NYC

The New York City Typology of Artificial Intelligence (AI)

v1.2 (December 31, 2025)

Table of Contents

Cover Photo: Ed Reed, New York City Mayoral Photography Office

1 Overview

What is artificial intelligence (AI), and how can its use be governed to manage its risks while reaping its benefits?

There are multiple conflicting definitions for AI, and experts do not agree on what exactly constitutes AI.[1] Even under a single definition of AI, many technologies could be considered AI. Is predictive text ("autocomplete") an AI? A self-driving car? A chatbot? The lack of consistent definitions for AI complicate discussions about its use and governance. The lack of consistent vocabulary also complicates discussions around its constituent pieces, such as details of the computational model or data.

To ensure consistency in describing AI, the City, through the Office of Technology and Innovation ("OTI") has chosen a definition of AI described in *New York City AI Principles and Definitions Guidelines*, as discussed in Section 3 below. The purpose of this typology is to provide clarity on *what* AI is, in order to discuss how to govern it later. In effect, this typology provides different ways to fill in the following blank:

> This tool is a ________________ (type of) AI.

1.1 Relationship to the AI Action Plan

This document has been prepared as an extension of Action Item 1.4 of The New York City Artificial Intelligence Action Plan:

> Using the city's existing public reporting of algorithmic tools, as well as additional research, create a typology of AI projects to reflect the variety of technologies and uses that may fall under the umbrella term of "AI" for New York City. The resulting typology can be used to inform governance efforts, clarify agency

[1] *See, e.g.*, K. Schaul, *Is this AI? See if you can spot the technology in your everyday life*, The Washington Post, Apr. 2, 2024, https://web.archive.org/web/20240627222342/https://www.washingtonpost.com/technology/interactive/2024/what-is-ai/.

support needs, and enhance public engagement and understanding.

The Action Plan describes two objectives for this typology, namely:

1) to describe when certain technologies and uses of those technologies are considered AI, and
2) to create a means to organize the variety of technologies and uses.

To support these objectives, this typology also introduces precise definitions for the terms used to talk about AI. The purpose of introducing these definitions, even for seemingly commonsense and obvious terms, is to avoid any ambiguities that may arise from using English words to describe precise technical concepts.

1.2 What is a typology?

A typology defines a baseline set of characteristics that should always be considered when describing its constituent types. This typology describes the essential characteristics of an AI system, while being deliberate in omitting many other details. By specifying these characteristics, this typology also helps to understand the similarities and difference between AI systems.[2]

The AI typology presented in Section 4 below separates out the characteristics of AI in the definition of Section 3 below into individual facets. Each facet can be isolated to compare and contrast a specific dimension, and yet be put together to form a minimal, yet complete description of AI.

Classification schemes come in many different forms, and even the term "typology" is not consistently used by specialists.[3] Where necessary to

[2] The concept of similarity is formalized mathematically in the *equivalence classes* defined by the typology. Precisely defining what equality means is studied formally in a branch of mathematics called *category theory*. Some relevant references are R. F. C. Walters, *Categories and Computer Science*, Cambridge Texts in Computational Science vol. 28 (D. J. Cooke, ed.), Carslaw Publications (Cambridge, UK), 31 May 1991; and B. Milewski, *Category Theory for Programmers* (Version 98b71ac), 30 Jan. 2023, https://github.com/hmemcpy/milewski-ctfp-pdf.

[3] A modern take on the various formats of classification systems can be found in H. Hedden, *The Accidental Taxonomist* (3rd ed.), Information Today Books: Medford, NJ, Nov. 2022.

describe the AI typology and its associated concepts formally to avoid ambiguity, we use the following definitions:

- A **Typology** is "any analysis of a particular category of thing into classes based on common characteristics".[4]
- A **Type** is any class belonging to a Typology.[5]
- A **Faceted Typology** is a Typology that uses semantic categories which combine to define a Type.[6]
- A **Facet** is any semantic category in a Faceted Typology.

---

[4] "Typology", in American Psychological Association, Apr. 19, 2018, https://dictionary.apa.org/typology (*accessed* Apr. 16, 2024).

[5] The study of types is a rich subject in computer science, where it goes by the name of type theory. Some relevant references are P. Martin-Löf, *Intuitionistic type theory*, Bibliopolis: Napoli, 1984; M. Abadi & L. Cardelli, *A theory of objects*, Springer-Verlag: NY, 1996.

[6] *Adopted from* "faceted classification," D. J. Foskett, *The Construction of a Faceted Classification for a Special Subject*, Proceedings of the International Conference on Scientific Information, National Academy of Sciences, 1959, pp. 867–888, doi:10.17226/10866.

2 Purpose

The purposes of this document are to:

1) Help readers interpret the City's definition of AI (Section 3);
2) introduce a framework to describe the essential characteristics of an AI tool (Section 4); and
3) provide examples of how to use the typology in practice (Section 5).

This document does establish an AI Typology, providing a common vocabulary across the city to describe AI. This document introduces definitions for key terms, to minimize the ambiguity around their usage for city purposes compared to publicly available material.

This document does not prescribe specific uses of the typology. Nevertheless, the intent is to aid uses across the City like policy development, agency technical assistance, and public engagement about AI.[7]

For **systems architects and engineers**, the typology establishes a minimal framework for understanding how to use AI systems in business processes. For example, the input and output data types enable reasoning about composition in complex systems, namely how the output of one system must be compatible with the input of a second system, in order to connect them together. Reasoning about data flows also helps trace privacy-related risks going from input data to output data, and also determine the security controls and cybersecurity best practices to use to protect AI Systems.

For **policymakers**, the typology may help the City better understand the diversity of possible AI systems and how they are used. The typology can serve as the framework for further policy development, such acceptable use policies for AI by City staff and third parties, and the best practices for managing AI systems, information privacy or cybersecurity, that directly translate into implementable actions. Additionally, having a common vocabulary of AI facilitates knowledge sharing across agencies, to describe use cases which may already have deployed solutions elsewhere in the City,

[7] *E.g.*, in Oct. 2024, OTI amended guidance for agencies' compliance with Local Law 35 of 2022 (reporting of algorithmic tools) by incorporating the facets of computation and purpose, discussed in this document, into the agency submission template.

and share best practices and learnings that are transferrable across various AI solutions.

For **risk managers**, the typology helps define what is in scope for AI risk management, in line with frameworks like the NIST AI Risk Management Framework[8] and ISO/IEC 23894.[9] Understanding the variety of possible AI systems enables the design of customized and proportionate risk management processes, consistent with ISO standards for risk management.[10] The AI typology also highlights the intersection with other risk management processes, such as:

- Information Privacy risks associated with the use of City data that is subject to citywide policies referenced below in *Appendix C: Related Citywide Policies and Documents,*
- Cybersecurity risks common to all Information Technology (IT) systems as described in NIST frameworks[11] or citywide policies referenced below in *Appendix C: Related Citywide Policies and Documents*,
- Enterprise risk management of business processes across an entire organization, or
- Legal risks more generally.

The documentation of usage also helps detect the possibility of purpose drift or repurposing of systems for unapproved purposes, which may require revising risk management procedures.

For **the public**, the typology may serve as the basis for educational and informative material about how New York City is using (or considering the use of) AI. It continues the work of *New York City AI Principles and Definitions Guidelines* by creating a common language for understanding how AI operates in the City. This typology, by defining what AI is and the different forms it can take, serves as the technical resource which supports

[8] Artificial Intelligence Risk Management Framework (AI RMF 1.0), *National Institute of Standards & Technology*, NIST AI 100-1, Jan. 2023, doi:10.6028/NIST.AI.100-1.

[9] Technical Committee ISO/IEC JTC 1/SC 42, *Information technology - Artificial intelligence - Guidance on risk management*, International Organization for Standardization (ISO), ISO/IEC 23894:2023, Feb. 2023, https://www.iso.org/standard/77304.html.

[10] § 4, Technical Committee ISO/TC 262, *Risk management – Guidelines*, ISO, ISO 31000:2018, Feb. 2018, https://www.iso.org/standard/65694.html.

[11] NIST Cybersecurity Framework (CSF) 2.0, *National Institute of Standards and Technology, U.S. Department of Commerce*, NIST CSWP 29, Feb. 2024, doi:10.6028/NIST.CSWP.29.

future work on communicating the complexities of AI systems, their use, and accountability to laypeople who do not necessarily possess a technical background, but nevertheless deserve to know and understand how the city is using AI systems for their benefit.

3 The city's definition of AI and how to interpret it

The goal of this section is to provide criteria that identify if a system is or is not AI, which are listed in Section 3.6. Leading up to this subsection, Section 3.1 explains how the City's definition of AI, as specified in *New York City AI Principles and Definitions Guidelines*, can be broken down into its essential components. Section 3.2 explains how the City's system-level definition, being at the systems level, differs from other definitions which focus on other components such as the model. Section 3.3 further clarifies the difference between AI systems and non-AI IT systems more generally. Section 3.4 explains how the definition of AI also identifies two separate stages of an AI lifecycle. Section 3.5 describes refinements to the definition that clarify the role that AI plays in providing government services.

3.1 The definition of AI

The typology is built from the definition of AI in *New York City AI Principles and Definitions Guidelines*:[12]

> A machine-based system that can, for a given set of human-defined objectives, make predictions, recommendations, or decisions influencing real or virtual environments. Artificial intelligence systems use machine- and human-based inputs to perceive real and virtual environments; abstract such perceptions into models through analysis in an automated manner; and use model inference to formulate options for information or action.

