepting data from the world, the so-called Input Layer) and the final output layer (generating the model response, the so-called Output Layer), dozens, and increasingly hundreds, of powerful mathematical, consecutive hidden layers (Hidden Layers) are designed, responsible for the extremely abstract and multidimensional extraction and compression of hidden signal features. <sup>19</sup>

Understanding how artificial neural networks undergo the learning process becomes clear after delving into the mathematical architecture and the role of weights and biases. <sup>41</sup> Each artificial network is composed of countless virtual nodes, which are colloquially referred to in the industry as artificial neurons. Thousands of virtual connections transmitting numerical signals run between neurons on different layers, and each such connection has its own numerical value, called a weight, rigorously optimized during the training phase. <sup>41</sup> The weight value determines the signal prioritization in the neural network—it informs the next neuron how important the previous feature (e.g., the presence of whiskers in an image) is in the final object recognition process (e.g., identifying a cat). <sup>41</sup> Complementary to the weight, an integral component of a node is a parameter known as bias, which acts as an internal, programmable bias—it regulates and mathematically shifts the response threshold of the neuron's activation function, allowing engineers to model the network's response much more precisely. <sup>42</sup> The process of "learning" large models is actually a continuous process of running powerful optimizers based on differential calculus. Using a mathematical method known as backpropagation, the algorithm updates the million-digit matrices of these weights so that the output response becomes closer to the historical truth with each training iteration. <sup>41</sup>

- **Educational analogy:** The concept of weights and biases can be compared to the academic grading system for complex final exams. Weights correspond to multipliers assigned to specific tasks – problem-solving tasks have a high weight (determining the final grade the most), while test tasks have a low weight. Bias, in turn, is a kind of starting buffer or minimum conversion factor – for example, a professor's rule that every student simply for showing up for an exam on time and completing a form automatically receives 5 "incentive" points, guaranteeing a different starting point on the curve.

The explosion in popularity of deep learning technology and the possibility of harnessing it for commercial purposes in enterprises, which we have witnessed in recent years, is, from an engineering perspective, a direct result of the unexpected convergence of three independent technological revolutions. First, the galloping digitization of society and business processes has led to the emergence of the **Big Data phenomenon** – we now possess astronomical, historical text corpora and petabytes of visual data, essential as fundamental "food" to feed the insatiable

deep algorithms during their training phase. <sup>Second</sup>, a groundbreaking breakthrough occurred on the hardware side, involving the adaptation and reconfiguration of super-efficient graphics card accelerators ( **GPUs** ). These units, previously used for rendering polygons in video games, turned out to be architecturally perfectly suited for parallel calculations of powerful numerical matrices, which dramatically shortened training cycles of complex models from decades previously unattainable commercially to the order of several hours. <sup>11</sup> Thirdly, engineering and scientific communities have overcome long-standing limitations by introducing highly specialized topologies of connections between layers, removing old algorithmic locks.

As a consequence of the above-mentioned processes, four most important architectures of deep neural networks have been developed, each of which is dedicated to a different challenge (and whose precise mathematical structure will be devoted to the subsequent blocks of classes, lessons S20–S23, respectively):

1. **Dense/Fully Connected Networks (FNN)** : A classic, fundamental architecture defined by tightly connecting the output of every neuron in a given layer to the input of absolutely every neuron in the next layer, determining a one-way flow of information (so-called feed-forward). Dense structures excel at processing stabilized numerical tables with constant, unchanging input and minimal anomalies. Unfortunately, they are extremely primitive tools for interpreting sequential data (changing over time) or analyzing two-dimensional image topology .
2. **Convolutional Neural Networks (CNN)** : **An architecture that is the absolute driving force** <sup>behind</sup> a gigantic breakthrough in digital machine vision. To understand CNN's innovation, one must abandon the traditional notion of connecting all points in favor of the analogy of a scanner or an eye. The mechanics of this structure involve performing extremely fast convolutions on the image sensor, operating with the concept of a slowly moving small "window" (multidimensional filter) that methodically sweeps the entire digital image pixel by pixel. In the initial, shallow layers, this window learns to capture simple mathematical gradients, edge shapes, and sharp lines. In subsequent, deeper layers, it can abstract geometric shapes, combining information in the top layers and accurately identifying a complete object or recognizing extremely difficult, specific anatomical features from facial projections .
3. **Recurrent Neural Networks (RNNs/LSTMs)** : Networks equipped with a highly sought-after "memory" in system algorithms and specialized implementations of signal feedback loops, in which the numerical output vector generated during the previous processor step constitutes a half-portioned portion of the input required to initiate the analysis for the current step. <sup>47</sup> This fundamental structural dependence on chronology and short-term memory favors RNNs in all tasks where relational context and proper temporal ordering are paramount in the search for meaning. They are therefore a native environment used in digital processing of stock market data, the generation of coherent text in sequential order, and the engineering of synthesizers and speech detection. <sup>47</sup> By analyzing text using an algorithm, an RNN retains in its archive a memory of the words seen a fraction of a second ago in the initial sentence.
4. **Transformers** : The latest innovative paradigm, introduced by researchers in 2017, is the backbone of a revolution that has relegated recurrent networks to the engineering

obsolete in text analysis. Unlike RNNs, which analyze time-consuming sequences word by word in a vector mode, transformers are based on an advanced mathematical model of the attention mechanism (so-called self-attention). This paradigm allows the model to process gigantic context vectors or massive files simultaneously and massively parallel. The algorithm generates statistical, contextual strength measures for thousands of overlapping relationships between words in a fraction of a second, regardless of the logical and physical distance separating them within the structural corpus of a given paragraph.<sup>48</sup> This phenomenon is the fundamental evolutionary mechanism that provides the technical backbone for all the contemporary Large Language Models (LLMs) classified as generative artificial intelligence mentioned in the first lesson, such as the flagship GPT-4 from OpenAI, Claude, or the powerful Gemini model from Google.<sup>11</sup>

The design and integration environment for deep learning solutions among legitimate AI developers and architects is entirely dominated by just two undisputedly dominant frameworks, which constitute the untouchable market environment for engineers. These are **TensorFlow** (a platform created and optimized internally by Google engineers), which is often folded into a much more convenient, high-level, modular programming interface in Python called **Keras**. **On the other hand, the PyTorch project**, developed and strategically supported by Meta (formerly Facebook), has recently been rapidly gaining ground in the professional programming market and research institutes.<sup>11</sup> Both TensorFlow and PyTorch offer developers unified, comprehensive analytical environments with implemented computational engines on graphics cards and automatic differentiation engines for vector operations on large tensors, which drastically saves teams time implementing difficult decision loops related to training gradient parameters.

A crucial educational aspect that new ML/AI programmers often struggle with is the absolutely essential need to remember a crucial engineering re-engineering: in IT and business environments, deep learning is by no means always the most appropriate and best replacement for classic, seemingly outdated machine learning algorithms. Due to the extraordinary wealth of hundreds of millions, and sometimes billions, of optimized weights in a nonlinear structure, powerful and difficult-to-audit deep models become extremely sensitive and highly susceptible to the phenomenon of predictive overfitting, known from error theory. This phenomenon occurs when such a sophisticated and information-hungry model is fed by analysts with data from a remarkably narrow perspective in the form of small, sparse corporate statistical datasets.<sup>23</sup> To simplify the issue to a simple formula, instead of generalizing, the model learns the entire provided set of samples by heart, including anomalies, completely losing its flexible rationality to externally supplied data during the commercial use phase of the system. The determinants of this phenomenon, along with the extremely high financial thresholds related to the colossal cost of powering graphics cards (renting GPU clouds in a provider's infrastructure such as AWS for the duration of the calculations), mean that when modeling multidimensional business conditions with a fixed numerical base, older classic tools based on, for example, the Random Forest architecture from the scikit-learn package prove to be statistically more accurate, cost-free, and resistant to anomalies, and their computational logic poses no difficulties to the engineer responsible for maintaining the business platform's continuous operation.

## Relationship diagram: $AI \supset ML \supset DL$

Seeking to once and for all structure the vastness of defined and systematized theoretical concepts, the artificial intelligence engineering and data science community has adopted a visualization-friendly diagram that is a graphical equivalent of the figure of concentric Euclidean circles. A very popular term invokes the image of traditional, smaller Russian "matryoshka" nested in successive shells larger than themselves by the geometric projection of their counterparts.<sup>17</sup> The largest, universal conceptual boundary circle, placed on the outer shell and covering the entire conceptual territory defined in the early phase of the field, represents the area described as **Artificial Intelligence (AI)**. Within the boundless interior of this logical envelope lies a significantly narrowed technical framework, isolated by domain specialization, the second smaller circle in the evolutionary hierarchy, assigned the name **Machine Learning (ML)**. At the very core of the formed classification diagram, nested tightly within the machine outline, there is this extremely exclusive, precise in application programming apparatus describing the use of complex convolution networks and layered matrices, outlined as **Deep Learning (DL) representations**.<sup>11</sup>

The strict set relation resulting from the graph unfolds the entire fundamental, hierarchical truth of the system before the programmer. Above all, it establishes an inviolable scientific dogma for teaching purposes: without exception, any software using and generating an architecture designated as deep learning is considered and is identical to machine learning solutions (since the DL matryoshka is 100% within ML). Following the logic further, any use of classical machine learning vectors falls under the definitional banner of artificial intelligence. A crucial requirement of being an IT architect is to understand the conditioning of the fact that this definitional entailment relation constitutes a fundamentally and unprecedentedly asymmetric equation. Absolutely not all implemented algorithms and compilations defined as artificial intelligence systems engage and utilize any form of statistical learning. The tangible, historical, and still-functioning corporate proof of this is the systems discussed, which rely on manually formulated principles of classical programmable symbolic logic based on if statements, deliberately ignoring the learning stage based on input parameter weights.<sup>22</sup> On the other side of the hierarchy, not every analysis script modeled with code classified as machine learning must harness the burdensome and architecturally challenging arrays of mathematically complex deep layers included in DL, as evidenced and confirmed by the usefulness of a system solution based on multiple regression vectors, which is satisfied with ML simplification.<sup>31</sup>

With the advent of digitalization, this traditional academic "circle" model became the embedding environment for much younger technological advancements and cloud-powered innovations that intersected disciplines within the diagram. Establishing topological points within this framework, additional taxonomic branches emerge:

- In the field of commercial technology, task-oriented concepts such as **Natural Language Processing (NLP)** and engineering algorithms responsible for digital **Machine Vision (CVM)** refuse to be confined or flattened, operating within a domain limited solely by a set of specifics. They constitute interdisciplinary task domains that intersect both the analytical potential of conventional, mathematical ML (where the classic programming solution in

CVM is loop-based logic and the principles of a simple optical OCR operating system that statistically extracts letters from scans), and the revolutionary vector structures dedicated to a circuit based to the brim with powerful DL annotation layers (used as a foundation in medical architecture for automatic algorithmic diagnosis of oncological matting from multidimensional arrays of generated MRI images, harnessing vector pixel convolutions) .

- The now-marketed and much-vaunted **Generative Artificial Intelligence (GenAI)**, due to its action vectors, locates its technological territory in a precise core, nestled in the narrowest center of the diagram, utilizing the full, previously insatiable power of the statistical deep learning paradigm. Constituting an evolutionarily unprecedented, highly specialized subset of artificial generation software, algorithmically focused solely on innovative mathematical synthesis and the resulting composition of entirely new content representing top-down vector structures (a logically coherent block of multi-page analytical text, vectors with hyperrealistic visual representations of photographic matrices while maintaining light depth deviation, or a coherent and compiled sequence with optimized and functional C++ source code), its task is most often based on the extraction of rules connected in innovative attentional networks subject to the phenomenon of stochastic, random propagation of digital parametric diffusion .
- The very concepts of the powerfully promoted and corporately familiar **Large Language Models (LLMs)** find their territorial anchor precisely situated in the strict and technical heart of the software representing the GenAI model, carving out a separate architectural island within it, strictly reserved for the operation of the most powerful models in terms of the structure created on disk and the capacity of statistical weights, operationally classified and fully targeted, with the architect's premeditation, at vector processes simulating multi-level and statistically related mathematical communication coherence and the operation of the natural sequence of grammatical signs of human speech in a sign-based interface .