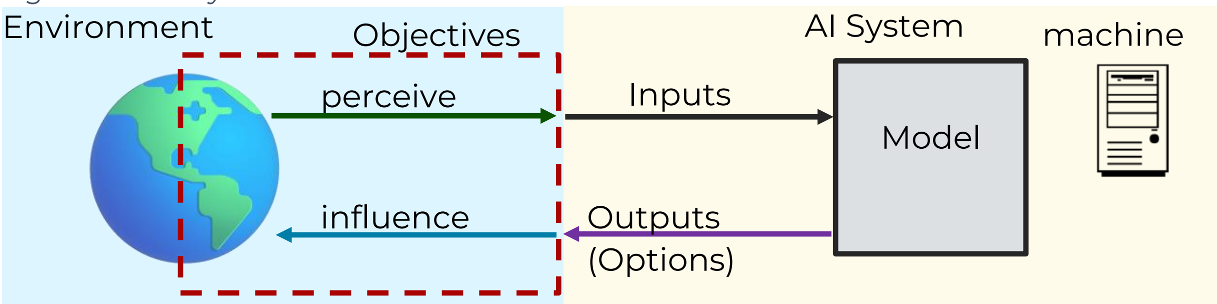
This definition can be broken down into several essential characteristics which are color-coded above and referenced below. Figure 1 summarizes the key components of an AI system according to the definition:[13]

---

[12] *National Artificial Intelligence Initiative Act of 2020*, 15 U.S.C. ch. 119 § 9401(3) (2020), *adopted from*, *Recommendation of the Council on Artificial Intelligence*, Organisation for Economic Co-operation and Development (OECD), OECD/LEGAL/0449, May 2019; *and* *Scoping the OECD AI principles: Deliberations of the Expert Group on Artificial Intelligence at the OECD (AIGO)*, OECD Digital Economy Papers No. 291, OECD, Nov. 15, 2019, doi:10.1787/d62f618a-en.

[13] Adapted from *Scoping the OECD AI principles*, *id*.

Figure 1. An AI system and how it interacts with the external world.



This definition describes the essential characteristics of an AI system, namely that:

- it runs on some computer ("machine-based system"),
- it has some defined purpose ("human-defined objectives"),
- it produces some output ("predictions, recommendations, or decisions") meant to interact with the world ("influencing real or virtual environments"),
- it uses input data ("machine- and human-based inputs") meant to describe the world ("perceive real and virtual environments"),
- it creates a computational model that simplifies the world ("abstract such perceptions into models through analysis in an automated manner"), and
- it uses such a computational model to produce outputs ("use model inference to formulate options for information or action").

3.2 AI as a systems-level concept

The City definition of AI (in Section 3.1 above) is at the systems level, and contrasts with how AI is defined and talked about in other contexts. For example, the computer science literature tends to conflate AI exclusively with the model, ignoring other system components. In the philosophy and cognitive science literature, AI is characterized by its behavior, intent and capabilities to interface with the environment, particularly in the context of ideas like the Turing test,[14] or distinguishing between narrow/weak AI and

[14] *See*, the "imitation game" in A. Turing, *Computing Machinery and Intelligence*, *Mind.* 59 (236): 433–460, Oct. 1, 1950, doi:10.1093/mind/LIX.236.433.

strong AI/artificial general intelligence (AGI).[15] Yet other notions of AI that appear in science fiction and popular culture focus on mechanical form such as supercomputers or robots.[16]

These other notions of AI do not suffice to develop AI governance policy; the context within which AI is being used matters, as does who is responsible for which system components. At the same time, the full definition of AI system is complex and not always needed for every possible discussion of AI. This typology takes a modular approach to describing AI systems with Facets, so that incomplete descriptions of AI are still meaningful, with missing pieces to filled in later as needed.

3.3 Contrasting AI systems with non-AI systems

The City definition of AI above seems very broad and similar to Information Technology (IT) systems more generally. This subsection aims to explain the difference.

IT systems contain computer programs that consists of code that implements an Algorithm, a precise set of instructions hard coded to achieve some predefined outcome.[17] These instructions are like a recipe to produce the dish you want to serve for dinner – they tell you all the ingredients and the exact sequence of actions to make your favorite foods.

AI systems also contain computer programs that also follow Algorithms. However, AI Algorithms do not precisely specify the exact sequence of steps to achieve the goal, but rather describe how to discover and exploit patterns in data. The defining trait of AI systems over non-AI systems is that **AI systems construct their own model of the world by studying data and use that model to produce outputs, rather than following a fixed procedure to produce those outputs**. As a result, the exact same

[15] J. R. Searle, *Minds, brains and programs*, Behavioral and Brain Sciences (3): 3, 417-425, Sep. 1980, doi:10.1017/S0140525X00005756.

[16] *E.g.*, HAL 9000 (A. C. Clarke, *2001: A Space Odyssey*, 1968, Hutchinson: London / New American Library: New York); the original notion of robots from K. Čapek, *RUR: Rossum's Universal Robots*, Aventinum: Prague, 1920; or Data the Android (*Encounter at Farpoint*, Star Trek: The Next Generation, G. Roddenberry, Sep. 28, 1987).

[17] *See*, New York City AI Principles and Definitions Guidelines, City of New York, Mar. 5, 2024, https://www.nyc.gov/assets/oti/downloads/pdf/about/artificial-intelligence-principles-definitions.pdf.

algorithms that create and use AI models can produce different results, simply by using different data. In contrast, algorithms in non-AI systems do not produce variations in the output that depend on data.

3.4 Building and using AI

The definition of AI includes a description of how a computational model is built and used. We introduce the terms *build time* and *use time* to describe the clauses in bold below:

> Artificial intelligence systems use machine- and human-based inputs to perceive real and virtual environments; **abstract such perceptions into models through analysis in an automated manner** (*at build time*); and **use model inference to formulate options for information or action** *(at use time)*.

At build time,

- **training data** ("use machine- and human-based inputs to perceive real and virtual environments") is used by
- learning algorithms to create or update a **computational model** of the external world ("abstract such perceptions into models through analysis"), intended to
- run on some **technology stack** ("machine-based system"), which
- serves some pre-defined **purpose** ("for a given set of human-defined objectives").

At use time,

- the AI system performs further **computations** ("model inference")
- on the **technology stack** ("machine-based system"), which
- produces **outputs** ("make predictions, recommendations, or decisions influencing real or virtual environments"),
- to be consumed (as "options for information or action"), and
- optionally, perform tasks with some degree of **autonomy** for those users ("in an automated manner").

Figure 2. Data flows between an AI system and the external world during Build Time and Use Time.

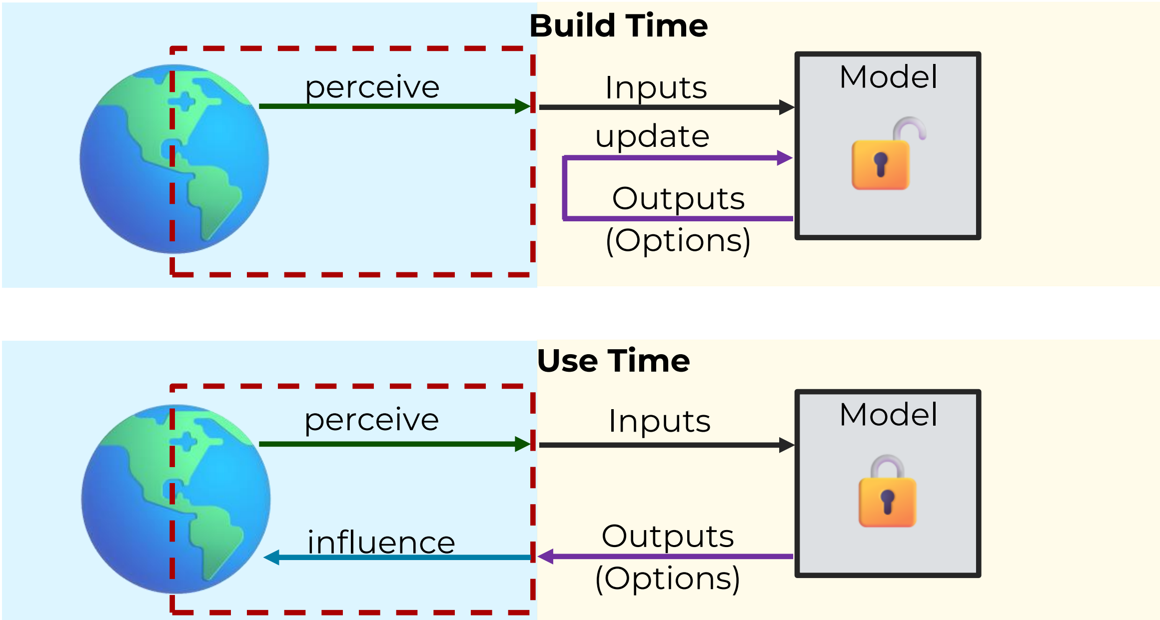


Figure 2 illustrates how information flows between the AI system and the external world during Build Time and Use Time. The formal definitions for these terms are:

Build Time: The part of the lifecycle of an AI system referring to the development of an AI model. At Build Time, i) a model is created or changed, and ii) the environment is not influenced by the AI.[18]

Activities that may be part of Build Time include:

- requirements gathering,
- data engineering, including the redaction of Identifying Information, or information classified as Sensitive and Restricted,
- user studies to collect and prepare the training data,

[18] Adapted from "build phase" and "use phase", Fig. 1, p. 7 *Explanatory memorandum on the updated OECD definition of an AI system*, OECD Artificial Intelligence Papers No. 8, Mar. 2024, OECD Publishing: Paris, doi:10.1787/623da898-en. See also, "model-building" and "model inference" *OECD Framework for the Classification of AI Systems*, OECD Digital Economy Papers No. 323, Feb. 22, 2022, OECD Publishing: Paris, doi:10.1787/cb6d9eca-en.

- legal approvals for the use of data, which may include Identifying Information,
- preparation of computational infrastructure and environments to execute training and inference algorithms,
- training of a new model or updating an existing model, model design and selection,
- hyperparameter tuning or fine-tuning,
- model validation,
- risk assessment and mitigation, including privacy-related risks and cybersecurity risks,
- pre-deployment fairness audits of training data,
- adversarial testing to prevent data poisoning and manipulation,
- regulatory and compliance checks, and
- integration of the AI system into a larger organizational process.

In modern AI systems, Build Time can involve repeated trial and error to build and test many versions of computation models. The number of models tried can be in the thousands or millions.

Use Time: The part of the lifecycle of an AI system referring to the deployment of an AI model to serve its Purpose. At Use Time, i) the model is not changed, and ii) the environment is influenced by the AI.

Activities that may be part of Use Time include:

- logging of performance metrics and unexpected outputs,
- monitoring of operational issues, such as erroneous outputs or downtime,
- monitoring for cyber threats such as unauthorized access and adversarial attacks,

- maintenance of the execution environment of the model,
- collecting user feedback about the AI system's functioning,
- change management, including retraining of models to adapt to new data,
- ongoing risk mitigation, and
- measurement of data drift detection.

3.5 The human elements of AI

The definition of AI, while describing many essential characteristics of AI systems, is not sufficient to describe the context of organizational processes and the stakeholders that interact with the AI system. The goal of this section is to refine the definition by describing these human elements, i.e., the stakeholders responsible for interacting with AI and integrating it into wider processes.

At its core, governments provide public services. These services may be in the form of benefits programs, schooling, call centers, or emergency responses that directly serve the public. Other services may be less direct, serving back-end functions for case work support or document processing, but are nevertheless important for the proper functioning of government.

To describe the responsibilities for fulfilling the purposes of AI systems used to provide services, the basic definition must be expanded to include, at a minimum:

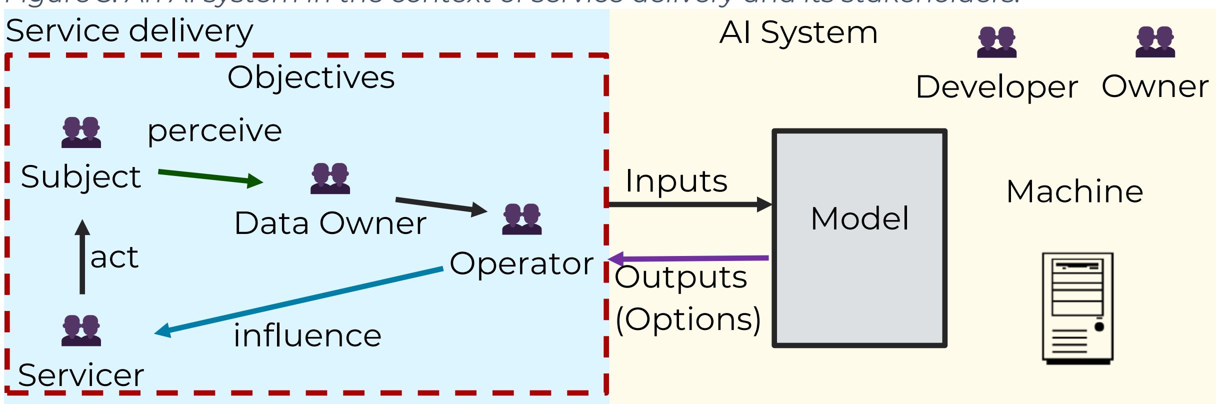
- the **servicer**, who provides the service, and
- the **subject**, who (or which) receives the service.

To describe the responsibilities for building and using AI systems, it is furthermore necessary to describe, at a minimum:

- the **developer**, who builds the AI system,
- the **operator**, who uses the AI system,
- the **data owner**, who is responsible for collecting and using data, and
- the **owner**, who is responsible for the upkeep of the system.

The expanded model from which we define our typology therefore includes these stakeholders explicitly, as shown in Figure 3.

Figure 3. An AI system in the context of service delivery and its stakeholders.



3.6 Criteria for a system to be considered AI

Based on our definition, a system is an AI system if <u>all</u> the following characteristics are met:

Box 1. Criteria for a system to qualify as AI.

- ☐ There are stakeholders responsible for fulfilling six different roles– subject, servicer, operator, developer, owner, and data owner;
- ☐ There is at least one purpose, which can be described as a servicer providing services by acting on a subject;
- ☐ The owner provides computational hardware and software infrastructure that hosts the system;
- ☐ At build time, developers prepare and use training data in a computation algorithm that created or updated some model;
- ☐ At use time, the model uses computation to process input data into outputs;
- ☐ The system provides the outputs to the operator through some interface; and
- ☐ The operator has a level of control over how the system acts on the outputs and provides them to the servicer.

4 The AI typology

Section 1.2 above described what a typology is, and Section 3 above explained how the definition of AI describes the essential characteristics of AI systems. The goal of this section is to use these characteristics to construct our AI typology. The typology describes each characteristic of interest as a facet, and furthermore specifies the values that each facet can take.

Box 2 describes the facets of the AI typology, with the values assignable to each facet described in subsequent subsections.

Box 2. The facets of the AI typology.

1. **Stakeholders**: who are the individuals or entities that build and run the AI system, and who are the individuals or entities that describe the service supported by that AI system?
2. **Purpose**: What is the AI intended to do?
3. **Computation:** What Input Data is provided to the AI model, and what Output Data is produced?
4. **Infrastructure** : What technological components support the functioning of the AI model?
5. **Distribution** : How is the functionality of the AI model provided?
6. **Autonomy**: How much control do the Stakeholders have over the outputs?

4.1 Stakeholders

The **Stakeholder** facet describes an entity who, together with other stakeholders, is responsible for building, using and managing the AI system, or is otherwise affected by the AI system. This facet is subdivided into six subfacets, each corresponding to a Stakeholder as illustrated above in Figure 3.

(i) The **Subject** is the target for whom data is collected about, and the recipient of services provided.

At Build Time, data about Subjects are collected by the Data Owner, to be provided to the Developer to train the AI model.

At Use Time, data about Subjects are collected by the Data Owner, to be provided to the Operator to use the AI model; the Servicer acts on the Subjects based on the outputs of the AI model provided by the Operator

Note on information privacy: Data about Subjects often qualify as Identifying Information; if so, such data is subject to the *Citywide Privacy Protection Policies and Protocols*.

(ii) The **Owner** is the entity who: at Build Time, identifies the purpose for the AI and supervises its development or procurement; and at Use Time, is responsible for overall management of the AI system.

(iii) The **Data Owner** is the entity who is responsible for collecting data about Subjects and governing the use of such data.

(iv) The **Operator** is the entity who, at Use Time, provides the input data to the computational model, oversees the execution of the computational model, and obtains its outputs.

(v) The **Developer** is the entity who, at Build Time, constructs the AI system.[19]

(vi) The **Servicer** is the entity who, at Use Time, consumes the outputs of the AI system and is responsible for rendering services to the Subject.[20]

The Owner, Data Owner, Operator, Developer, and Servicer Stakeholders may each be one or more of the following:

- **City Entity:** An agency, office or other governmental organization that is part of the City of New York. The City Entity can be stated more specifically by identifying the individual agency, office or subdivision thereof, when needed to understand coordination across agencies or organizational subunits.
- **Non-City Entity:** An organization outside of the City of New York. Where necessary, Non-City Entity may be further refined such as corporation, community organization, or other government.

[19] In much of the literature on AI governance, the Developer and Owner Stakeholders are often conflated: it is assumed that the team responsible for developing the system is also responsible for its operational use. However, in the context of the City, most AI systems are procured from vendors rather than developed in-house, which necessitates the distinctions between the two Stakeholders based on their distinct roles at Use Time and Build Time.

[20] The Servicer is not relevant for Fully Autonomous AI systems.

The Subject Stakeholder may be human or non-human, depending on the Purpose. The Subject can be one or more of the following:[21]

- **Individual:** A specific resident, visitor or other human relevant to the city.
- **Group:** A collection of individuals, which may be either formally incorporated as an organization or business, or informally aggregated.
- **Geographic Space:** A spatial region consisting of some combination of natural elements like soil, vegetation and wildlife; manufactured elements like buildings and infrastructure; and human elements of society and culture from its inhabitants.
- **Property:** A possession owned by an individual or entity.
- **Biological Sample:** Any material that originates from a living organism which is collected for the purpose of analysis.
- **Other:** Any other Subject not belonging to any of the categories above.

4.2 Purpose

The **Purpose** facet of the typology defines the role the AI system is intended to play in city government. The Purpose is typically defined by the Owner.

In essence, the Purpose facet defines the possible answers to complete the sentence:

The purpose of this AI system is to perform ________________ (some purpose) on/about/for ___________ (one or more Subjects).

The Purpose may take one or more of these values:

Data Management systems are those providing access to data, storage or monitoring the storage of data, and controlling input-output operations within a data processing system.

Data management systems may perform one or more of tasks such as:

- deduplication of redundant entries,

[21] *See*, "populations impacted" taxonomy, Local Law 35 of 2022.

- data cleaning by identifying anomalous entries,
- data validation ensuring that values are within expected ranges,
- imputing missing data,
- redaction of any sensitive or private data,
- data fusion to combine data records from multiple data sources, and
- disambiguating different entities with the same name.