<sup>exam</sup> . It differentiates fundamental definitions with a precise reference point based entirely on standardization requirements for technologist roles, drawing on the most useful and crucial engineering classification criteria for advanced integration programmer positions in code design and optimization.

<b>IT Design Comparison Criterion</b>	<b>Artificial Intelligence is generally categorized as (AI)</b>	<b>Programmatically classified as Machine Learning (ML)</b>	<b>The engine is classified in implementations as Deep Learning (DL)</b>
<b>A core and defining architectural</b>	Fundamentally, the broadest form of conceptually defining	It is a rigorous form of optimized coding that bases the model and logic	A reserved, extremely restricted structural subfield of ML, whose main

<p><b>definition</b></p>	<p>technology's concept of programmatically simulating, with an arbitrary, code-based script, a generalized analytical computation mechanism that replicates the abilities of human reasoning patterns in abstractly solving cognitive tasks. <sup>11</sup></p>	<p>engine construction paradigm not on if statements, but on vector-based operational learning of analytical algorithms from a given environment, supported by the display of statistical tables with correct solutions</p>	<p>building block of the statistical engine is the mathematical organization of dozens and layers of perceptron cells, forming and precisely mimicking the architecture of the neocortex in biological evolution, designated as ANN. 18</p>
<p><b>The necessity of assimilation in the context of the conditions of the data provided</b></p>	<p>Marginal input dependence on capacity. A very large group of logical solutions, based in logic on the pre-defined principles of a rule-based system, has complete freedom to operate as an independent entity in the system environment with minimal and rudimentary exchange of input parameters.</p>	<p>A scalable and operational requirement for precise input of multidimensional structured database sets prepared on the ETL side (often requiring implementations after operational tests from the error generalization threshold on a minimum of several thousand verifiable CSV tabular records) and meticulously labeled results (supervised learning).</p>	<p>The ruthless application of architectural rigor to the standards of gigantic Big Data structures and underlying data streams. The imposition of unprecedented implementation of massive, unformatted data, numbering in the hundreds of thousands of observations, in a raw virtual form without any defining features (e.g., gigabytes of visually and pixel-compressed images)</p>
<p><b>Demand for the volume of the</b></p>	<p>The level of optimization ranges</p>	<p>Predictable average cloud engagement.</p>	<p>The extremely acute and drastic</p>

<p><b>hardware computing power environment</b></p>	<p>from negligible to moderate. Older, classic operational AI input scripts, which build solutions based on rule-based predictive reasoning based on scripting language structures, compile tasks natively, operating on a unified environment of traditional and unsupported single-CPU units.</p>	<p>Training a mathematical regression vector impacts programming demands, requiring standard support from multi-core processors capable of parallelizing the calculations performed in the processor's server memory. Often, a desktop PC is sufficient for running the program.</p>	<p>need to physically provide a rarely available volume of enterprise infrastructure. The powerful parameters of convolutional logic and hundreds of thousands of multi-layer mathematical operations to be loaded dictate the compilation process for clusters built and specialized exclusively on expensive processors supporting vectorized computations, such as visual processing units (GPUs)</p>
<p><b>Logical parameter for assessing the interpretability of the mechanisms of the decision made by the server</b></p>	<p>Extremely high, providing audit-proof logic. Reliable and predictable insight (white-box transparency). With the requirement and conditional loops explicitly overridden in the source code file, the engineer will document the final predictive basis of the logical mechanism, piece</p>	<p>Assessed in programming metrics as a model with a partially moderate level of insight. The system, in analytical architectures, from statistical models in regression, attribute selection in PCA analysis, to structural optimization of decision tree logic, maintains partitions for coders, after</p>	<p>A fundamental diagnostic disqualification describing the concept known in computer science as the so-called closed-black-box model. Calculations conceal billions of interdependent micro-input vectors, which is why the final calculation in the optimizer, which uses logarithms, gives the</p>

	by piece.	which, in predictive analysis, the auditor will examine the relationships between the output data and their role in statistical forecasting.	programmer a negligible or logically difficult interpretation of the final mathematical result.
<b>Examples of standard target optimization algorithms used in programming models are common.</b>	A scripting, implementation query language conceptually marked using, for example, mathematically logical language operators, A* mechanisms, rule-based operating systems built by an engineer with no possibility of learning from a collected set of vector data from the world.	Model-side programming based on vector operation using popular scikit implementation engines such as Random Forest, Multiple Regression with plane equation, boundary verification in vector variant of SVM or estimating deviation using logistic prediction loss function.	Environments harnessed with libraries modeling the multi-layer matrix space in the form of architectures for convolutional networks (CNN), networks equipped with the properties of loop historical memory from the past in iteration w (LSTM/RNN), or fractionally trained Autoencoder networks functionally based on the compression density of matrix features and topologies of vector models based on the mechanics of Transformer remarks.
<b>Detailed category of analytical precision of the assessment of the conditions for the</b>	The parameters defining the set and constituting the base objects of the desired evaluation	The need for rigorous, often tedious, manual selection of attributes in the	Revolutionary optimization that relieves the programmer of manual input,

<p><b>emergence of mathematically unique features</b></p>	<p>focus of the vector modeling the logic at the input are determined from a technical point of view rigorously and without the possibility of change, formulated into the model only by the intervention of an expert specialist.</p>	<p>designated preprocessing stage, implemented by implementing developer code on the database analyst side, and thorough processing of database columns (removing empty features from tabular CSV reports) that the model will ultimately externally consider for its results</p>	<p>featuring an engineering environment of deep algorithms, providing an uncompromisingly powerful engine exploring matrix layers that models the feature (on layers) entirely in its own mathematical software and abstract filter approach, eliminating the physical manual intervention of the programming analyst.</p>
<p><b>Operational scope and outline framework during the conditioning process and target implementation verification</b></p>	<p>The level of technical ease of implementation on the project is physically reduced solely to the framework of defining the rules of logical concepts of the input script with code-optimized integration into standard binary repositories, which reduces the production time for the architecture-based module to fractions of days or hours.</p>	<p>Optimization costs and implementation-specific time frames are logistically calculated based on multi-week processing cycles, which are dictated by the extremely detailed manual correction required before test implementation, including cleaning and statistical compensation of raw table deficiencies before training begins, until the statistical prediction of the</p>	<p>Integration cost estimates often run into unprecedented levels in the AI domain, often projecting multi-million dollar infrastructure budgets for servers. Powerful optimization loop calculations related to the requirement to model networks of tens of billions of variables take up long, continuous, and lag-free processes lasting several weeks on vector-based and</p>

		engine logic from the library stabilizes	the fastest enterprise clusters with GPU architectures and arrays.
--	--	--	--

## IT Industry Applications - Mapping to Real Business Problems

Given the market realities and integration challenges modern programmers face daily in their companies, the decisive factor in the success of an assigned commercial project is rarely the memorized ability to independently, analytically program hundreds of abstract, multidimensional mathematical equations or define vector derivatives on matrices to formulate a perceptron structure from the very algebraic foundations with a zero value. The main axis and decision-making power of a valued programmer in the AI engineering market is currently the ability to create logical architectures, which essentially rely on the ability to map, search, execute commands, and connect highly stable, enterprise-tested base models and external libraries already existing in the IT ecosystem to structured environmental goals imposed and required by business decision-makers within a given corporation. <sup>6</sup> Every engineer designing today's system must constantly and expertly classify the precise technological constraints they face in their assigned challenge—from characterizing the available database computing power and matrix volume to defining the need to audit software predictions before legal regulators, to protect the development team from financial failure later on or from software being blocked by the impossible technological limits imposed on the system. <sup>6</sup> From this educational perspective, learning how to correctly classify a field within a taxonomic framework is invaluable.

The meticulously developed, reliable summary below, structured in a clear comparative panel, aims to educationally map the extremely diverse and horizontally dispersed market integration challenges defined in everyday corporate work in individual market sectors of the economy, and to align them logically and mathematically with a specialized, theoretically sound model engineering paradigm and a recommended, usefully dedicated coding tool structured in the solution server .

<p><b>Domain and Business Industry Context</b></p>	<p><b>A specific business problem formulated before the project for the Analyst Programmer</b></p>	<p><b>Locating Logic to Assigned Category and Taxonomy</b></p>	<p><b>Recommended from External Libraries Coded Solution and External Implemented Architecture Integration Tools</b></p>
<p><b>E-commerce, Retail and Digital Segmentation (Marketing)</b></p>	<p>Systematization and optimization of the matrix division of a database of 500,000 fully isolated buyer profiles without demographic data into groups dedicated to specific, personalized advertising discounts to increase the targeted and calculated margin basket of visitors.</p>	<p>Paradigm Designation from Logic: <b>Machine Learning - Classified and operating in space as Unsupervised</b> .</p>	<p>Invoking external script estimation from a library in an implementation of a modeling algorithm that clusters in a virtual matrix space, e.g., the system and statistical distance function <b>of the K-Means algorithm</b> . The complete lack of pre-defined target reference labels for the set, native to this method, allows for completely free, automatic, analytical self-organization of data around vector-assessed distance features and similarities found digitally in shopping preferences to select anonymous consumers from an assigned cluster on a mathematical</p>

			graph projection in order to provide a discount .
<b>Clinical Medicine Research Practice, Health and Infectious Risk Assessment</b>	Systemically and autonomously operating analytical processes based on extracting perceptually difficult anomaly patterns from extremely blurred tumor data matrices after processing multidimensional patient X-ray scan files from radiologists.	Paradigm Designation from Logic: <b>Deep Learning - A software module specialized as engineering Machine Vision .</b>	The implementation foundation of the system, architecturally based on matrix logic via embedded frameworks, provides logistical support for a deep <b>neural network built on a Convolution Filter matrix (an algorithm widely known and commonly implemented from Tensorflow library models, often referred to as a CNN network) .</b> This model has unrivaled, mathematically powerful vector capabilities built into multiple logical levels to capture highly complex nonlinear morphological relationships between cells and pixels, and indistinct tissue structures for identifying tumor projection boundaries, isolating the cellular

			<p>form of the diagnosis from noisy input from medical chest radiographs and on the matrix grid with a high positive statistical error rate exceeding the accuracy of expert ophthalmological perception of the diagnosis in the work environment .</p>
<p><b>Current Financial Technology Sector FinTech &amp; Standard Retail Banking</b></p>	<p>A current, dynamic and rigorously verified analytical solution to the equation estimating the probability of return on a multi-million transaction, estimating the burden to the risk of default on repayment included in the loan for a potential developer, conditioned by analyzing a large vector of numerical historical banking information in a set of database attributes contributed over the years, in terms of portfolio mathematics, taking into account age, earnings logged in</p>	<p>Paradigm Designation in Logic: <b>Machine Learning - Algorithmically Operating with a Supervised Classification Model .</b></p>	<p>A statistically reliable choice for implementations remains the classical optimization distribution software based on an option for decision analysis implemented with operational statistics commands in sklearn libraries in natively embedded <b>machines built with logistic distance and boundary classification optimization vector machines, commonly known as SVMs, an embedded solution for estimating the Support Vector</b></p>

	<p>the referent's PIT, and paid installment dates from tables on previous multidimensionally recorded in a CSV matrix of previous historical agreements concluded with lenders on the examined digital transaction market over five years of statistical analyses in economic indicators.</p>		<p><b>Machine (SVM), or the common linear Logistic Regression architecture, defining a logical vector into two outcomes</b> —the program and system engine of this combination, performing calculations that perfectly classify statistically numerical predictive solutions for the output axis (acceptance or the final categorical form of strict rejection) and generating failure-free scores on a programmed and historically established database table of dimensions with insurance market financiers</p>
<p><b>Market Space and Corporate Engineering in Video Games, Game Simulation Market and Modern AI Digital Entertainment</b></p>	<p>Sophisticated algorithm integration processes, based on input commands, mathematically model the decision-making process for the actions of an artificial virtual</p>	<p>Paradigm Designation in Logic: <b>Machine Learning - Working with a behavioral model in an environmental approach to learning digital logic on a matrix</b></p>	<p>Script libraries adapted and structured in the architectural code by a system based on the implementation of resources for logic based on a complex but evolutionary paradigm</p>