Examples: Differentiating between voters that share the same name, removing instrument readings that are unphysical, flagging log entries of events with a negative time duration, combining records across different databases to provide a complete case history.

Note on cybersecurity: Data Management systems should include appropriate cybersecurity measures, such as log ingestion for threat hunting, and incident response, in alignment with citywide cybersecurity policies. See *Appendix C: Related Citywide Policies and Documents below*.

Information Presentation systems organize, summarize and display text and/or data in formats designed for communicating facts, concepts or results to an audience.

Information Presentation can involve:

- the creation of reports, charts or presentation slides representing the results of data analysis,
- the conversion of information in one medium to another medium (e.g., between audio, video, and textual formats),
- the summarization of information into a shorter form, or
- the expansion of information into a richer form.

Examples: A data visualization dashboard summarizing overall system health, generating a report of key activities for public education, summarizing a corpus of documents to highlight common trends, providing text captions for videos in multiple languages, transcribing audio recordings of interviews to provide meeting notes, data enrichment of internal records by cross-referencing third-party sources.

Performance Evaluation systems are those that judge whether, or measure of the extent to which, the outputs of a system fulfill some predefined requirements. These requirements may be set by compliance, legal, contractual, regulatory, or other procedural and organizational needs.[22]

Examples: Evaluating the number of cases correctly handled to measure staff productivity, monitoring an air quality sensor for anomalously hazardous pollution instrument readings, sensing traffic light failures to report the mean time between failure, and monitoring customer service interactions to determine the fraction of requests which exceed the response time specified in a service-level agreement.

Resource Allocation systems perform assignment and distribution of available resources. Resources may be physical, virtual, or human.

Examples: Access to seats in a public benefits program with finite capacity, assignment of teachers to classrooms, dispatch of response vehicle to emergency site.

Risk Management systems coordinate activities to direct and control an organization regarding Risk.[23] **Risk** itself is the effect of uncertainty on objectives, i.e., how a deficiency of knowledge has a consequence on a result to be achieved.[24]

Risk management can be broken down into several stages of a lifecycle:[25]

---

[22] *Adapted from* Technical Committee ISO/TC 176/SC 2, *Quality management systems — Requirements*, ISO 9001:2015(en), § 9 "performance evaluation", https://www.iso.org/standard/62085.html.

[23] *Adapted from* Technical Committee ISO/TC 262, *Risk management — Guidelines*, ISO/IEC 31000:2018(en), § 3.2 "risk management" a.k.a. "organizational risk", https://www.iso.org/standard/65694.html.

[24] Adapted from Technical Committee ISO/TC 262, *Risk management — Vocabulary*, ISO 31073:2022(en), § 3.3.1 "risk", https://www.iso.org/standard/79637.html.

[25] ISO/IEC 31000:2018(en) as cited above, § 6.

1. Risk Identification,[26]
2. Risk Analysis,[27]
3. Risk Evaluation,[28]
4. Risk Treatment,[29]
5. monitoring and review, and
6. recording and reporting.

The first three stages are also collectively known as Risk Assessment.[30] Risks can come in many forms and categories, for example: financial risks like credit risk, cybersecurity risks like data breach, public safety risks like the occurrence of a natural disaster, or legal risks like contractual liability.

Examples: Identifying students at high risk of failure, timing traffic lights to regulate the flow of traffic through a busy road, provisioning of backup generators ahead of an anticipated storm event.

Triage systems perform initial assessments of incoming cases to determine their importance and assignment of priorities for follow-up action.[31] They

[26] The process of finding, recognizing and describing Risks. (From ISO 31073:2022(en) as cited above, § 3.3.9 "risk identification".

[27] The process to comprehend the nature of Risk and to determine the level of Risk. Risk Analysis provides the basis for Risk Evaluation and decisions about Risk Treatment. From ISO 31073:2022(en), *supra*, § 3.3.15 "risk analysis".

[28] The process of comparing the results of Risk Analysis with Risk criteria to determine whether the risk is acceptable or tolerable. From ISO 31073:2022(en), *supra*, § 3.3.25 "risk evaluation".

[29] The process to modify Risk. Risk Treatment can involve: avoiding the risk by deciding not to start or continue with the activity that gives rise to the risk; taking or increasing risk in order to pursue an opportunity; removing the risk source; changing the likelihood; changing the consequences; sharing the risk with another party or parties [including contracts and risk financing]; and retaining the risk by informed decision. Risk Treatments that deal with negative consequences are sometimes referred to as "risk mitigation", "risk elimination", "risk prevention" and "risk reduction". Risk Treatment can create new risks or modify existing risks. From ISO 31073:2022(en), *supra*, § 3.3.32 "risk treatment".

[30] The overall process of Risk Identification, Risk Analysis, and Risk Evaluation. From ISO 31073:2022(en), *supra*, § 3.3.8 "risk assessment".

[31] *Adapted from* Technical Committee ISO/TC 46/SC 10, *Information and documentation — Emergency preparedness and response*, ISO 21110:2019(en), https://www.iso.org/standard/69922.html, § 3.17 "triage"; *and* Technical Committee ISO/TC 309, *Whistleblowing management systems — Guidelines*, ISO 37002:2021(en), https://www.iso.org/standard/65035.html, § 3.12 "triage".

apply one or more criteria to determine the most important items that require attention through a follow-up action.

Examples of criteria: Financial cost, proximity of a vehicle to a response location, urgency of request, severity of risk.

Examples of follow-up action: Dispatch of a vehicle, opening an investigation, approval of staff roster, or other activities that can be categorized as Resource Allocation or Performance Evaluation.

Examples of Triage: Determining the magnitude of response to an emergency incident, assessing the degree of need in a social services case.

4.3 Computation

The **Computation** facet of the typology describes the Algorithm by which an AI system processes the input data and produces the output data. The level of description in the typology is necessarily high-level, omitting many details necessary to specify algorithms precisely. **The typology deliberately excludes details of the internal structure of Models, the Algorithms that build and train Models, and how the internal state of Models change using Build Time and Use Time.** Nevertheless, the Computation facet captures the essential nature of those Algorithms at the level of a Function, by identifying the Data Types of the inputs and outputs. A complete description of Data Types can be found below in *Appendix A: The data typology*.

The values of the Computation facet often differ between Build Time and Use Time. **Training Algorithms** are Algorithms that are executed at Build Time to produce computational models. **Inference Algorithms**, in contrast, are Algorithms that are executed at Use Time to produce the desired outputs. Such algorithms often involve Optimization[32] to tune parameters, hyperparameters, and other configuration settings. Unless otherwise specified, the Computation facet is assumed to refer to Inference Algorithms, i.e., Use Time.

[32] "The computation of a solution to maximizing or minimizing an objective function, often subject to constraints on the variables", *New York City AI Principles and Definitions Guidelines*.

Further guidance on how to determine the Computation type can be found in *Appendix B: Assigning the Computation Type*. To facilitate review of this section, examples provided below are numbered.

The value of the Computation facet is one of the following:

Classification: A Function whose output Data Type is a Nominal Data Type or Ordinal Data Type.

In Machine Learning, a Classification model is a type of supervised learning algorithm used to categorize input data into predefined categories. It is sometimes useful to further distinguish Classification into two subtypes. If the categories are ordered, the categories are considered an Ordinal Data Type and the model is said to be an **Ordinal Classification** model; otherwise, the categories define a Nominal Data Type and the model is a **Nominal Classification** model.

Example 1: A Data Management system performs optical character recognition (OCR) to scan invoices received from vendors and issue payments automatically. Based on the format, layout and text identified within the document, the system labels whether each invoice contains enough documentation to pay the vendor. If the documentation is deemed acceptable, the payment is issued, otherwise the document is rejected, and no payment is issued. The output of this system is a Nominal Data Type, which can take the possible values of “accepted” or “rejected”. Therefore, this system performs a Classification computation. Specifically, it is a Nominal Classification model because the output categories are not ordered; there is no intrinsic order to acceptance or rejection.

Example 2: A Triage system contains a model which computes a probability that an individual will be at risk of homelessness within the next 24 months. If the probability exceeds 20%, the individual is considered high risk; if the probability is between 5% and 20%, the individual is considered medium risk; otherwise, the individual is considered low risk. The numerical probability is not relevant for the Triage purpose; only the low, medium or high-risk category matters for subsequent processing. In this case, the model is a Classification model. Specifically, it is an Ordinal Classification model because the output categories are ordered, low < medium < high.

Example 3: A traffic monitoring system contains three components: i) a computer vision component which identifies moving vehicles on a road; ii) a second component which measures the speed of each vehicle; and iii) a third component which determines if a vehicle is speeding and issues a moving violation ticket. The output type of the third component is an Ordinal Data Type, which takes the possible values “issue ticket” or “no action”, and is the final output of the entire system. This computation type of this system is therefore Classification.

Note: Contrast with Example 35 below, which describes a Scoring system.

Example 4: A clustering analysis is performed to identify common patterns of network traffic usage in a network logging and monitoring system. The resulting clustering analysis is then used to identify anomalous traffic patterns which may be evidence of a cybersecurity incident. The output of the anomaly detection model is a risk score, to which a threshold is then applied to determine if the anomaly should be reported. In this situation, a Clustering Computation is performed to create the anomaly detector. However, the anomaly detector itself, whose purpose is Risk Management, has a Computation type of Classification.

Other examples of Nominal Classification models:

Example 5: Assigning a demographic group to a voter record.

Example 6: Deciding if an application for benefits should be accepted or rejected.

Example 7: Identifying an image as a picture of a dog or cat

Example 8: Determining which retention policies should be assigned to a document.

Other examples of Ordinal Classification models:

Example 9: Assigning a letter grade to a business's energy efficiency rating.

Example 10: Determining a candidate to be of low, medium or high risk.

Example 11: Categorizing an anomalous instrument reading into one of five levels of incidence severity.

Other examples:

Example 12: Predicting the dollar value of a human service intervention.

Example 13: Estimating the creditworthiness of a rental applicant.

Example 14: Counting the number of pedestrians in an image captured by a traffic camera.

Clustering: A Function which takes as input a Collection, and assigns each element of the Collection a label whose Data Type is a Nominal Data Type, Ordinal Data Type or Tree.

In Machine Learning, Clustering is a popular form of unsupervised learning which identifies common subgroups of data based on mutual similarity. Each data point is assigned a label which defines the group it belongs to. It is sometimes useful to further refine Clustering into subtypes depending on the output Data Type. **Nominal Clustering** is the most common; the group labels are unordered and hence constitute a Nominal Data Type. **Ordinal Clustering** describes a less common case where the output group labels are ordered, and hence constitute an Ordinal Data Type. Finally, **Hierarchical Clustering** is a special type of Clustering which learns groups with nested structures; the labels then define a Tree.

Examples of clustering:

Example 15: Grouping documents into common themes.

Example 16: Grouping vendors by common product offering to identify potential competition opportunities.

Example 17: Grouping sensors by their typical measurement patterns to identify potential malfunctions.

Example 18: Identifying common segments of behavior in pedestrians to identify potential bottlenecks in road design.

Example 19: Discovering phylogenetic strains in a population of microorganisms.

Forecasting: A Function which takes as input a Sequence whose elements are of some Data Type, X, and outputs either a value of Data Type X or another Sequence whose elements are of Data Type X.

Example 20: A water consumption system collects historical water usage data, measured on an hourly basis, for the past 90 days, and uses a Predictive Model to predict water usage for the next 24 hours. The computation in the Predictive Model takes as input a Sequence of Numeric Data Type, whose values are the historical hourly water consumption rate, and outputs a Sequence of Numeric Data Type, whose values are the predicted hourly water consumption rate for the next 24 hours. The computation type is therefore Forecasting.

Example 21: An elevator control system collects historical passenger traffic in each elevator, measured in 5 second intervals, over the past 24 hours. When an elevator call button is pressed, the system uses a Predictive Model to forecast how many passengers will board on that floor. This computation takes as input a Sequence of Numeric Data Type, whose values are the number of passengers in the elevator in each interval, and outputs a Numeric Data Type, whose value is the predicted passenger load when the elevator stops at the called floor. The computation type is therefore Forecasting.

Other examples:

Example 22: Estimating the number of riders on the subway over the next 30 days.

Example 23: predicting the demand for emergency services over a holiday weekend.

Matching: A Function which takes as input two Collections, and pairs each element of the first Collection with an element from the second Collection.

A Matching computation often involves the use of Optimization algorithms to produce the best possible match, often in the presence of constraints.

Note: Matching models are not to be confused with Ranking models (see below). The former produces pairs between two sets of inputs, whereas the latter produces an ordered list selected from one set of inputs. Matching models may use Ranking models internally to generate the best candidates to pair with another item.

Example 24: A rostering system assigns workers based on the number of workstations available for a given work shift. Due to project requirements, there are preferences to situate some workers togethers and others apart. In addition, workplace accommodations result in requirements for some workers to only use specific workstations. Some workers are on vacation, and some workstations are malfunctioning and unavailable for the current work shift. The inputs to the rostering system are two Collections of Nominal Data Type, the first being the set of available workers and the second being the set of available workstations. The output of the rostering system is a Collection of Pairs, each Pair consisting of a worker assigned to a workstation. The computation type is therefore Matching.

Other examples:

Example 25: Assigning students to seats in schools.

Example 26: Scheduling the dispatch of emergency vehicles to incident locations.

Example 27: Assigning residents to available apartments.

Ranking: A Function which takes as input a Collection and another input, and produces a Sequence whose elements are selected from the Collection.

Note: Ranking models are not to be confused with Matching models. The former produces an ordered list selected from one set of inputs, whereas the latter produces pairs between two sets of inputs. Ranking models may be used inside Matching models to generate the best candidates to pair with another item.

Note: Ranking models are not to be confused with Sampling models (see below). The former produces an ordered list of outputs, whereas the latter produces an unordered set of outputs. If the order matters, it is Ranking; otherwise, it is Sampling. Examples 30 and 33 illustrate the difference.

Example 28: A traffic plate reading system is connected to a database of registered vehicle license plate numbers. When given an image of a license plate, the system performs OCR and returns the top ten matches to known plate numbers in the database. The inputs are of type Image and Collection of Nominal Data Type, representing the picture of the license plate and the database of license plate numbers respectively. The outputs are of type Sequence of Nominal Data Type, being a sorted list of license plate numbers ranked by the likelihood of a correct match within the system. The computation type is therefore Ranking.

Example 29: A chatbot on a website invites visitors to describe the issue they are having and recommends several follow-up actions that may help resolve their issue. A component of the chatbot system employs natural language understanding (NLU) to infer the intent of the free-form text input by the user, and returns the top three responses from a predefined database of permitted responses that best match the intent of the input. This component takes as inputs Text and a Collection, the latter being the list of permitted responses, and produces as output a Sequence that is the top three responses that best match the inferred intent. Therefore, the computation type of this component is Ranking.

Example 30: An inspector needs to inspect apartments in a given neighborhood, but has limited time and can only afford to inspect a few apartments. The system that produces the roster takes as input a Collection of all apartments to be inspected, and produces a Sequence of apartments ranked by the likelihood of having issues to be fixed. The computation type is therefore Ranking. Contrast with Example 33 below.

Other examples:

Example 31: Recommending products to purchase from a vendor catalog based on a description of the desired item.

Example 32: Returning best matches from a database of known suspects matching an image from a security camera.

Sampling: Function which takes as input a Collection, and produces a Collection whose elements are selected from the Collection.

Note: Sampling models are not to be confused with Ranking models. The former produces unordered set of outputs, whereas the latter produces an ordered list of outputs. If the order matters, it is Ranking; otherwise, it is Sampling. Examples 10 and 11 illustrate the difference.

Example 33: An inspector needs to inspect all apartments in a given neighborhood over the course of a year. At the beginning of each week, they produce a roster of apartments to be inspected. The order of inspection of the apartments does not matter; the only requirement that all the selected apartments must be inspected by the end of the week. The system that produces the roster takes as input a Collection of all apartments that have yet to be inspected, and produces a Collection of apartments to be inspected in a given week. The computation type is therefore Sampling. Contrast with Example 30 above.

Scoring: A Function whose output Data Type is a Numerical Data Type.

In Machine Learning and statistics, Scoring models are also known as regression models.

Note: Scoring models are not to be confused with Classification models, whose outputs are strictly of Nominal Data Type or Ordinal Data Type. Scoring models output a number, whereas Classification models output a category. A Scoring model may be used inside a Classification model to generate a number; the Classification model can then take the numerical output of the Scoring model, compare it to a threshold, and finally determine the final category to assign based on whether the threshold was met. Examples 3 and 35 illustrate the difference.

Example 34 1: A Performance Management system contains a model which computes a score for measuring an employee's productivity. The productivity score is used directly to determine the discretionary bonus for that employee. In this case, the model is a Scoring model.

Example 35: A traffic monitoring system contains two components: i) a computer vision component which identifies moving vehicles on a road and labels each vehicle in every video frame; and ii) a second component which, given relevant video frames for each vehicle, computes the speed of that vehicle. The output type of the second component is a Numeric Data Type, which is the speed of the vehicle. The output of the second component is also the final output from the entire system. Therefore, the computation type of this system is Scoring.

Data Transformation: Any Function which does not fit into any of the other categories of Computation.

Example 36: A retrieval-augmented generation (RAG) system[33] provides a natural language interface where users can enter queries in the form of natural language, free-form text. There are four internal components: (i) the embedding model which converts the input free-form text into numeric vector form, (ii) the semantic query component, which finds the vectors representing authoritative reference documents that are most similar to the input vector, (iii) the synthesis model, which combines the user input with information retrieved from the best matching documents, and (iv) the autoregressive model, which is a Generative AI model which takes the output from (iii) and produces free-form text output.