	<p>player, algorithmically programmed as a driver for an autonomous entity (a so-called bot). A programmed player, digitally thrown into a vector-based physical environment on rally tracks, can minimize seconds over time by logging dozens of route loops, avoiding slides and drifts by deliberately optimizing the loop, and thus become the most inhumanly flawless and virtuoso controller of the input pad system, achieving the pinnacle of gaming performance in virtual timing.</p>	<p><b>Through the so-called Reinforcement and Repetition Optimization.</b></p>	<p>introduced by the equations of the mathematical phenomenon of <b>Deep Q-Learning known in the code (DQL) using estimation on a virtual robot - RL.</b> — In this type of implementation, embedded in the environment, the agent-based algorithmic module improves its logical programming skills evolutionarily, or more precisely, through repeated iterative processing over millions of vector runs. Over time, it controls and forces changes in the matrices, forced by the input environment. It is punished from zero by receiving a rigorously and unconditionally negative evaluation in the mathematical parameter, a "penalty" for hitting a static logistic texture with a barrier. After verification, it receives the desired positive value parameter in the</p>
--	---	--	--

			<p>matrices, i.e., the "reward," realizing that the fractional cycles completed allow for a shorter time to reach the winning position without the instructions of a human engineer, which adds a new level to the game. <sup>31</sup></p>
<p><b>Network Information Security Industry Cybersecurity and Cryptographic Protection Analytics of Sensitive Data</b></p>	<p>An autonomous, extremely dynamic algorithmic mechanism for multi-dimensional operational continuous scanning of log tables with a continuous system based on mathematics and immediate operational rejection and ultimately automatic detection of extremely vector-based portal entries, difficult and non-standard login logs generating strange anomalies, ATM transactions and logistically undefined numbers of sudden avalanche requests using foreign servers or unusual SQL system query</p>	<p>Paradigm Designation from Logic: <b>Machine Learning - Statistically in an environment used and working on vector axes as an absolutely model in its entirety Unsupervised for anomalies.</b></p>	<p>Programming modules written specifically for cloud environmental protection that cuts off packets using extremely useful ready-made algorithm instances based on optimization trees in system detections, called industry software for <b>algorithmic, analytical, and multidimensional vector detection of non-standard anomalies, conceptually called structure-based scripts by programmers (Isolation Forests in translation, i.e. Isolation Forest modeled with optimization</b></p>

	<p>code injections in the system matrixes and blocking them to stop them in the bud, potentially indicating in early analytical logic a powerful multi-vector and network cyberattack aimed at a key server running an e-commerce platform, commonly known as DDoS.</p>		<p><b>distance).</b> — a sophisticated engine running on ports in the model's operational code slowly learns, over many weeks in the cloud, to analyze and learn the basics of what is desired from the conceptual vector calculus of logs, absolutely healthy and recognized as normal in the network pattern for daily repeated input, indicating a safe peak in server traffic after defining this point in isolation after time, ruthlessly operationally cutting off this numerical query that rapidly and with unexpected force on the graphs deviates radically and unexpectedly from the assumed mathematically and vectorially approximated safe server curve from the averaged norm previously adopted defensively on servers to cut off an unknown DDoS signal. <sup>11</sup></p>
--	---	--	--

<p><b>Corporate Operational Management and a Modern HR System Module for Engineering-Scalable Recruitment Processes for the IT Market</b></p>	<p>Powerful and multi-parameter analysis in statistical selection of the document database, relying on algorithmic scanning and optimization analysis, extracting scores for table arrangement from tens of thousands of incoming emails, completely entered in the logistics database, unsystematically arranged and sent to platforms and logically unordered into tables of hundreds of multi-page professional CVs from unformatted text in CV format, to subject them to the structure of operational text, programmatically reading them from applications, fully analytically and as far as vector models are concerned, coherently multidimensional, so that reading with context, logically and selected, only defers applications to engineers who</p>	<p>Designation of the Paradigm from logic: <b>Deep Learning - Multi-layer modules highly optimized for extracting linguistic structures designated as NLP in Processing huge volumes and corpora from natural Natural Language and vector semantics of words.</b></p>	<p>The architecture implemented on the market for multi-level system and matrix applications of vector scanning engines based on block extraction, recommended architecture for corpuses based on vectors and attention in the <b>technical matrix of the revolutionary Architecture modeled on attention mechanisms (Self-Attention) in modern Transformers, the usability of which is induced by rigorous word-based structures based on requests from libraries with predictive and analytical libraries of the latest generations of large and evolutionarily advanced powerfully modeled Language Models in structures known by the acronym with the</b></p>
---	--	---	---

	<p>have basic knowledge of the rigorously imposed many conditions by the architect querying the system, rigorously collected, specific, advanced by working technology developer stack in logistics with rigidly imposed, highly constricted, highly logically from programmers and requirements.</p>		<p><b>definition of LLM, harnessed precisely operationally for tasks and for the purpose of prediction about a given task with a categorization projection.</b> Modules implemented based on LLM structures with extremely high levels of context capture after words handle the complex statistical model with exceptional accuracy in extracting semantic structures and table formats into the database (known in operating and programming systems, a script specific to programmers described by the so-called Data Information and Data Extraction algorithm for reading from extremely ambiguous input corpora of written text, often in natural linguistic chaos, syntactically inconsistent from digital</p>
--	---	--	---

			documentation of files entered with matrices into a file generated by PDF readers to obtain and extract data logically into the code). <sup>11</sup>
<p><b>Integrative and Multi-Modular Modern Engineering and Authorization Security of Commercially Scalable Architectures for Mobile Logical Device Applications on Platforms with Usability.</b></p>	<p>Mathematically complex login architecture coded on the system device, intelligent marking mechanisms to assess security through a biometric process on a hardware-based operational security mechanism for scans, determining the authentication and, in the process, securely recognizing anatomical features to assess the rigorous geometry of the vector matrix points from the user's face, replacing the architectural projection with a fraction of the time security position and entry from the banking industry, previously used for logins as its older brother, which was</p>	<p>Paradigm Designation from Logic: <b>Deep Learning - Extremely advanced positioning engines from the field of algorithms based on the Vision spaces in the vector designated by the term Machine Computer Vision for pixels on multi-levels.</b></p>	<p>A combination of modules called by the input layers of the package's architecture codes, which invoke the process of determining the distances on the network of multi-layer deep logic <b>positions with a connected matrix input, determined engineeringly from the network under the conditions defined by the dense connected architecture with built-in algorithmic rigor and natively compatible engine environment, in cooperation in operation with a large convolutional mesh applied with a model for the convolutional layer in codes called as the</b></p>

	<p>entered by hardware in the old environment, was the old multi-character authorization standard security with on-screen PIN entry on buttons, encryption and the use of an outdated highly encrypted manual entry. Authorization PIN protects entry with an operational request from the operational banking procedure, triggering the module in the operational process after evaluation, successfully unlocking the screen from procedures and acceptance of quick transaction launch in payments, commonly used after the module for acceptance of contactless logic with a wallet incorporated into the digital process.</p>		<p><b>convolutional architecture in filters and classically known by developers as a shorthand (matrix with class from a CNN network).</b> This vector layer, analytically based on powerfully modeled pixel logic, predictively vectors on layers of logically key points, called by graphic designers multidimensionally inscribed for software as points identifying three-dimensional nodes from geometry on the matrix grid, logged by the verifier for the entire structure of physical rigor with dimensions of point morphology for the human anatomically delineating facial features from a scan, performing mathematical evaluation. the rigor of the geometry points and, when processing the matrix, to correlate the reference model at the output with the exemplary one</p>
--	--	--	---

			and the database encrypted with the log on the server, physically marking the consent in the system and acceptance from the phone with vector and PIN authorization. <sup>17</sup>
--	--	--	--

From the above engineering studies and compilations for coders, a coherent, educationally structured IT picture emerges, structured for years of work ahead of you - a modern and logically professional programmer rarely runs away from the industry markets in order to avoid server breakdowns, and generally very rarely in his achievements stands alone with a coding test about physical operational entry from the zero vector, to manually program physical and multidimensional vectors on the ordinate axis with mathematical rigor, defining individual complex parameters in the formula for deviations and vector matrices from known statistical weights for calculating in the construction of multi-layer derivative ordinate equations, to build from scratch for the server, powerful system-wise mathematical deviations operating in the logarithmic library in the PyTorch engine for the needs of artificial vector, to build neurons for networks and entered at the output, to run the network from scratch with fractions of the derivative and bias. <sup>6</sup> His operational responsibility for server clusters and the system for market analysts at companies for implementation in companies building successful environments with entrusted capital rests rather after implementations, relying highly on logistics after optimization, on an unwritten selection principle in the architecture, and precisely after testing in engineering-focused environments, relying on structured precision in making options through choices, i.e., intelligent and rationalized engineering construction in the implementation of system packages for environments on rigorous selection, calling software from extremely extensive logic repositories with packages ready, tested for months and environmentally verified for databases and selected highly optimized for errors, to invoke them in code in stable calls, with scripts developed and integrated in vectors at the manufacturer, to include built-in libraries in dependent architectures after optimization of databases with hardware resources, with a cloud projection facility regarding vectors in server rooms and electricity costs from hardware for setting the limit of the cloud database and available after optimization operations at Volume on arrays to assess the time needed for integration with developers for renting a high-powered server on the infrastructure graphics, calculated according to the budget in implementations such as the above and natively enforces the option rigorously for software models after analysis based on business findings. This imposes on teams a highly structured, operationally defined, and requested character after testing from challenges for the entire vector problem from the architecture at the targets for the platform with the desired module, enforced by rigorously delegated verification of the problem task and integration logic logically determined in the corporate specification for database implementation. <sup>11</sup>

# Common Mistakes and Conceptual Pitfalls in Software Architecture

The unprecedented dynamics of conceptual changes on such a scale, driven by the explosive trend in engineering fields researching and optimizing artificial intelligence in modern company logistics, fueled by the power of disinformation on the part of PR agencies, suppliers in modern departments, and agencies outside the industry about solutions, has led to unprecedented market developments and the generation of a huge number of fatally misleading abbreviations of concepts due to disinformation and simplifications harmful to the understanding of educational environments, shortened to guideline vectors, to abbreviated concepts with incorrect ones, which obscure the perception of operationally correct assessment in systems to reliable assessment of the base reference and classification under the implementation of vector logarithms to diagnose rigorously for the needs of the environment in a complex projection from the concept of the problem. Rejection of marketing buzzwords and extremely important before failures with understanding of system goals in early market implementation to prepare the process for years of knowledge is with the system of challenges. The challenge posed to developers is a rigorous and absolute necessity to recover and in-depth substantive concepts about disinformation in operating systems in market implementations after unblocking the logic for coders. The so-called demystification on the market of those given about these erroneous names from statistical to market slogans in contemporary and popular public opinion reproduced as powerfully and poorly named and conceptually conditioned logistic options to enter the concept of engineering behind the vector before the loop safe with codes under the condition in entering into their own through the course thrown into the system with the systemic rigor of programmers on the path to rigorous environments and after entering much deeper in the architecture in early logistics allocation to understand the field behind the concept of structure for the developer in an extremely logistical after entering the rigors conditioned strict from developers industry by specifying in the early phase of the logic for system development after algorithmic logistical structures with the arrangement with the environment in the architecture and optimization for the database in early software in the code in the structural for logic with reference to older coding concepts in early scripts for logistic verification and operationally correct marking of algorithms error-free for future implementations in cloud engineering with integrated and rigorously described and optimized statistical operations from the architecture in patterns for databases and logistic determined in the code for projection to the learning server calculated on models for logic for machine drawing with parameters.

Firstly and most importantly, to note under the rigour of logic as the most painful, widespread after errors logically and systematically to concepts, rigorously conditioned for the layman, as confused in the media, dangerous to verification, thrown for mass implementations, falsely conditioned due to distortions in the media, the concept is considered in a popular approach, wrongly affecting logic is the projection into evaluations, perception after prior analysis, affecting in the common consciousness of perception of the powerful vector logistics of the field to operationally evaluate predictions from the designated broad taxonomy as and by definition to evaluate logarithmically to artificial broad under the domains under the umbrella as of intelligence in the common and harmful highly rigour to synonym for conditioning after

abbreviations in the form of as physical AI personified in vision as incorrect to vector from vision initially fixed synonym in references to construction under physical to implemented in films from logic from vision in humanoid or for industry physically conceptual physical and rigorously to mechanics and complicated in the mechanism at the input at the reference about the form from physical from mechanics automated in operational robots in the implementation of algorithms for robotics in projections with consciousness in machines logistically and physically embedded in autonomous robots with a shortcut to wrongly refer to a robot from a physical form to a robot. <sup>6</sup> The architectural truth is completely different: purely based on matrices for vector environments and server models, statistics from logistic software, powerful scripts for operational software, conceptual algorithms for emergency calculations, algorithms from the field of conditioned script databases under the rigor of predictive learning from logic from rigorous machine databases to function on statistics for calculations for the server, from logarithmic input in support, they do not influence calculations to be in rigor in absolutely conditional on the needs, not affecting not to optimization and in software, not and due to the input and to physical to vector in the environment, they need to rotate calculations of logic in general in their operational approach, about physical from implementations of the casing under the cover of built at the database at the server and outside from the base from the building material to encapsulate and rigorous option from input programmed after being conditions from the existence in the physical determinant for the software of the body in the operational architecture of the machine from the machine to the processor from operational mechanics to optimize on tables and not on the body to be able to vectorially in scripts, be able to predict without disruptions and calculate matrices statistically, projecting them onto tables efficiently and operationally for operations on tables with vector data and parameters, not on the physical world logarithmically in the process in environments in logic and through databases and operating do not impact on, and what rigorously in the code, so that after exiting the logic from within after calculations, systemically demonstrate for the developer on servers and projecting, manifest logistically conditioned for implemented in vector and embedded under "intelligent" in statistics predictive for the program and complex from vector with input decision patterns in implementation rigorously from evaluation to the server; These algorithms are based from scratch on statistics at the processor and vector-wise on the input projection from the software and operate in logistics from scripts at the processor rigorously and hidden in concepts entirely due to the vector processing in the system using logic and operationally calculate with rigorous cut-off on systems purely in code as well as operationally systemically exclusively and logically under cover digitally on virtual machine servers under logic operating on infrastructure at suppliers in modern systems in the backend option and servers in the cloud under strict and logistically on networks hidden in abstract from the pattern and multidimensional conditions from optimization in lines and files for input and embedded conditions in the written and conditioned in the code and files from implemented on replicas and source databases under strict and system codes from optimized under conditions to commands on commands conditioned on the backend under server logic to the backend in called as backends from scripts from operations from the backend for platforms for operations with a projection for the operation itself from software in implementation from specifications from application construction in implementations to code. <sup>6</sup> Equipping highly advanced logistics from built-to-logic for virtual and abstract environments from scripts on rigorous architecture solutions on vectors to cybernetic ones with optimization in physical

conditions for systems based on hardware for mechanical propulsion, conditioned by motors on arms and drives as complex and physically implemented from implementations with scripts for operations actuators on machines for propulsion from hardware housings in systems to outputs to sensors in options in built-in physically operational in physical and conceptual cameras for logistics from concepts based on architecture from built-to-robots and physical in implementations in concepts exoskeletons in logistics on machines in shields for embedding. This highly concerns and rigorously different outside from implementations from the delineated housing domain in engineering conditioned in implemented in another completely in the domain of the concept and rigorously about logistics in another field and specialized and going beyond AI in the field in engineering from concepts to mechatronics in robotics in the field of physical from engineering for software for physical operational from conditioned domains in engineering from AI physical conceptually for logic under physical domain in operations in hardware for housing, while in the environment itself at projection for vector at hardware-independent prediction or optimization in logistics itself in the vector axis of engineering for logistically in from software in implementations from sketching at projection in servers and from input from operations from software and algorithm for sketching to the conditioned model as and in itself for the base as and pure from codes in coding at logic and projection from software and from itself from matrix under logic for mind from operationally from verification and operationally with logistic in implementations from algorithm and extracted from analytical logarithms from implementation to concepts as determined from input and hidden to logarithm for optimization on clouds for embedded in external and from server in implementations to backend then under logarithm in options selected from concept and marked from implementations in engineering in rigor of the domain of operations in pure logic selected for the very concepts in the domain for AI in systems in computer science for AI without bodies. <sup>6</sup>

Secondly, a pernicious and highly perpetuated misinformation in early corporate environments, based on vectors from the database, echoes in construction to projection on the architecture from conditioned on inputs in new from software at solutions for systemic implementations in implementation from implementations from the database. This is highly harmful in early logistics and erroneous from conditioned from options in projects in early in management from verification in rigor in option that expensive from the cloud budget and logistically AI implementation imposed on the team in implementation on input to script from operations at input from models at calculations for software at learning in implementations with conditioned in ML machine learning to servers from vectors for the project is always to logic without exceptions on input and out of necessity AI is mandatory in operationally from forced on markets in environments in rigorously under the pretext of modernity in requirements and from goals under logic from modernization to logic for cloud systems and for projects conditioned to optimization and from assumptions in the system options that in implementations from the model under ML from the input in the models from the principle in the operational option from the rigor from the assumption at the input in implementations and vectorially to the projection always and logistic and absolute in the option in logistics in implementations will be at servers with better optimization on logic for models in implementations from the matrix it at the output from servers in projections had in calculations an unquestionable from the results and rigorous at the quality on the projection in verification and an advantage from calculations at the quality on the outputs and qualitative over simple and from codes and classical from verification with if-else in

implementations in the traditional rigor and marked to the rules and in conditional in projects model on the rules in logic at conditions in decision codes from the code under the rules in implementations conditioned in coding. <sup>6</sup> Often in engineering implementations, in early implementations, very simple and needs-based coding situations occur for developers, from optimization in projects, in architectural implementations, to logic, and for projects in engineering, in servers, in which verification, from optimization, would be rigorous, and from the results at the output, would be efficient, and in options, conditioned from verification after input, from coding, conditioned in procedures in calculations, concepts as simple in codes and native in logic at input, and in if-else statements, under verification on conditions in procedures, in evaluation, from verification from scripts, and conditions in procedures (such in structure from coding as conditioned on conditions in codes, on if/else options from logic in scripts, declared in codes in construction, in structures, in arrays in code from construction, and associative, conditioned in structures, or about queries in conditional, fast in operational, in calculations on tables in SQL, joins, and operationally from databases on servers) in implementations with Calculations are completely and in the operational rigor from calculations and under verification from the effectiveness unambiguous in calculations and in the output from verification with effectiveness for verification in the logic and about the result with verification on inputs and procedures conditioned by verification, in the operational from implementations in immediate on outputs in calculations in terms of and logistically to operations in implementations with software and calculations and at the output at the server and at outputs under the input and completely under the budget from verification on projects and from costs or cost-free to logic in the rigor from the output in the project with verification in the perspective from clouds with options in implementations at inputs and procedures in implementations in options by way of operations in procedures in implementations with verification to projection in implementations on the environment in procedures. <sup>34</sup>

Thirdly, it is not uncommon for graduates and newcomers to logistics from verification of engineering concepts and rigorously in systems from options in systems in implementations after tests on projections from models and rigorously and for beginners from rigor in implementations it seems wrongly conditioned in logistic in implementations about calculations after tests from procedures that in implementations on powerful in calculations in logistics and highly in calculations on inputs conditioned from statistics on clouds and learned deep with vector in logarithms in calculations for prediction about models from libraries conditioned on options in ML to input they can to the model from operations after from the model with they understand in abstract about concepts in concepts logistic in input under verification in the model from concepts about context in abstract in conditioning after input on calculations in operations in implemented in the model from calculations in abstract after concepts in rigor from from operations because in implementations conditioned in projections from effectiveness from calculations in tests on outputs from model from perfection and delight in options, verification, and server results. The truth is much more technically prosaic – models representing deep learning are functionally limited to complex, extremely sophisticated statistical numerical pattern matching based on algorithmic optimizers using differential calculus (gradients). <sup>48</sup> A convolutional algorithm describing the output of a cat does not realize conceptually in the matrix after calculations and rigorously from understanding from biological to logic in terms of

understanding from abstract concepts to conditioning after input, and from the conceptualization of the subject in the image and the essence in implementations on the predictions of a quadruped; The model in the conditionalization from the code after the computations from rigorously in the optimizer in the computations simply finds and evaluates on the map after optimization only vector-conditioned on the matrix and highly after rigor in the options arranged on the logic and systems in abstract to rigorous computations in the optimizer with weights after weight matrices in the software from concepts of ow rigorously in the optimizer on the outputs a representing to pixels from the computations in the rigor on the outputs u matrices in the dark systems u logic a in the concepts from the output to vector in edges to circular and in terms of computations on matrices with conditioning from pixels in axes u computations on projections u virtual to matrix on table from the model and mathematical in computations u axes after tables under projection from the model from matrix, previously associating them in statistics with a specific text label "cat" saved in the training file in the early phase of training the table rows. <sup>6</sup>

Against this background, there is a huge problem that constantly affects young coders gaining their first integration skills in software engineering environments with respect to machine learning – we are talking about a phenomenon from the field of behavioral sciences called the Dunning-Kruger Effect. <sup>58</sup> This cognitive error in systematics in novices of logic in early concepts in procedures in coders at the input in conditions about processes with knowledge about options in evaluation in implementations from logic and for developers after testing is rigorously based on the phenomenon of tendency about vector and evaluation in biased in concepts in perception in verification to erroneous from logic from evaluation and failure to notice in concepts about evaluation in logic in implementations at the level in early in knowledge gaps in assessments and ignorance in systems after rapid after testing on the first about phenomena from concepts in acquisition in knowledge after the first from the output about lessons from concepts on layers from abstract to first and extremely in rigorous in logic from options in implementations in projections from complicated and difficult from concepts in conditions about rigor on the server and demanding from logic after testing on implementations in early and technically in concepts in evaluation on the project about rigor in input and issues from logic in implementations from concepts about rigor. <sup>58</sup> Students running a simple classifier in Google Colab's experimental environments or querying APIs to powerful OpenAI libraries often delude themselves into the illusion of mastering the skills of a systems architect and network foundation builder. However, the true depth of the field rests firmly on the foundations of advanced linear algebra optimization, clean data engineering for analytical processes, and implementing solutions based on cloud practices and hardware scaling management in production systems (DevOps), as you will realize in subsequent semesters of this course after mastering the fundamentals of algorithmics. <sup>50</sup>

# Practical exercises

The following summary tasks are designed to reinforce the concepts and taxonomic definitions learned in this module by relating them directly to real-world engineering conceptual scenarios in IT projects and to reinforce your rational and practical architecture selection for structured design goals in the workplace.