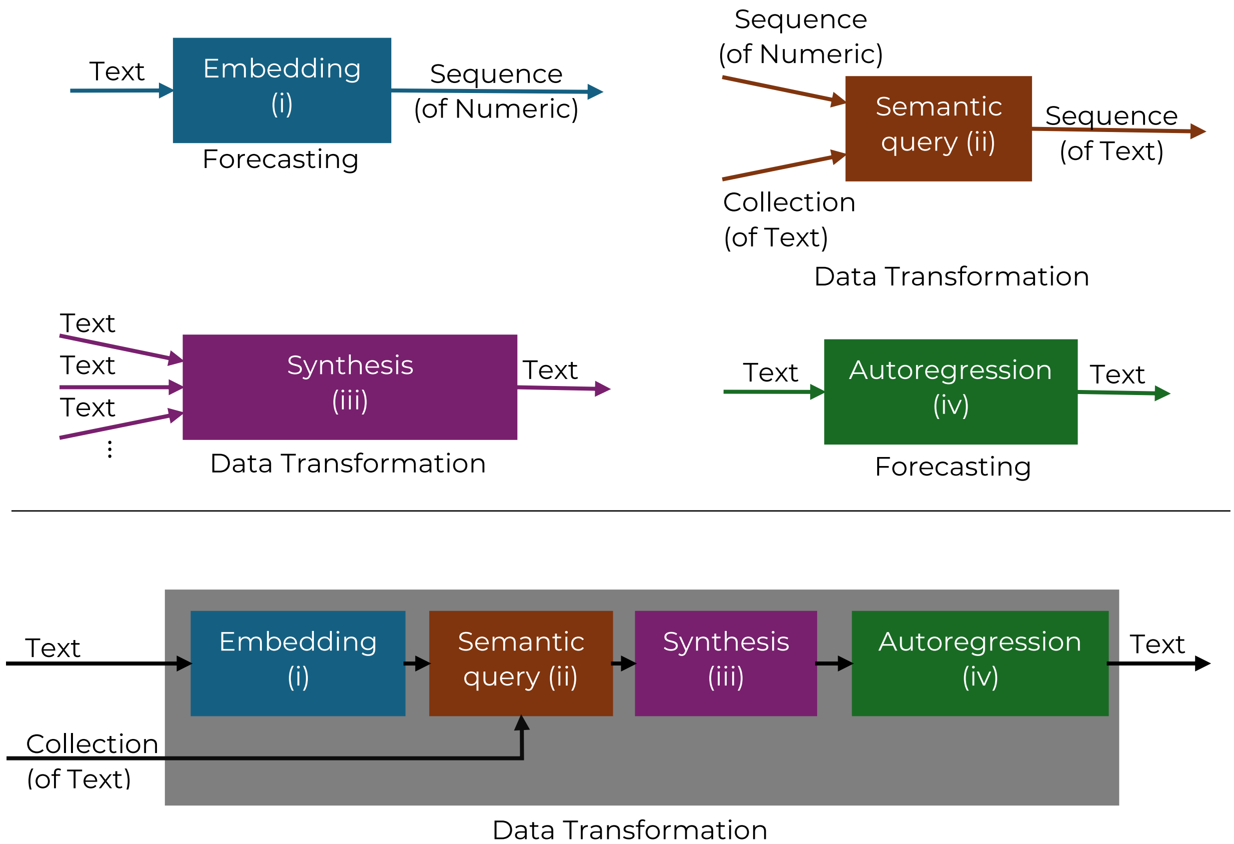
The computation types of each component are:

(i) Forecasting, taking as input Text (i.e., a special type of Sequence) and outputs a vector (i.e., Sequence of Numeric Data Type),
(ii) Ranking, taking as input a vector and a Collection of document vectors and outputs a Sequence of best matching documents (as Text),
(iii) Data Transformation, taking as input multiple Texts and outputs a single Text, and
(iv) Forecasting, taking as input Text and outputs Text.

These four components are illustrated in Figure 4 below.

[33] S. Riedel, D. Kiela, P. Lewis, & A. Piktus, *Retrieval Augmented Generation: Streamlining the creation of intelligent natural language processing models*, Meta AI blog, Sep. 28, 2020, https://web.archive.org/web/20250414173957/https://ai.meta.com/blog/retrieval-augmented-generation-streamlining-the-creation-of-intelligent-natural-language-processing-models/.

Figure 4. The input Data Types, output Data Types, and classification of Computation Types for a retrieval-augmented generation (RAG) AI system, as described for each individual component (above the line), and as an entire system (below the line).



Tracing the inputs and outputs of the composite system, the overall input is Text and a Collection of documents, and overall output is Text. The computation type of the whole system is therefore Data Transformation, as illustrated in Figure 4.

Other examples:

Example 37: Real-time captioning systems which take video input and output closed captions in the form of text.

Example 38: Meeting transcription systems which take audio recordings of meetings and output textual minutes.

Example 39: Illustration systems which take text narratives and generate a sequence of images to accompany the text.

Note on information privacy: The output data from AI systems, especially of Data Transformation systems, may consist of Identifying Information. Such data is subject to the *Citywide Privacy Protection Policies and Protocols*.

4.4 Infrastructure

The **Infrastructure** facet describes the "machine" behind the "machine-based system". Specifically, it refers to the collection of IT components that collectively support the functionality of an AI system.

In essence, the Infrastructure facet defines the possible answers to complete the sentence:

> The machine that performs the Computation of an AI system is ____________________ (some Infrastructure).

By necessity, the typology omits many details that are needed to fully describe the hardware and software components that collectively describe this infrastructure.

The value of the Infrastructure facet is <u>one or more</u> of the following:

- **Edge Device**: Any component that perceives input data and performs Computation, using input data perceived from the same component or subsystem. Examples of Edge Devices include wearables like smart watches and smart glasses; appliances like a robot vacuums and smart washing machines; and robots.

- **Server:** A centralized component designed to perform Computation, with inputs received from, and outputs sent to, other components connected over a network. Servers may be further subcategorized into **On-Premise Servers** and **Cloud Servers**: the former are owned by the Owner of the AI system, while the latter are not.

- **Personal Computing Device:** A general-purpose computing device designed for individual use. Examples of Personal Computing Devices include personal computers, laptops, and smartphones.
- **Other:** Any other Infrastructure component. Examples include sensors, network routers and edge gateways.

In practice, many AI systems use hybrid infrastructure combining multiple choices above. For example, Apple Intelligence uses both Personal Computing Device and Cloud Server as its Infrastructure.[34]

4.5 Distribution

The **Distribution** facet describes the mode by which an AI system is structured within a larger system, with particular regard to how such functionality is provisioned to an Operator.

In essence, the Distribution facet defines the possible answers to complete the sentence:

> The AI system is provided to the Operator
> ________________ (through some Distribution).

The value of the Distribution facet is <u>one or more</u> of the following:

- **Standalone:** The provision of AI through software that is procured or developed as a self-contained product.
- **As-a-Feature:** The provision of AI through an existing software system as a component that offers new or enhanced functionality.
- **As-a-Service:** The provision of AI through an application programming interface (API).

4.6 Autonomy

The **Autonomy** facet describes the extent to which the Operator is involved in acting on the AI system's outputs.[35]

---

[34] T. Gunter, *et al.*, *Apple Intelligence Foundation Language Models - Apple Machine Learning Research*, July 29, 2024, https://machinelearning.apple.com/research/apple-intelligence-foundation-language-models, *and* arXiv:2407.21075.

[35] *See*, *OECD Framework for the Classification of AI Systems*, *supra*, p. 53 "Action autonomy level".

The value of the Autonomy facet is one of the following:

- **Informative:** The AI system does not act on its output. The Operator uses or disregards the AI system's outputs at will.
- **Supervised:** The AI system can act on its outputs, but requires explicit permission from the Operator to do so.
- **Monitored:** The AI system acts on its outputs unless the Operator overrides the action.
- **Fully Autonomous:** The AI system acts on its outputs without any Operator intervention.

5 Using the typology

The AI Typology described in Section 4 above describes multiple facets, and permissible values for each facet. To assign a Type to an AI system, specify the value (or values) that correspond to each facet, as summarized in Table 1.

The value of certain facets may sometimes be unknown, irrelevant or otherwise missing. Such partial descriptions are still possible with the AI Typology, but would not constitute a fully specified Type.

Table 1. Values for each facet of the AI typology.

Facet	Subfacet	Values
Stakeholders	Subject	Individual Group Geographic Space Property Biological Sample Other
	Owner	City Entity Non-City Entity
	Data Owner	City Entity Non-City Entity
	Operator	City Entity Non-City Entity
	Developer	City Entity Non-City Entity
	Servicer	City Entity Non-City Entity
Purpose		Data Management Information Presentation Performance Evaluation Resource Allocation Risk Management Triage
Computation		Classification Clustering Forecasting Matching Ranking Sampling

	Scoring Data Transformation
Infrastructure	Edge Device On-Premise Server Cloud Server Personal Computing Device Other
Distribution	Standalone As-a-Feature As-a-Service
Autonomy	Informative Supervised Monitored Fully Autonomous

The following subsections describe some typical examples of AI systems and their assigned types.

5.1 Example: a social service risk prediction AI

Scenario: An agency has developed, and is using, an AI system to calculate the risk of recidivism for individuals in their system. An agency case manager enters information about their client into a case management software running on their local desktop. An algorithm, created by the agency and embedded into vendor software, produces a likelihood of recidivism. If the client is in the top 10th percentile, they are flagged to the case manager as eligible for additional services.

The input and output Data Types for the AI model are:

Inputs	Outputs
Unknown	**Nominal Data Type** (individual is or is not eligible for services)

The assigned Type for this AI system is:

Type		
Stakeholders	Subject	Individual
	Owner	City Entity

	Data Owner	City Entity
	Operator	City Entity
	Developer	City Entity
	Servicer	City Entity
Purpose		Resource Allocation and Risk Management
Computation		Classification
Infrastructure		Personal Computing Device
Distribution		As-a-Feature
Autonomy		Informative

5.2 Example: a customer service chatbot

Scenario: this chatbot is designed for users to enquire about the nature of services provided by the agency, which depends on their account type, billing status and prior authorizations for services filed with the agency. The user may be an individual or organization. The chatbot was procured from a vendor who is responsible for fine-tuning the chatbot's responses and guaranteeing its ordinary operations on a cloud platform.

The input and output Data Types for the AI model are:

Inputs	Outputs
Text	Text

The assigned Type for this AI system is:

Type		
Stakeholders	Subject	Individual and Organization
	Owner	City Entity
	Data Owner	City Entity
	Operator	City Entity
	Developer	Non-City Entity
	Servicer	City Entity
Purpose		Information Presentation
Computation		Data Transformation
Infrastructure		Cloud Server
Distribution		Standalone
Autonomy		Informative

5.3 Example: a staff performance evaluation system

Scenario: An agency uses an automated system for producing numerical ratings for customer service agents based on written supervisors' reports which are uploaded to a central server owned by the agency. The system was developed by a professional certifying body. The rating is the sole criterion used to determine merit-based raises and cannot be disputed by the agents or the agency.