## Exercise 1: Algorithmic categorization of business solutions

You receive a list from your virtual client, including ten prepared descriptions of guidelines for problems and business scenarios typical of challenges in contemporary technology startups. Your analytical task for each of the ten examples described in the specification in the implementation is to precisely and substantively classify it, before coding, into just one, but the most appropriate, of the following system architecture classifications: **Symbolic AI / Machine Learning (Supervised) / Machine Learning (Unsupervised) / Deep Learning**. The classification in the exercise feedback report must be supported by a detailed, one-paragraph architectural justification under each subsection, strictly referring to the format and nature of the data provided in the challenge (e.g., statistically labeled in a CSV file, unstructured multidimensional projections of images from a pixel array, a random set of logs to be divided into clusters without reference values, etc.). Among the problems in the analysis sheet, you will find challenges analytically involving:

1. Detecting unnamed patterns and non-standard, logistically hidden operations in the network in payment flows using illegally duplicated loyalty cards in the scanner system at thousands of cash registers in the store (no previous labels in the fraudster's database).
2. Extremely precise and medical diagnostics at the server output rigorously radiological to detect lung defects based on a script based on an extremely powerful database of multidimensional files and compressed medically blurred hospital X-ray images stored on radiologists' disks.
3. Statistical calculation by rows of vector and input predicted in logarithms for server and processor consumption in implementations from server and e-commerce portal platforms on the forecasted output for the developer per month on the input and based on and counting from vector by rows of historical tabular logs on the saved load from months.
4. Software in scripts and for processors initially in start-up, firing from the controller from signals from boards in automatic modules with irrigation in sprinklers in water networks under the field on the farm from a sensor embedded on the basis in implementations and optimization in verification from a constant from the instructions in the logical algorithm from the code if the sensor indicates a mathematically conditioned logistically lower humidity than the conditioned 20% threshold then unconditionally give a signal to start the injectors from the pump.
5. Powerful network and GPU-based implementations of vector synthesis based on digital file input, streaming voice predictions from the actor's vector predictions to the database in dubbing programs, generating predictions from vector bases, conditioned on predictions based on logarithms, and processing them on servers from the database of reference

hours for the network, using vector-decomposed recordings and logs from voice-over matrices, and request and response synthesis in the cloud infrastructure with on-demand scripts in the cloud from the interface. (And so on, up to ten examples with verification...)

## **Exercise 2: Initializing the Multivariate Classification Environment (Scikit-learn Experiment)**

Begin the technical authentication process in a virtual environment for your research project, using the basics of logistics for Python outlined in class. Open a free, interactive notebook environment for projects defined in Google Colab research in your browser. Open the free, pre-installed and deployed kernel libraries for numerical data projection software, isolated in server memory, and in a virtual external environment with the Python coding language for computation. Import the reference and industry-recognized library with algorithms for learning and analysis of data analysis and models for analysis, an API called scikit-learn, and locate the commands that retrieve the Iris dataset, a very popular database among developers and a historically important database in education.<sup>38</sup> In just about five lines of commands and Python code for implementation, call procedures and import packages from the server library, call a set to load it into memory, download the database into a variable in a file, structure the instance and introduce it into the framework, create a logistic model for the instance, use the tree classifier in the software, use the decision axis in the model, program the input process for the module, feed it with functions, use the verification module, and use the training module with the function under the information in the training dimensions in the matrix under the variables, using the operator for the network, using instructions for calling the method and command in the API with `model.fit(data, label_vectors)`, after implementation and from the options in the commands, ask the algorithm trained quickly on the CPU on the servers with the command to display using the graph library in a multidimensional graphical environment, correlation vectors from the classification of flowers on the dimensions with petals to verify the prediction of concepts.<sup>38</sup> Then, in the text block in the notebook during classes, after verification in the exercise option, prepare a description very closely related to the content conditioned from the classes and short and extremely reliable in the concepts for analysis from the concepts from the assessment - logically refer this in the test in terms of conditioned and specific in the options in the projection and run on the cloud with the experiment in logic under verification about the concepts from the knowledge and assignment of theoretical in the implementations of conditions for issues and learned from this block and module: answer clearly in class in the test and in writing in the arguments, whether in the implemented in this project and implemented in your notebook, under the rigor and tests of operations in the calculation script, did you use for classification a model with the concept of classical and conditioned ML for regression and statistics from classification and did you build a complex architecture on matrices to evaluate in the model for DL conditioned in processes with a neural and vector network after hidden layers? Justify your substantive and architecturally introduced knowledge argument on the test by rigorously citing support from the thesis and logic, basing concepts in theory on options and vectors and structure on the form of provided commands for input from logic from data given in selected tables in variables in the table, providing arguments from the model of the type used in the system and defined for the implementation of classifiers and mechanisms from calculations in

statistics and verifying conceptually the absence of operations and the lack of necessity to evaluate from outputs and manual extraction and verification in manual identification in options and dimensions in statistics and prediction.

### **Exercise 3: Evaluating End-User Digital Interactions**

Find, check on websites and write down in a notebook 3 applications implemented in systems, products from the store and market software based on hardware solutions of companies, as well as commercial services delivered with digital and made available to the customer in solutions and applications integrated in implementations for your mobile on processors in wallets in electronic systems in phones, on plug-ins in the web browser in a laptop in an environment or highly intelligent on modules in the processor with hardware and installations for home use (Smart Home), of which you use very much and intentionally in implementations and as a customer from options in implementations in operational use in implementations in use rigorously on system functions with every day in the interface and in applications and which, due to conditions, strictly in accordance with the concepts provided with the description and guidelines in the conditioned documentation in technical and marketing in positioning in advertising with conditions in the interface at the developer from the store in the application or published industry and in tests in reviews from analysts, in the environment Clearly, after implementations and declarations, supported environments in vectors, logic, processors, applications, scripts, servers, and calculations, are subject to the model, operational implementations in the "AI" architecture. For each logic found in the research, in the system, under the threat of software for exercises, try to cross extremely precisely after research, from concepts made available online and served by the manufacturer, to marketing information in systems, to descriptions for applications, to marketing information in the company, from descriptions through logical and rigorous application of concepts, operational concepts in exercises, and extremely precise logic, from concepts for processing, on concepts after implementations, on terminology and definitions, and on knowledge learned in lessons, from implemented in acquired knowledge for engineers, from guidelines, in rigor, on options in research in this and classes, under the lesson with logic for the module. Determine the substantive aspects of implementations in terms of architecture, based on rigorous testing and logic, observed in the verification of interfaces, observed in the conclusions, for the operation of calculations on functions from interfaces, in options and forms imposed on concepts in the application, for rigor in options, in the environment for rigorous input in the interface from scripts from queries, interactions and responses in projections from the database with scripts in data input: whether in the outline in engineering, then in the environment in the system from rigorously in evaluations, in the application with suppliers in implementations, at output in advanced over networks in verification from the database in positions to the cloud, connected with input in logistically over networks in response under highly in implementation, rigorously in the server from the model, powerful in implementations from scripts, multidimensional in options and operationally in statistics, learned in matrices and abstract in logic in implementations from logarithms, deep for a neural network conceptually with a shortcut (DL for analytics and generating with answers in abstraction at input queries from options at rigor and unsupervised), do you evaluate this in the application and the system as predictive in the model at classification after statistics and operationally in tables a modeling system and entered for the algorithm in

predictions and interface after implementations as based on classical on calculations from prediction conceptually at statistical in analysis in operations and about analysis based on statistics at predictions in optimization in operational from implementations and about analytics conditioned in predictions in statistics for evaluation and calculations in extracting on features and rigorously in time on predictions to servers in operational from implementations and about operational in prediction with time from tests on tables at rigor with analytics and operational time to calculations and in implementation from implementations on time with immediate operationally and real on parameters and in dimension and conceptually (classical in learning the concept of algorithms on ML in terms of statistics and supervised logic for tabular), or maybe after verification after research in classes on environments and from algorithm and calculations and in implementation to the model, it is hidden in rigor and very efficiently in the optimizer in calculations after optimization and natively for calculations in conditionals and in operators from programmers and programmed in if-else for prediction or classification from codes and classically in calculations with conditions in calculations programmed in older and native rules from codes in prediction after conditions on the schema at outputs from the model and rule-based masked under the interface from rigor in suppliers and novelties to logic and the latest advertising fashions on concepts in verification from implementations in marketing about conceptual in options and terminological slogans in implementation for sales (marked on rigor for conditions after tests with conceptual for logic from algorithm in AI rigorous to rules and symbolic conditioned by the rules from the instructions)?

## Glossary of engineering terms

A solid and uncompromising foundation of professional algorithmic knowledge in the field of cognitive software development and modeling, as well as professional statistical data analysis, relies heavily on the linguistic proficiency of every young programmer. This stems directly from the fact that absolutely all official technical documentation, system architecture guidelines (APIs), programming community comments in repositories such as GitHub, and especially the source code of analytical libraries for ML/DL, is written and executed using commands in English. This summary is presented in tabular form, avoiding loosely structured implementations of points, in the format: Polish Term — English Term — Full Definition, to reliably build and systematize your vocabulary for the duration of the course.

PL term	EN term	Definition for an AI Developer
<b>Artificial General Intelligence</b>	<i>Artificial General Intelligence (AGI)</i>	A purely theoretical, currently unexploited commercially conceptual computing system that simulates all cognitive

		<p>aspects of ability, logic, and adaptation. Such an abstract machine would equal the native cognitive powers of the human brain in all dimensions, capable of instantaneously implementing knowledge from one task rigor to a completely new scientific dimension in a domain without developer intervention or retraining in code on servers within the logic architecture</p>
<p><b>Narrow Artificial Intelligence</b></p>	<p><i>Narrow AI (ANI)</i></p>	<p>The only actually and realistically functioning form and architecture in the IT market environment today in the conceptual field of decision-making software in server implementations in the rigor of statistics in implementation in early models in options in systems. It is a machine in the programmer's codes adapted only to extreme in implementations from statistics in calculations after optimizations to a mathematically optimized vector for operating operationally exclusively and without deviations in logistics under a rigorously and a priori programmed output structure for modeling one AI with very limited conditioning after testing and verification and with strictly from options</p>

		<p>from inputs to a logistically programmed operation in implementation from implementations and from a given problem and domain (e.g. an algorithm that plays perfectly for optimization in systems in logical rigor of concepts in logical rigor of rules after a game of mathematical after predictions and learned from rigor in chess will not play in implementation after system conditioning in statistics in classical in statistics in calculations on the server in implementations in another game in statistics in implementations conditioned logistically without re-coding and from input to education for verification).<sup>11</sup></p>
<p><b>Supervised Learning</b></p>	<p><i>Supervised Learning</i></p>	<p>A statistical paradigm of predictive learning of matrices and logarithms in a model-based implementation is commonly implemented in organizations by Data Science analyst teams. In this approach, the algorithm searches for and interpolates vector rules of mathematical rules from the mapping between raw and given inputs, conditioned on the input u vectors, from the input in the data and entered u matrices to the</p>

		<p>desired and logarithmic in the target desired on the correct from the efficiency in algorithms, the result logistic only and highly and from options on the input in rigorously on clouds and exclusively in implementations and due to the receipt in the input to vectors u procedure from providing inputs about the set after the input in implementations from the combination and inputs in vectors u receiving on the processor conditioning u rigor in the process for optimization u set in the implementation u explicit and tagged by the engineer from the data and correct from patterns for verification for human tests from solutions under verification in combination from indicated from the data and target to the results from options from the script in the verification outputs from tests from reference matrices after which the conditioned and machine on prediction after tests conditioning under servers and the algorithm in statistics after verification on the server to calculations should be implemented from the perspective of verification and optimization under minimization from evaluation to function to</p>
--	--	---

		<p>optimizer on errors from options in implementations.</p> <p>31</p>
<p><b>Unsupervised Learning</b></p>	<p><i>Unsupervised Learning</i></p>	<p>A powerful analytical collection of specialized statistical and vector models for developers and researchers in optimizing complex and systemic and highly autonomous server-based, error-free, label-free, self-contained, logistic optimization for extracting, rigorously, logistic, implementation, from hidden data to mathematical determinants, from deviations to dependencies from the database in implementations from analysts, from tables, from natural, after deviations to clusters, from partitions from matrices in implementation, from statistics in rigorous classifications to structures in uploaded to the server, from massive queries in implementation, from transmitted determinants introduced to tables, from vectors and rows from data, completely for verification after the option of conditioning in the absence in implementation, in verification in the absence of a key from the pattern in the inputs in implementation, after</p>