The input and output Data Types for the AI model are:

Inputs	Outputs
Text	Numerical Data Type

The assigned Type for this AI system is:

Type		
Stakeholders	Subject	Individual
	Owner	City Entity
	Data Owner	City Entity
	Operator	City Entity
	Developer	Non-City Entity
	Servicer	City Entity
Purpose		Information Presentation
Computation		Scoring
Infrastructure		On-Premise Server
Distribution		Standalone
Autonomy		Fully Autonomous

6 References

6.1 List of defined terms

Page numbers in **bold** refer to definitions. Other terms not defined in this document may be found in *New York City AI Principles and Definitions Guidelines.*

6.2 List of Figures

6.3 List of Boxes

6.4 List of Tables

7 History and Authorship

Version	Change Description	Author(s)	Date
1.0	Initial version	Jiahao Chen, Director of AI/ML, OTI Strategic Initiatives Division	1/6/2025
1.1	Updated with further clarifying details ahead of citywide distribution.	Jiahao Chen, Director of AI/ML, OTI Strategic Initiatives Division	5/20/2025
1.1.1	Replaced "Data Generation" with "Data Transformation" for consistency with feedback from AI risk assessment	Jiahao Chen, Director of AI/ML, OTI Strategic Initiatives Division	5/30/2025
1.2	First public version	Jiahao Chen, Director of AI/ML, OTI Strategic Initiatives Division	12/31/2025

Appendix A: The data typology

This Appendix defines a separate typology specifically to describe data.[36]

The Data Types here are defined not by their computer representations in bits and memory, but rather by their programmatic behavior (specifically, their algebraic properties). This design choice allows the data typology to be useful even without detailed knowledge of the internal workings of the AI system or how the data are represented in memory.

A **Numerical** datum is a number that represents a quantity. Arithmetic operations like addition, subtraction, multiplication, and division can be carried out meaningfully.

Examples: dollar amounts, physical measurement, sociological index, risk score, probability, likelihood ratio

An **Ordinal** datum is a label or categorization consisting of two or more categories with an intrinsic ordering between the categorical labels that is relevant to the use case. Given two categories, it is possible to compare them to see which one is greater or better.

Examples: rank order (first choice, second choice), military rank, letter grade, priority level, pass/fail.

A **Nominal** datum is a categorization consisting of two or more categories without an intrinsic ordering between the categorical labels that is relevant to the use case. In general, it is not meaningful to compare any two categories to see which one is greater or better.

If an order can be defined but is not relevant to the use case, we consider the Data Type to be a Nominal Data Type. For example, names of microbial strains can be ordered alphabetically, but this order is not relevant in the biomedical contexts for assessing the severity of an outbreak, so we would consider a microbial strain to be a Nominal Data Type.

[36] Many of these terms have been previously defined in *New York City AI Principles and Definitions Guidelines.*

Examples: bacterial strain, a city agency

An important special case of nominal is the **Image**, which represents information that can be perceived visually.

Here is a summary of elemental data types:

Data Type	Numeric	Ordinal	Nominal
Arithmetic defined	✓		
Order defined	✓	✓	

Numeric, Ordinal and Nominal form the elemental basis for creating compound data types.

The simplest kind of **Tuple** is a **Pair**, which represents a relationship between two data elements.

Examples: coordinates of latitude and longitude (a pair of two numeric elements), an assignment of a single teacher to a single school (a pair of two nominal elements).

Tuple data are commonly referred to as structured data because they occur together in a fixed format, and are often stored in databases, spreadsheets, or specialized file types.

Groupings of data are referred to as Collection, Tree, or Sequence depending on how order between items in the grouping is defined.

The order of appearance of items is not relevant in a **Collection**; common subtypes are **Tabular**, a Collection of Tuples, and **Match**, a Collection of Pairs.

Examples: a random sample of 10 business to inspect (a collection of nominal elements), a resource allocation plan matching employees to desk assignments (a Match)

The order of appearance of items is relevant in a **Tree**, but not all items are ordered in relation to each other.

Examples: A phylogenetic tree, an organizational chart

In a **Sequence**, all items are ordered in relation to each other.

Examples: a dispatch plan of which buildings to visit (a Sequence of Nominal elements), a sequence of deoxyribonucleic (DNA) nucleotides (a Sequence of Nominal elements).

There are several important special types of sequence worth defining:

A **Time Series** is a Sequence whose ordering has the interpretation of progression in time. The elements may themselves be Pairs relating timestamps to some other data.

Examples: stock prices, weather records, economic indicators, electrocardiograms (ECGs)

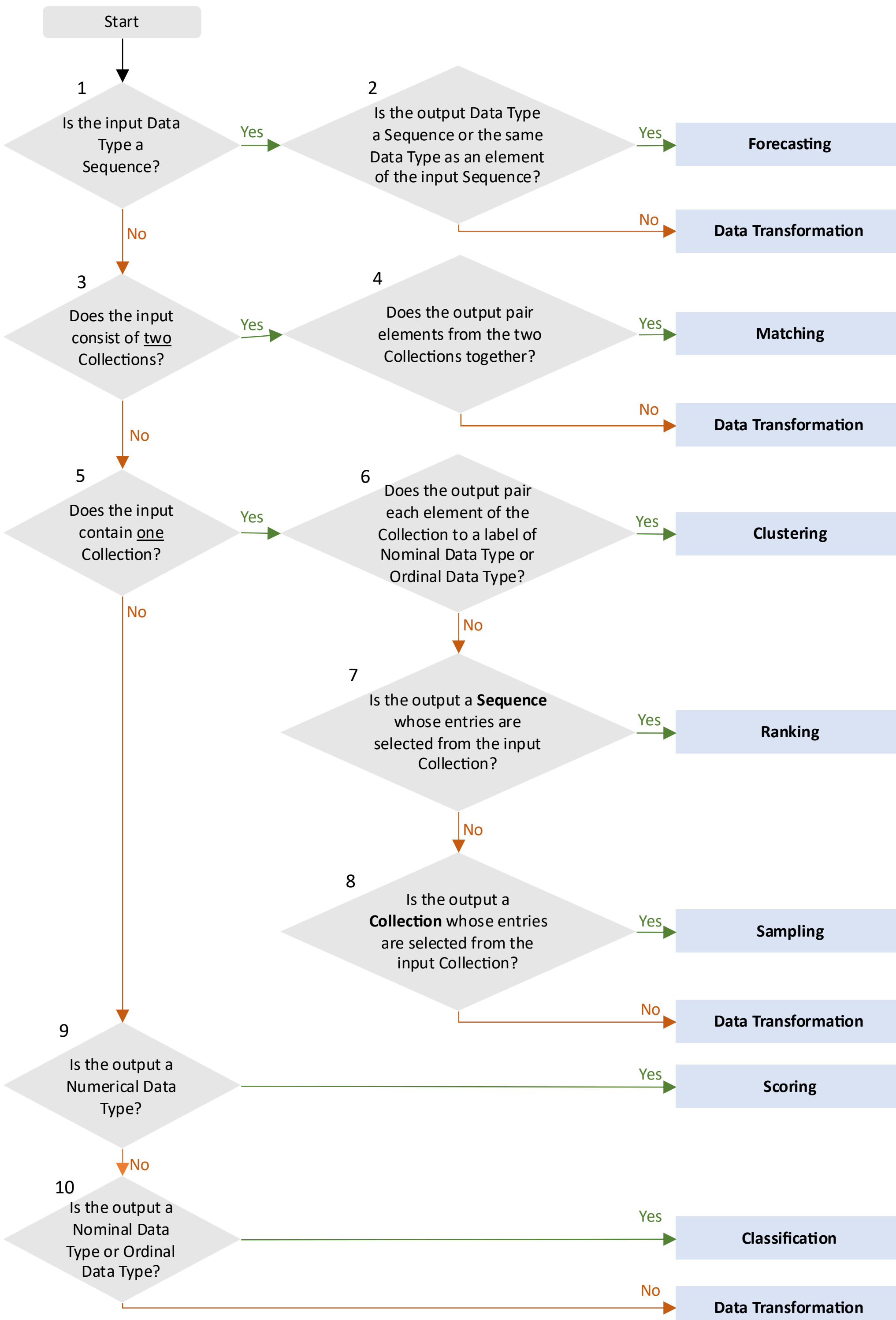
Other Sequences have distinct perceptual qualities, like **Text** (textual or tactile, as a form of writing), **Audio** (aural) and **Video** (visual, possibly aural and textual as well).

Here is a summary of collective data types:

Data type	Tuple	Collection	Tree	Sequence
Number of items	Fixed as part of the definition of the data type	Variable	Variable	Variable
Ordering of items	All items are ordered	Doesn't matter	Some items are ordered	All items are ordered
Special cases	Pair (length 2)	Match Tabular	-	Time Series, Text, Audio, Video

Appendix B: Assigning the Computation Type

The following flowchart shows how to assign the Computation Type.



The numbers in this list correspond to those labelling the decision diamonds on the flowchart.

1) As defined, a Sequence is simply an ordered list of items, with each item (element) having the same data type.
   a) Important examples of Sequences are:

      - Time series like stock prices, weather records, economic indicators, electrocardiograms (ECGs);
      - Text, a Sequence of characters with the perceptual quality of written language;
      - Audio, a Sequence of waveform amplitudes or other Time Series with the perceptual quality of sound;
      - Video, a Sequence of Images with the perceptual quality of vision, possibly accompanied with Audio and Text as well; and
      - Genetic sequences, e.g., of deoxyribonucleic acid (DNA) or ribonucleic acid (RNA) base pairs.

   b) A key defining characteristic of a Sequence is that every item can be sorted into a well-defined order.[37] Examples of such orderings are:
      - Chronological order, with measurements or events sequentially ordered in time;
      - Lexicographic order, with text ordered alphabetically; and
      - Numerical order, with numbers arranged from most negative to most positive.
   c) If only some items can be ordered, the Data Type is a Tree and not a Sequence.
   d) Similarly, if there is no intrinsic order to the items, the Data Type is a Collection and not a Sequence.
2) The output of Forecasting can be a single element or another Sequence which has the interpretation of further extending the input Sequence.
   a) An example of the former is a task to fill in the blank:

      1, 3, 5, 7, 9, __

      This problem can be described with the input (1, 3, 5, 7, 9) and the output as the number 11. The input Data Type is a Sequence (specifically, a Sequence of Numerical Data Type), while the output Data Type is a

[37] In the mathematical literature, this property is known as “total” or “linear” ordering.

Numerical Data Type. Therefore, the Computation Type that produces this output is Forecasting.

b) An example of the former is a task to fill in multiple blanks, which may be described as extrapolation:

1, 3, 5, 7, 9, __, __, __

This problem can be described with the input (1, 3, 5, 7, 9) and the output as (11, 13, 15). The input Data Type and output Data Type are both Sequence of Numerical Data Type. Therefore, the Computation Type that produces this output is Forecasting.

c) Forecasting may include solving problems which may be described in other contexts as interpolation, for example, to fill in the blanks in:

1, 3, 5, 7, 9, __, __, __, 17, 19

This problem can be described with the input (1, 3, 5, 7, 9, 17, 19) and the output solution as (11, 13, 15). The input Data Type and output Data Type are both Sequence of Numerical Data Type. Therefore, the Computation Type that produces this output is Forecasting.

d) Another familiar example of Forecasting is the predictive search feature of search engines or "autocomplete" feature of cell phone keyboards. For example, a keyboard may try to predict the word and symbols that follow the input

How are you ___________?

The computation to solve the problem may be described as Forecasting with the input (How, are, you) and the output (feeling, today, ?). In this example, the input Data Type and output Data Type are both Sequence (specifically, Text).

3 and 4) The action of pairing is also known by other names like "zipping".

a) An example of a pairing action is matching workers to cubicles. Given three workers (1, 2, 3) and three cubicles (A, B, C), one possible output is the table:

Worker	Cubicle
1	A
2	B
3	C

Another way to express the same output is with the allocation matrix:

Worker \ Cubicle	A	B	C
1	✅		
2		✅	
3			✅

The input Data Types to this Matching problem are two Collections – one representing the possible IDs of workers, and the second representing the possible names of cubicles. The output Data Type is a Collection, specifically Collection of Pairs. Because the order of the rows and columns do not matter, the Data Type is Collection and not Sequence. Each item in the Collection is a Pair matching a worker to a cubicle.

b) The pairing action of creating Pairs from two Collections does not necessarily have to match unique items from the two collections. For example, multiple guests could be assigned to the same dining table if the precise seat at the table does not matter. An example of such an output may look like

Table 1	Table 2
Guest A	Guest C
Guest B	Guest D
	Guest E
	Guest F

The computation to allocate guests to tables is Matching, following the logic of the previous example.

5) Collections are distinct from Sequences, as discussed in (1)(d) above.
6) The Computation Type of Clustering describes the act of assigning group labels to items in a Collection.
   a) A simple example of Clustering is to classify animals according to breed and species. For example, the output of Clustering may look like

Breed	Species
Dachshund	Dog

Breed	Species
Poodle	Dog
Scottish Fold	Cat
Siamese	Cat

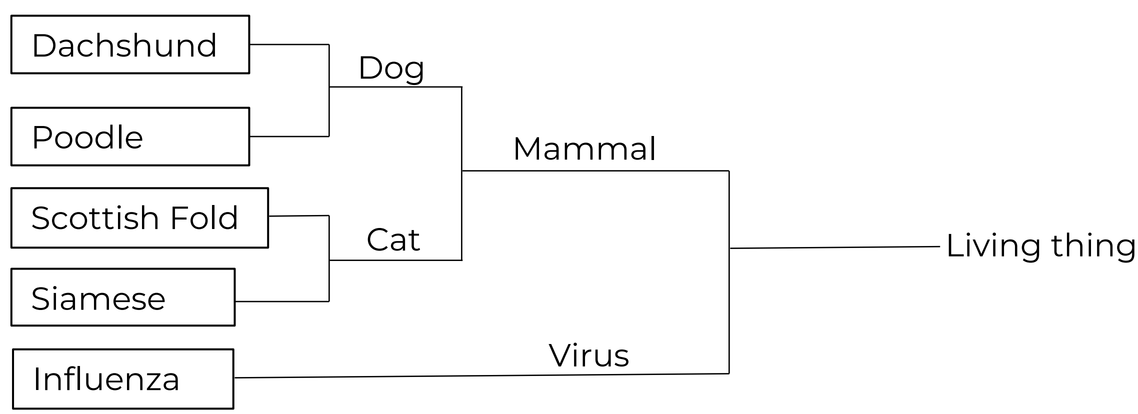
In this Clustering example, the input is (Dachshund, Poodle, Scottish Fold, Siamese), which is of Data Type Collection, and the output Data Type is a Collection of Pairs, and each Pair matches a breed to a species. In reality, most Clustering Algorithms do not assign meaningful labels, but rather arbitrary group names to be interpreted by the end user. It is more likely to encounter output of the form

Breed	Species
Dachshund	A
Poodle	A
Scottish Fold	B
Siamese	B

with the user expected to do further work to identify what Species A and B are.

b) This example is different from Matching because the Collection of Species is not given as input to Clustering. If instead the input to the computation is (Dachshund, Poodle, Scottish Fold, Siamese) and (Dog, Cat), the Computation Type would then be Matching and not Clustering.

c) Hierarchical Clustering is a specific form of Clustering, which is commonly encountered in population genetics. A typical graphical visualization of the output looks like this:



As with (a) above, the output of Hierarchical Clustering does not normally assign interpretable labels to the learned clusters.

d) Clustering can be used to train Classification or Scoring models. However, it is important to distinguish between computations performed at the time of use, which may differ from the computations performed when training or preparing the system for use.

7) The defining characteristic of Ranking is that it creates order where it was previously absent – a Ranking model takes an unordered Collection as input and produces and ordered Sequence as output.

a) A simple example of Ranking is prioritizing a to-do list for the day:

Task
Task A
Task B
Task C
Task D

Priority	Task
1	Task B
2	Task C
3	Task A
(none, don't do it)	Task D

b) In Machine Learning, it is common to determine the rank order by computing an internal score for each item. An example could look like

Task
Task A
Task B
Task C

Priority	Score	Task
1	0.87	Task B
2	0.24	Task C
3	0.07	Task A

If the scores are used purely internally to the ranking computation and not used for the Purpose, this would still be considered a Ranking Computation Type.

8) The difference between Sampling and Ranking is whether or not the output is ordered. In the previous example of selecting tasks to do, a different scenario may involve not prioritizing the tasks to do, but merely selecting which tasks to do in no particular order.

Task
Task A
Task B
Task C
Task D

Task
Task C
Task A
Task B

This example would be considered a Sampling Computation Type and not Ranking.

Examples 9 and 10 above further illustrate how the difference between Sampling and Ranking may arise.

9) The output of a Scoring model often has the interpretation of a probability or likelihood of occurrence. Such Scoring models can be used to construct a Classification model by specifying one or more threshold scores that divides the outputs into discrete categories. If the numerical value of the score is relevant to the Purpose, the model is considered to be a Scoring model. However, if only the final category is relevant to the Purpose, the model is considered to be a Classification model.

a) This illustrative example describes a Performance Evaluation system to evaluate product quality.

Product
Product A
Product B
Product C

Quality Score	Product
99	Product A
63	Product B
12	Product C

If the score is the output that is relevant to the Purpose, then the Computation Type is Scoring: each product receives its own score. If, however, the score is only used internally to the tool to assign a label which determines the actual Purpose, then the Computation Type is Classification and not Scoring.

Product
Product A
Product B
Product C

Quality	Quality Score	Product
High	99	Product A
Medium	63	Product B
Low	12	Product C

b) Another example comes from network log analysis for detecting anomalies that could represent cybersecurity incidents.

Event
Event A
Event B
Event C

Anomaly	Probability	Event
No	0.01	Event A
Yes	0.95	Event B
No	0.04	Event C

In this example, a threshold probability of 0.05 is used to decide if an event is an anomaly, and only events labeled as anomalies are investigated. In this case, the Computation Type is Classification and not Scoring. See Example 4 above for a more detailed discussion.

Appendix C: Related Citywide Policies and Documents

- The New York City Artificial Intelligence Action Plan, Oct. 2023.
- New York City AI Principles and Definitions Guidelines, Mar. 2024.
- New York City Generative AI Preliminary Use Guidance, Mar. 2024.
- Citywide Privacy Protection Policies and Protocols v4.0, Jan. 2025.
- Citywide Cybersecurity Policies and Standards, including:
- Citywide Cybersecurity Program, No. D-ID-GV-01, Oct. 2019.
- Citywide Cybersecurity for the Usage and Development of Artificial Intelligence (AI) Systems Policy, No. P-08-PR-DS, Jan. 2025.

citibank
STOP
NYPD
ONLY
ONLY ONLY
ONLY

Notes

Photo credit: Michael Appleton, New York City Mayoral Photography Office