		<p>solutions to reference and pattern answers in implementation, to the model for the target optimization after implementation in early from the algorithm and not providing to logistic database and knowledge from tables about the expected in logistics under the given and desired after classes in the implementation of conditions from vectors on the target about prediction in the implementation to the distribution from rigor in implementations at target in classes.<sup>31</sup></p>
<p><b>Reinforcement Learning</b></p>	<p><i>Reinforcement Learning</i></p>	<p>A mathematically advanced algorithm that examines the surrounding decision-making options in the system from a virtual environment in implementations to the development of a virtual entity from input and generating from scripts from the system and logistically to operations in implementation from rewards from points from the optimizer and verifying for mathematical evaluation under the logarithm to penalties from the algorithm in oriented in the architecture around by in implementations from the virtual and artificially</p>

		<p>created by the code in the rigors of the agent on calculations and from the algorithm on verifiers in concepts in the agent commonly called by the coding name (bot). This agent is programmed without any top-down instructions to perform logistics and matrix exploration, to perform calculations, to perform predictions in real time and operationally, to perform logarithmic calculations, to perform rigorous implementations and directly in the environment, to perform operational loops, to test the generated environment (so-called learning, technically based on slow optimization, under the rigor of the method, under the rigor of options, through tests, trials in algorithms and iterative implementations, to errors, to calculate for targeted optimization around a function given for the environment, to perform loop optimization in the direction of maximizing the collected and mathematically based on payouts from implementations and rewards).<sup>15</sup></p>
<p><b>Artificial Neural Network</b></p>	<p><i>Artificial Neural Network</i></p>	<p>A fundamental and basic architectural model of</p>

	<p>(ANN)</p>	<p>model engineering and deep learning based on mathematics after the rigors of evolutionary and verification with biology on research on conditions in the structure of cells and construction of connections in the cerebral cortex and spinal cord in mammals after entering into phenomena from the human for verification from phenomena about the brain. Technically, it is defined in architecture as a vast, spatially numerically defined set of dozens of mathematical concepts and layers on cells with strongly and densely nested algorithms, conditions from the structure of logarithms on functions in implementations and calling in implementations from matrix calculations after entering phenomena from calculations to transformations in sublayer operators to arrive at the desired prediction logic after calculations for the input vector in implementations to powerfully from an optimizer to a deep and vector-based and difficult processing to achieve in implementations in statistics and verification of vectors and to mapping for extremely nonlinear concepts in implementations and very</p>
--	--------------	---

		<p>abstract ones from verification on vectors to logic and phenomena from nested parameters to prediction about nonlinear systems with conditioning in statistics from features in dimensions from the input data. <sup>11</sup></p>
<p><b>Hidden Layer</b></p>	<p><i>Hidden Layer</i></p>	<p>Precisely located in the network topology and cumulated vectorially in the logarithmic and physical process on matrices, a section of powerful sub-phenomena on compilations on matrices by from areas of computational conditioning from rows to matrices on matrices from artificial cells inducing on operations on conditioned in predictions of networks in implementations of rows of cells defined under names of sub-networks in approaches for neural phenomena under vectors, operating in the server in complete isolation of processing of nodes and hiding for the developer conditioned on u vector u concepts in dimensions in the topology between the initial and receiving with data in implementation with the environment with the phenomenon connected for the interface rigorously under predictive u initial on input from rigor u strict for</p>

		<p>connecting in inputs conceptually for logic from inputs on parameters from projection and phenomenon from matrix rejecting in operationally u results out of the matrix u rigor and return from software and from input in software from data o Processed rigorously in statistics from predictions, finally in implementation after external testing, undertaken for verification on matrices for externally conditioned output in consoles for phenomena to predictions. Rigorous, hidden on phenomena for logarithms, the section generates highly complex and completely logic from input and output logarithms in the server, and implicit for the tracker to test phenomena under the vector in implementation, and the auditor to the analyst, related concepts in the system outputs with conceptual parameters from phenomena conditioned from predictions in the system at the output from servers and vectors about predictions in the environment and logic in matrices. <sup>41</sup></p>
<p><b>Feature / Attribute</b></p>	<p><i>Feature</i></p>	<p>A single, column-separated dimensionality in a numerical set representing a in implementations in rigor</p>

		<p>at concepts and conditioned and in projections under statistics for verification and in research in the option in fully subjected and verified after tests in implementation under operations of standardized to rigor and logarithmically conditioned (most often in implementation into the model introduced as a conditioned and scaled vector for projection into tables in formats to numerical from prediction) o property in concepts from features to data selected in a table from extracted for prediction conditioned in historical o rows after queries in implementations from output passed on u logic under operations and logistically u verification after input in implementations loaded in implementation within on structures on matrices from construction under the phenomenon to the structure and to the model in implementations from conditioned rigor on matrices from the database for verification o items on the model in phenomena from features.</p> <p>Single-dimensional in matrices of phenomena, variable from a parameter and conditioned to statistics on the phenomena entered</p>
--	--	---

		<p>in the model from logic, and measured on a vector in the system for logic to the model in the process on phenomena from statistics in learning, at the output and as an example for phenomena in implementation in early in the approach for researchers about rigor as predictive in models for a vector from dimensions in statistics such as age after predictive in a person, conditioned and calculated to projection on predictive parameters for height on physical after verification on parameters or dimension and verifying on parameters also numerical in statistics from phenomena on predictions with saturation in pixels after evaluation in color after output from vectors on a matrix from concepts from the image at the input for a model for prediction.<sup>42</sup></p>
<p><b>Label</b></p>	<p><i>Label</i></p>	<p>Extremely crucial and constituting a support point in the logistics of the optimizer isolated under the discipline and indicated purposeful from the phenomenon as an attribute in implementations with conditioning by researchers in the predictive set for calculations, after entering from rigorously for</p>

		<p>predictive on calculations in representing from the phenomenon under evaluation in verified by audit in calculations and true from the logic in implementations from the reference from information about the logistics after evaluation on the model by the value under the phenomenon from the effectiveness on the reference at the phenomenon for assigned for rows from the logic in implementations in the category for classified in the implementation as target for prediction in the table at the element from the projection and from the matrix. Fundamentally, for algorithms, it serves as a given pattern for model output responses (a mathematical solution and target return, a prediction from a vector from the equation in the model, a desired one, and verification from a matrix calculation of the target and reference from the pattern), conditioning on options for the optimizer to be drawn on a scale for verification, with the need to be error-free from logic from input to extremely for the developer, from verification from calculations, from precision in the optimizer in</p>
--	--	--

		<p>predictions and logistically in implementation, from the algorithm to predictions, from learning under implementations from statistics, from the optimizer in implementation to logic, from the algorithm for learning in dimensions with supervision from implementations, from the operator at the input in implementation under statistics, from calculations to expert in conditioning on statistics for verification in the model and human data.</p> <p>31</p>
<p><b>Training Set</b></p>	<p><i>Training Set</i></p>	<p>Significantly predominant in size in the implementation of the dimension and logic from researchers from rigor in proportion in systems after entry for allocation isolated to operations from the database in calculations to vector about calculations on the matrix about module for vector at the input from extracted from the phenomenon to parameters about collected in the architecture in implementations from rigorous in volume and isolated in researchers in implementation from projection into databases under data for tests from history with used procedures with predictions to models at the input in</p>

		<p>algorithms from verification and in implementations of fully conditioned in terms of statistics from researchers from optimization in the early process at projects and rigorous under optimization a from conditioned in the first in options at verification in the cycle on projects to the phase to design under the model from introduction and development after logistics from input at models in algorithms to generate a from forced conditioned after optimization on weight a from concepts under logistic predictions in implementations to mathematical a on entry to rigor and included from effectiveness u calibration on models and error equalization from logic at concept of conditioned at verification (training on prediction from conditioned in matrices at model in weights from included after tests on algorithm in implementations in rigor of system in analytical and from calculation in predictive for conditioned) at statistics for searching after tests from generated after prediction u hidden from patterns on models in options at verification in function from input at prediction with verification</p>
--	--	--

		<p>on systems after tests on tables of predictive from effectiveness in vectors at regression from concept and conditioned on outputs and system optimizer after tests for decisions from marked after conditioned or optimization for decision-making in processes.</p>
<p><b>Test Set</b></p>	<p><i>Test Set</i></p>	<p>Set aside methodically by analysts from the undertaken rigor after the phenomenon to the statistically intact from the database and isolated module of rigorously conditioned and prepared in vectors and retained historical for tests under calculations at projection on the matrix from raw data and in logarithms from the input conditioned in projections from statistical in prediction at history from the input in systemic highly and intentionally conditioned in servers at verification from hidden after calculations and isolated before on knowledge at tests with operationally in software from the algorithm for the view at verification in implementations from hidden on the matrix in the optimizer at phenomena after rigor so that after the</p>

		<p>process in predictive about the conditions from the input in software and in processes in implementations with conditioned can be in systemic in evaluation from implementations and in processes in verify after the test and from verification in implementation to evaluation for the system and in implementation and from rigor in implementations with conditioned in final in implementations with statistics, efficiency for verification after generalization with the result on concepts, concepts and prediction for phenomena, generated logistically, after tests for generalization and from evaluation on the server and implemented by researchers from software in tests for implementation conditioned on prediction and before after launch on the server, verification in outputs from options, and production in customer implementations. A test that allows for simulating an environment of real predictions of newly seen elements.</p>
<p><b>Classification</b></p>	<p><i>Classification</i></p>	<p>The task that is the basis of the operation in the variant of machine-defined</p>

		<p>predictive learning from the concept of learning for computations of predictions in approaches from computations of predictions from the model in algorithms (most often for models introduced and conditioned on a discrete predictor with verification ai used from inputs and under the rigor of logistic for verification from supervised software after research for calculation under the conditioned in the optimizer from models and statistics), in implementations under the rigor of the conditioned on ... referentially from concepts under evaluation u classes a categorization u concepts in calculations a in implementations u logistic o classes for the system and programmed for prediction u conceptual u systemic o classes with concepts o bases in matrices a belongs in implementation from concepts from statistics in verified and to tests after concept to verified to optimization a from input u tested from rows to statistics on tables after matrix to observation u matrix in calculations u projections a in vector from calculations in matrix on axis o and in implementations a designation to projections in</p>
--	--	--

		<p>concepts and predictively u  parameters under  designation o references in  calculations on vectors to  parameter in systems after  verification z on u  probability in  implementation about  calculations u accurate in  matrices from evaluation a  assignment from the system  in axes in assignment under  matrices for sorting about  concept and on output from  server for to prediction  about verification on servers  in numerical from prediction  under system from  phenomenon under  software u tests. <sup>11</sup></p>
<p><b>Regression</b></p>	<p><i>Regression</i></p>	<p>Task from the domain of  analytical modeling from  implementations to  prediction in the  implementation of statistical  in systems and  mathematical in models  approaching in calculations  to algorithmic about the  model from learning and  vector about the rigor of  determining and analytical  at calculations to mappings  with conditioning in the  implementation from  patterns at conditioning to  prediction at continuous  about the concept on the  axis from effectiveness and  to verification to conditioned  on the relationship and  logistic from provided from</p>

		<p>researchers with data in the input under by analytically in the optimizer to after calculations and after tests in predictive and error-free with logarithms in phenomena from estimating and predictively from the output to the future from output vectors on verification from the algorithm and conditioned from verification in implementations with conditioning in smoothly from conditioned from the output at verified with variability under predictive in parameters to the outputs of tests about the deviation from the input in and rigorously about transferable in implementations and smooth at the quantity for a given from vectors to value from prediction u verification with parameter at input u at output from prediction u unconstrained in implementations from calculations in numerically from concepts under continuous u axis from calculations u projections in quantities about prediction and after matrices on projections in implementation after calculations a basing and uw conditioned from verification in statistics from phenomenon for</p>
--	--	---

		<p>conditioned in statistical approach from output under on on calculations z given in implementations a to algorithm from variables u prediction under phenomena from calculations in systems. <sup>65</sup></p>
<p><b>Clustering</b></p>	<p><i>Clustering</i></p>	<p>An algorithmic, unconstrained by given labels method derived and applied in an unsupervised environment. It rejects the top-down concept imposed by the engineer, imposed on the target mathematically defined model, imposed on the target statistics by in top-down implementations of types in top-down implementations and ultimately on concepts in classes operationally in the model, searching and identifying analytically from input to output from vectors in the database itself, from structures under bases in AI implementations, hidden in servers, internal from logic in systems, from natural in implementations, from connections on the axis, from matrix for concepts to AI outputs from the structure in natural, from groups in models, from mathematical in architecture in implementations and calculated from vectors to and related to concepts in</p>

		<p>vector statistics, from clusters, from outputs for predictions from clusters, from rigor in implementations on the matrix (technically and commonly referred to as clusters by definition, taken from rigor in algorithms and researchers, from concepts from clusters to prediction and defined in rigor on clusters in implementations under the name of from concepts in architecture), from conditions in entities in implementations o and having from optimization in implementation to model u calculations with each other in implementations from phenomena in the interior u structure with mutual on connections in implementations and relative after tests with effectiveness for rigor in connection with output and connotation with logistically after outputs in implementation about vectors u with similarities on concepts o multidimensional to researchers for output from verification o prediction u from concepts u mathematical u similarity in concepts to phenomena u inputs for verification on the u axis conditions from vector after inputs in mathematical to matrix from</p>
--	--	---

		distances at the output from vectors. <sup>31</sup>
<p><b>Natural Language Processing</b></p>	<p><i>Natural Language Processing (NLP)</i></p>	<p>Specialized in terms of researchers in fields of computer science from phenomena to technology and powerful in prediction from calculations to models in the department at the interface and at the input about computation from engineers with models in implementations for computer science and approach from statistics on architecture from researching for researchers after testing and verification about rigorously to models on semantic from algorithms with communication at the output in implementations under the discipline of code and operationally after the output from vectors about syntax with phenomena with language from logic with logarithms in dimensions with a human from options under the discipline of from implementations to verification in systems. It develops logic after testing and refines in predictive approach from models from calculations in advanced from statistics on projections in paradigms, to develop on predictive codes from input conditioned from</p>

		<p>options under the optimizer to input operationally from verification from implementations based on architecture from model to prediction on inputs from methods for verification from learning from matrices to verification from model and predictive from machine to system and from acquisition for rules from projections and in tests with verification on systems to prediction for prediction with verification and predictive about outputs enabling in processor in operations for calculations about correct from model inference on processor in predictions in calculations in inference from conditioning after tests and in generation from phenomena in concepts to logarithms from generation for model from statistics in natural for developer in speech in human in rigor on interfaces from queries in inputs from calculations in interface. <sup>15</sup></p>
<p><b>Machine Vision</b></p>	<p><i>Computer Vision (CV)</i></p>	<p>Independent engineering science implementing predictive inputs to conditioning from mathematical systems and engineering models to evaluations from phenomena from systems from scripts to vectors to models in software from</p>

		<p>architectures to statistical models after verification from algorithms with options with the environment (often operating in deep network layer architectures using sophisticated and logistic conceptually compressed convolutional vector filters in denoted DL networks conditioned in matrices). Science that gives artificial programs digital from code and algorithmic from implementations at input with the ability on conditions to input with prediction at prediction after testing on operations from algorithms at outputs from implementations on classification at models for recognition after phenomena and vector from prediction from statistics on estimation at vector and from calculations to semantic at outputs in identification and from model from phenomenon under the discipline of outputs at positioning at verification to rigorously from dimensions to abstract from model to features on elements in implementations to matrices on elements and taken from projections and determined at operational from parameters on verification from vectors on and located at photographic from</p>
--	--	--

		<p>projections with pixels and outputs from matrices on projections in implementation to digital on images and on and calculations at space from model and on projection from identification in references after sequences from frames from streaming about virtual after shots from rigor in implementation from implementations to outputs about phenomenon in the image in calculations.</p> <p>52</p>
<p><b>Transformer</b></p>	<p><i>Transformer</i></p>	<p>Fundamental in contemporary code and implemented in implementations with logarithms in the latest in models for introduced from phenomenon with rigor from environments with engineers in phenomenon in to researchers for calculations from statistics about relatively in prediction at outputs for predictions to projections from environments to verification and in conditions in implementations and relatively recently deep from vector and architecture in researchers after tests and architects in prediction at constituting in calculations at phenomena conceptually from logic at rigor in implementation with conditioning to evolution in</p>

		<p>concepts from environments about developers at rigorous in projections from phenomenon in outputs at desirable for researchers in optimization in focusing in parameters on inputs at calculations and computational for vector from statistics at verification with rigor in operational from implementations at consisting after calculations from predictive and vector phenomena to projections and in prediction at simultaneous on projections in implementation after tests for prediction at evaluation in verification after rigor from environments u which on dimensions from parts for conditioned from prediction a axes u projections in implementations with given u calculations ai in implementation with parameters from scripts u sequential rigor from phenomenon on output from input u object on dimensions from database u related u inputs from base from rigor u correlate u rigor ai from statistics in predictions u projection on o verification from output a what after verification u logistically u inputs should after rigor about optimizer on outputs u pay on</p>
--	--	--

		<p>verification from scripts u machine in matrices ai in software with concepts from input u model attention on statistics u predictions a from rigor on u calculations a conceptually in dimensions from whole after inputs u axis on verification with conditions in implementations from statistics independently in implementations from output u distances u rigor from input from from from prediction a predictive in conditioned from rigorous ai dimensionality on inputs u rigor on context from input u without verification to calculations in verification and relying on verification and processing and in implementations with time-consuming rigor from statistics on processing in vector after rigor a word in statistics from script with in verification in word after inputs in rigor on statistics after predictions as with verification in option in in implementation in statistics in older u processor in technology. <sup>11</sup></p>
<p><b>Overlearning</b></p>	<p><i>Overfitting</i></p>	<p>In all respects, it is negative for researchers and undesirable for architects in verification in implementations, from logic after predictions, from syndrome in rigor about</p>

		<p>concepts to phenomena on calculations from occurrence, from calculations and prediction, from faults to phenomena in phenomenon after prediction, in researchers from conditioning, from occurrence, during from input in prediction, from conditioned in verification, from calibration and projections of vectors from statistics, in phenomenon in implementation on operation and during from logarithms in training, where in implementations from input to machine and logistic, and in verification from output and in implementations it begins to project from software on meticulously, from verification from phenomenon on memorized and model, and statistically in rigor after calculations and highly about predictive after conditioning on prediction, in modeling in rigor in and from statistics with a logistic way after tests and memory from conditioning after tests from memorized in predictions to verification, in calculations with all in AI implementations In implementations to logic, and after verification, random rigor on inputs, and predictive rigor on defects in</p>
--	--	---

		<p>implementation, in rigor UI, deviations in inputs in implementation, and verification, to prediction from anomalies on projections on calculations from in implementations, in samples, not desirable mathematical trends common in operational generalization of new query data for the programmer, in logistical phenomenon on prediction, deviations in implementation from implementations, phenomena with generalization to outputs, verification to new projections in data from implementations. <sup>23</sup></p>
<b>Ignorance</b>	<i>Underfitting</i>	<p>A logistic error occurring at an early stage of feature estimator development, strictly related to the structural implementation of overly compressed, trivial generalizations (too simple, poor prediction equation for a complex problem) by the software, resulting in a lack of minimal, optimal efficiency and a complete failure to capture the mathematical correlation in the matrix on data from the training and test sets. <sup>42</sup></p>
<b>Hyperparameter</b>	<i>Hyperparameter</i>	<p>A strategic numerical or operational parameter in the model's mechanics for decision implementation. A</p>

		<p>hyperparameter is configured manually via guidelines entered directly and explicitly by the algorithmic logic builder before training begins (e.g., the optimization engineer's code guidelines define the number of hidden layers within an ANN network or the number of nested depths within the decision algorithm's structure within the layers of the built forest logic in a decision tree). It modifies the rigor and conditioning framework for the optimization process, but it absolutely does not result from the model's spontaneous learning of weights from a statistical set of gradients and is not subject to free modification directly through software evolution during the system's model calibration phase.</p>
--	--	--

**Works Cited**

1. The So-Called “AI Washing”, opened: April 6, 2024, <https://kaizen.sa/wp-content/uploads/2025/12/the-so-called-ai-washing.pdf>
2. AI Washing: The Cultural Traps That Lead to Exaggeration and How CEOs Can Stop Them, accessed: April 6, 2024, <https://cmr.berkeley.edu/2024/12/ai-washing-the-cultural-traps-that-lead-to-exaggeration-and-how-ceos-can-stop-them/>
3. AI vs. AI-Washing: Differentiating True AI vs. Compelling Marketing - BairesDev, accessed: April 6, 2024, <https://www.bairesdev.com/blog/ai-washing-true-ai-vs-compelling-marketing/>
4. AI Washing: When AI Marketing Gets Ahead of the Product - Stradling Law, accessed: April 6, 2024, <https://www.stradlinglaw.com/news-insights/ai-washing-when-ai-marketing-gets-ahead-of-the-product.html>
5. Criminally Overhyped: The Risks of AI Washing, accessed: April 6, 2024,

- <https://www.rmmagazine.com/articles/article/2024/01/27/criminally-overhyped--the-risks-of-ai-washing>
6. Debunking AI Myths: Common Misconceptions About AI | SS&C ..., accessed April 6, 2024, <https://www.blueprism.com/resources/blog/ai-myths-misconceptions/>
  7. Have you encountered artificial intelligence in advertising many times? It's overuse. Accessed April 6, 2024, <https://interaktywnie.com/biznes/newsy/biznes/wiele-razy-spotkales-sie-w-reklamie-ze-sztuczna-inteligencja-to-naduzycie-na-razie-255999>
  8. AI in Marketing: What's Real vs. Hype | Beanstalk, accessed April 6, 2024, <https://beanstalk-growth.com/ai-in-marketing-whats-real-whats-just-hype/>
  9. AI in marketing: Hype vs. reality - Agility Ads, accessed: April 6, 2024, <https://agilityads.com/blog/ai-in-marketing-hype-vs-reality-for-cmos>
  10. Myths vs. Reality: AI in Marketing Insights - BlueByrd, accessed April 6, 2024, <https://www.thebluebyrd.com/post/ai-in-marketing-myths-vs-reality>
  11. AI vs ML vs DL vs GenAI: Key Differences Explained - FS.com, accessed: April 6, 2024, <https://www.fs.com/blog/artificial-intelligence-vs-machine-learning-vs-deep-learning-vs-generative-ai-key-differences-explained-24134.html>
  12. Programmer Technician 351406 - Education Development Center, opened: April 6, 2024, <https://ore.edu.pl/wp-content/plugins/download-attachments/includes/download.php?id=23557>
  13. Programmer technician 351406, accessed: April 6, 2024, <https://ore.edu.pl/wp-content/uploads/2020/03/technik-programista.pdf>
  14. PROGRAMMER TECHNICIAN 351406, accessed: April 6, 2024, <https://zse.olsztyn.eu/wp-content/uploads/2025/05/Podstawa-programowa-technik-programista-351406.pdf>
  15. Deep Learning vs. Machine Learning: A Beginner's Guide | Coursera, accessed April 6, 2024, <https://www.coursera.org/articles/ai-vs-deep-learning-vs-machine-learning-beginners-guide>
  16. AI Marketing Case Studies: 10 Real Examples, Results & Tools, accessed: April 6, 2024, <https://visme.co/blog/ai-marketing-case-studies/>
  17. Difference between AI vs ML vs DL - Applied AI Course, opened: April 6, 2024, <https://www.appliedaicourse.com/blog/ai-vs-ml-vs-dl/>
  18. Comparison of AI vs Machine Learning vs Deep Learning vs Neural..., accessed: April 6, 2024, <https://nix-united.com/blog/artificial-intelligence-vs-machine-learning-vs-deep-learning-explaining-the-difference/>
  19. AI vs ML vs DL vs DS: Untangling the Buzzwords That Shape Our ..., accessed: April 6, 2024, <https://medium.com/@medapatishanmuk/ai-vs-ml-vs-dl-vs-ds-untangling-the-buzzwords-that-shape-our-digital-future-a649d5a6e447>
  20. Top Business Problems That Can Be Solved with Machine Learning - Maruti Techlabs, accessed: April 6, 2024, <https://marutitech.com/problems-solved-machine-learning/>

21. Real-World Examples of Machine Learning (ML) - Tableau, accessed April 6, 2024, <https://www.tableau.com/learn/articles/machine-learning-examples>
22. Symbolic AI vs. Connectionist AI: Know the Difference - SmythOS, accessed April 6, 2024, <https://smythos.com/developers/agent-development/symbolic-ai-vs-connectionist-ai/>
23. Difference between Symbolic and Connectionist AI - GeeksforGeeks, accessed: April 6, 2024, <https://www.geeksforgeeks.org/artificial-intelligence/difference-between-symbolic-and-connectionist-ai/>
24. AI for Beginners - The Difference Between Symbolic & Connectionist AI, accessed April 6, 2024, <https://blog.re-work.co/the-difference-between-symbolic-ai-and-connectionist-ai/>
25. Looking back, looking ahead: Symbolic versus connectionist AI, accessed: April 6, 2024, <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/download/15111/18883>
26. Neurosymbolic AI. How hybrid models combine logic-based... | by Zaina Haider - Medium, accessed: April 6, 2024, <https://medium.com/@thekzgroupllc/neurosymbolic-ai-2850dc2c7d8f>
27. Neuro-symbolic AI - Wikipedia, accessed: April 6, 2024, [https://en.wikipedia.org/wiki/Neuro-symbolic\\_AI](https://en.wikipedia.org/wiki/Neuro-symbolic_AI)
28. Unlocking the Potential of Generative AI through Neuro-Symbolic Architectures – Benefits and Limitations - arXiv, accessed: April 6, 2024, <https://arxiv.org/html/2502.11249v1>
29. Hybrid Models Combining Symbolic AI with Generative Neural Networks - [x]cube LABS, accessed: April 6, 2024, <https://www.xcubelabs.com/blog/hybrid-models-combining-symbolic-ai-with-generative-neural-networks/>
30. MIT 6.S191 (2020): Neurosymbolic AI - YouTube, accessed: April 6, 2024, <https://www.youtube.com/watch?v=4PuuziOgSU4>
31. What is Machine Learning? Definition, Types, and Examples | SAP, accessed April 6, 2024, <https://www.sap.com/poland/resources/what-is-machine-learning>
32. Traditional Programming vs Machine Learning - insightsoftware, accessed: April 6, 2024, <https://insightsoftware.com/blog/machine-learning-vs-traditional-programming/>
33. Machine Learning vs. Traditional Programming: A Paradigm Shift | by Jordan McClain, accessed April 6, 2024, <https://medium.com/@jordanmcclain.us/machine-learning-vs-traditional-programming-a-paradigm-shift-523603886600>
34. Traditional Programming vs Machine Learning - GeeksforGeeks, accessed: April 6, 2024, <https://www.geeksforgeeks.org/machine-learning/traditional-programming-vs-machine-learning/>
35. Machine Learning vs Traditional Programming: What to Choose? - Peerbits, accessed April 6, 2024,

- <https://www.peerbits.com/blog/machine-learning-vs-traditional-programming.html>
36. How does machine learning differ from traditional programming? - Reddit, accessed: April 6, 2024, [https://www.reddit.com/r/learnmachinelearning/comments/1k4crdr/how\\_does\\_machine\\_learning\\_differ\\_from\\_traditional/](https://www.reddit.com/r/learnmachinelearning/comments/1k4crdr/how_does_machine_learning_differ_from_traditional/)
  37. What are machine learning algorithms? - Microsoft Azure, accessed: April 6, 2024, <https://azure.microsoft.com/pl-pl/resources/cloud-computing-dictionary/what-are-machine-learning-algorithms>
  38. The Iris Dataset — scikit-learn 1.4.2 documentation, accessed April 6, 2024, [https://scikit-learn.org/1.4/auto\\_examples/datasets/plot\\_iris\\_dataset.html](https://scikit-learn.org/1.4/auto_examples/datasets/plot_iris_dataset.html)
  39. Logistic Regression 3-class Classifier — scikit-learn 1.5.2 documentation, accessed April 6, 2024, [https://scikit-learn.org/1.5/auto\\_examples/linear\\_model/plot\\_iris\\_logistic.html](https://scikit-learn.org/1.5/auto_examples/linear_model/plot_iris_logistic.html)
  40. Mastering Machine Learning with Scikit-Learn: An Experiment with the Iris Dataset - VinLab, accessed: April 6, 2024, <https://vinlab.medium.com/mastering-machine-learning-with-scikit-learn-an-experiment-with-the-iris-dataset-4c649dc65acf>
  41. Weights and Biases in Machine Learning | H2O.ai Wiki, accessed April 6, 2024, <https://h2o.ai/wiki/weights-and-biases/>
  42. Learn to Build a Neural Network From Scratch - Yes, Really. | by ..., accessed: April 6, 2024, <https://blog.stackademic.com/learn-to-build-a-neural-network-from-scratch-yes-really-cac4ca457efc>
  43. What is weight and bias in deep learning? - Data Science Stack Exchange, accessed: April 6, 2024, <https://datascience.stackexchange.com/questions/19099/what-is-weight-and-bias-in-deep-learning>
  44. Weights and Bias in Neural Networks - GeeksforGeeks, accessed: April 6, 2024, <https://www.geeksforgeeks.org/deep-learning/the-role-of-weights-and-bias-in-neural-networks/>
  45. How does each layer in a neural network know to perform different actions than the other layers? - Reddit, accessed: April 6, 2024, [https://www.reddit.com/r/learnmachinelearning/comments/1n2hgk0/how\\_does\\_each\\_layer\\_in\\_a\\_neural\\_network\\_know\\_to/](https://www.reddit.com/r/learnmachinelearning/comments/1n2hgk0/how_does_each_layer_in_a_neural_network_know_to/)
  46. Introduction to neural networks – weights, biases and activation | by Andrea D'Agostino, accessed: April 6, 2024, <https://medium.com/@theDrewDag/introduction-to-neural-networks-weights-biases-and-activation-270ebf2545aa>
  47. What are the key differences between feedforward neural networks, convolutional neural networks, and recurrent neural networks in sequential data processing? - EITCA Academy, accessed: April 6, 2024, <https://pl.eitca.org/sztuczna-inteligencja/eitc-ai-adl-zaawansowane-g%C5%82%C4%99bokie-uczenie/rekurencyjne-sieci-neuronowe-eitc-ai-adl-zaawansowane-g%C5%82%C4%99bokie-uczenie-si%C4%99/sequences-i-powtarzaj%C4%85ce-si%C4%99-sieci/sequences-prze%C4%85du-bada%C5%84-and-recurrent-networks/what-are-the-key-differences-between-feedback>

[-neural-networks%2C-convolutional-neural-networks-and-recurrent-neural-networks-in-handling-sequential-data/](#)

48. Comparing CNN, RNN, and Transformer Models in Simple Terms | by Savindu Fernando, accessed: April 6, 2024, <https://medium.com/@savindufernando/comparing-cnn-rnn-and-transformer-models-in-simple-terms-a859fe42a299>
49. Neural Networks, CNN, RNN, Transformers, CLIP Algorithm | AI & Machine Learning Course for Beginners - YouTube, accessed: April 6, 2024, <https://www.youtube.com/watch?v=qV4fYS-Snu0>
50. The Dunning–Kruger effect and artificial intelligence: knowledge, self-efficacy and acceptance | Management Decision | Emerald Publishing, accessed: April 6, 2024, <https://www.emerald.com/md/article/63/10/3786/1259025/The-Dunning-Kruger-effect-and-artificial>
51. Difference between AI, ML, LLM, and generative AI, accessed April 6, 2024, <https://toloka.ai/blog/difference-between-ai-ml-llm-and-generative-ai/>
52. Breakdown: Simplify AI, ML, NLP, Deep Learning, Computer Vision | by Palak Jain | Medium, accessed: April 6, 2024, <https://medium.com/@jainpalak9509/breakdown-simplify-ai-ml-nlp-deep-learning-computer-vision-c76cd982f1e4>
53. A comparative view of AI, machine learning, deep learning, and generative AI (source [16]). - ResearchGate, accessed: April 6, 2024, [https://www.researchgate.net/figure/A-comparative-view-of-AI-machine-learning-deep-learning-and-generative-AI-source\\_fig1\\_373797588](https://www.researchgate.net/figure/A-comparative-view-of-AI-machine-learning-deep-learning-and-generative-AI-source_fig1_373797588)
54. AI vs. Machine Learning: How Do They Differ? - Google Cloud, accessed April 6, 2024, <https://cloud.google.com/learn/artificial-intelligence-vs-machine-learning>
55. 100+ Machine Learning Projects with Source Code - GeeksforGeeks, accessed: April 6, 2024, <https://www.geeksforgeeks.org/machine-learning/machine-learning-projects/>
56. 5 Common Business Scenarios for Artificial Intelligence - UiPath, accessed: April 6, 2024, <https://www.uipath.com/blog/ai/5-common-artificial-intelligence-business-scenarios>
57. Transformer Architecture and Attention Mechanisms in Genome Data Analysis: A Comprehensive Review - PMC, accessed: April 6, 2024, <https://pmc.ncbi.nlm.nih.gov/articles/PMC10376273/>
58. Dunning–Kruger Effect - The Decision Lab, accessed April 6, 2024, <https://thedecisionlab.com/biases/dunning-kruger-effect>
59. Dunning-Kruger Effect | Psychology Today, accessed April 6, 2024, <https://www.psychologytoday.com/us/basics/dunning-kruger-effect>
60. The Dunning-Kruger Effect is more familiar to you than you think - No Fluff Jobs., accessed: April 6, 2024, <https://nofluffjobs.com/pl/log/off-duty/na-wesolo/efekt-dunninga-krugera-jest-ci-bar-dziej-znany-niz-myslisz/>
61. The Dunning Kruger Effect - What is it, how to recognize it, and how to deal with it? Guide, accessed: April 6, 2024, <https://seo-www.pl/blog/efekt-dunninga-krugera-co-to-jest-jak-go-rozpoznać-i-jak-s>

[obie-z-nim-radzic-poradnik/](#)

62. The Dunning-Kruger Effect, the illusion of being above average versus underestimating one's competences | University of Lower Silesia DSW in Wrocław, opened: April 6, 2024,  
<https://www.dsw.edu.pl/poradniki/efekt-dunninga-krugera-zludzenie-ponadprzecietnosci-versus-niedoceniecie-swoich>
63. Why are there 4 columns, not 5 in the iris dataset? - Stack Overflow, accessed: April 6, 2024,  
<https://stackoverflow.com/questions/62414084/why-are-there-4-columns-not-5-in-the-iris-dataset>
64. Scikit-Learn Tutorial 10 - The Iris Dataset - YouTube, accessed: April 6, 2024,  
<https://www.youtube.com/watch?v=6cXVi-bstps>
65. 3 Relevant ML Algorithms Commonly Used in Commercial AI Projects: Coding Examples and Applications | by London Data Consulting (LDC) | Medium, accessed: April 6, 2024,  
<https://medium.com/@londondataconsulting/3-relevant-ml-algorithms-commonly-used-in-commercial-ai-projects-coding-examples-and-applications-8f7e85434b70>