INTERNATIONAL STANDARD

ISO/IEC 23053

First edition
2022-06

Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

Cadre méthodologique pour les systèmes d'intelligence artificielle (IA) utilisant l'apprentissage machine

Reference number
ISO/IEC 23053:2022(E)

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directives or www.iec.ch/members_experts/refdocs).

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 42, *Artificial Intelligence*.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committees.

Introduction

Artificial intelligence (AI) systems, in general, are engineered systems that generate outputs such as content, forecasts, recommendations or decisions for a given set of human-defined objectives. AI covers a wide range of technologies that reflect different approaches to dealing with these complex problems.

ML is a branch of AI that employs computational techniques to enable systems to learn from data or experiences. In other words, ML systems are developed through the optimisation of algorithms to fit to training data, or improve their performance based through maximizing a reward. ML methods include deep learning, which is also addressed in this document.

Terms such as knowledge, learning and decisions are used throughout the document. However, it is not the intent to anthropomorphize machine learning (ML).

This document aims to provide a framework for the description of AI systems that use ML. By establishing a common terminology and a common set of concepts for such systems, this document provides a basis for the clear explanation of the systems and various considerations that apply to their engineering and to their use. This document is intended for a wide audience including experts and non-practitioners. However, some of the clauses (identified in the overview in Clause 5), include more in-depth technical descriptions.

This document also provides the basis for other standards directed at specific aspects of ML systems and their components.

Framework for Artificial Intelligence (AI) Systems Using Machine Learning (ML)

1 Scope

This document establishes an Artificial Intelligence (AI) and Machine Learning (ML) framework for describing a generic AI system using ML technology. The framework describes the system components and their functions in the AI ecosystem. This document is applicable to all types and sizes of organizations, including public and private companies, government entities, and not-for-profit organizations, that are implementing or using AI systems.

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 22989, *Information technology—Artificial intelligence — Artificial intelligence concepts and terminology*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 22989 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ISO Online browsing platform: available at https://www.iso.org/obp

— IEC Electropedia: available at https://www.electropedia.org/

3.1 Model development and use

3.1.1
classification model
<machine learning> machine learning model whose expected output for a given input is one or more classes

3.1.2
regression model
<machine learning> machine learning model whose expected output for a given input is a continuous variable

3.1.3
generalization
<machine learning> ability of a trained model to make correct predictions on previously unseen input data

Note 1 to entry: A machine learning model that generalizes well is one that has acceptable prediction accuracies using previously unseen input data.

Note 2 to entry: Generalization is closely related to overfitting. An overfit machine learning model will not generalize well as the model fits the training data too precisely.

3.1.4
overfitting
<machine learning> creating a model which fits the training data too precisely and fails to generalize on new data

Note 1 to entry: Overfitting can occur because the trained model has learned from non-essential features in the training data (i.e. features that do not generalize to useful outputs), excessive noise in the training data (e.g. excessive number of outliers) or because the model is too complex for the training data.

Note 2 to entry: Overfitting can be identified when there is a significant difference between errors measured on training data and on separate test and validation data. The performance of overfitted models is especially impacted when there is a significant mismatch between training data and production data.

3.1.5
underfitting
<machine learning> creating a model that does not fit the training data closely enough and produces incorrect predictions on new data

Note 1 to entry: Underfitting can occur when features are poorly selected, insufficient training time or when the model is too simple to learn from large training data due to limited model capacity (i.e. expressive power).

3.2 Tools

3.2.1
backpropagation
neural network training method that uses the error at the output layer to adjust and optimise the weights for the connections from the successive previous layers

3.2.2
learning rate
step size for a gradient method

Note 1 to entry: Learning rate determines whether and how fast a model converges to an optimal solution, making it an important hyperparameter to set for neural networks.

3.3 Data

3.3.1
class
human-defined category of elements that are part of the dataset and that share common attributes

EXAMPLE "telephone", "table", "chair", "ball bearing" and "tennis ball" are classes. The "table" class includes: a work table, a dining table, a study desk, a coffee table, a workbench.

Note 1 to entry: Classes are typically target variables and designated by a name.

3.3.2
cluster
automatically induced category of elements that are part of the dataset and that share common attributes

Note 1 to entry: Clusters do not necessarily have a name.

3.3.3
feature
<machine learning> measurable property of an object or event with respect to a set of characteristics

Note 1 to entry: Features play a role in training and prediction.

Note 2 to entry: Features provide a machine-readable way to describe the relevant objects. As the algorithm will not go back to the objects or events themselves, feature representations are designed to contain all useful information.

3.3.4
distance
<machine learning> measured proximity of two points in space

Note 1 to entry: Euclidean, or straight-line, distance is ordinarily used in machine learning.

3.3.5
unlabelled
property of a sample that does not include a target variable

4 Abbreviated terms

AI	artificial intelligence
API	application programming interface
AUC	area under the curve
BM	Boltzmann machines
CapsNet	capsule neural network
CG	conjugate gradient
CNN	convolutional neural network
DBN	deep belief networks
DCNN	deep convolutional neural network
FFNN	feed forward neural network
FNR	false negative rate
FPR	false positive rate
GRU	gated recurrent unit
LSTM	long short-term memory
MAE	mean absolute error
MDP	Markov decision process
ML	machine learning
NN	neural network
NNEF	neural network exchange format
NPV	negative predictive value
ONNX	open neural network exchange
PCA	principal component analysis
PHI	personal or protected health information
PII	personally identifiable information
PPV	positive predictive value

REST	representational state transfer
RNN	recurrent neural network
ROC	receiver operating characteristics
SGD	stochastic gradient descent
SVM	support vector machine
TNR	true negative rate
TPR	true positive rate

5 Overview

ISO/IEC 22989 defines ML as the process of optimising model parameters through computational techniques, such that the model's behaviour reflects the data or experience. Since the early 1940s, modelling of neurons (i.e. neural networks) and the development of computer programs that can learn from data have been explored. ML is an expanding field with the emergence of new applications in a wide array of industry sectors. This progression is enabled by the availability of large amounts of data and computation resources. ML methods include neural networks and deep learning.

In ISO/IEC 22989, an AI ecosystem is presented in terms of its functional layers and ML is a significant component of this AI ecosystem. Figure 1 illustrates the ML system which breaks down into the components of model, software tools and techniques and data.

Clause 6 in this document describes in further detail the different components of the ML system.

Clause 7 in this document describes different ML approaches and describes their dependency on training data.

Clause 8 in this document describes an ML pipeline: the processes involved in developing, deploying and operating an ML model.

Clauses 6.5 and 7 are more technical than the rest of the document. A stronger technical background can help the reader to better understand this content.

6 Machine learning system

6.1 Overview

Figure 1 depicts the elements of an ML system. They delineate the roles and their ML-specific functions that can be implemented by different entities (e.g. different vendors). The examples provided in Figure 1 are not meant to be an exhaustive list. Further explanation on each section of Figure 1 continues through Clause 6.

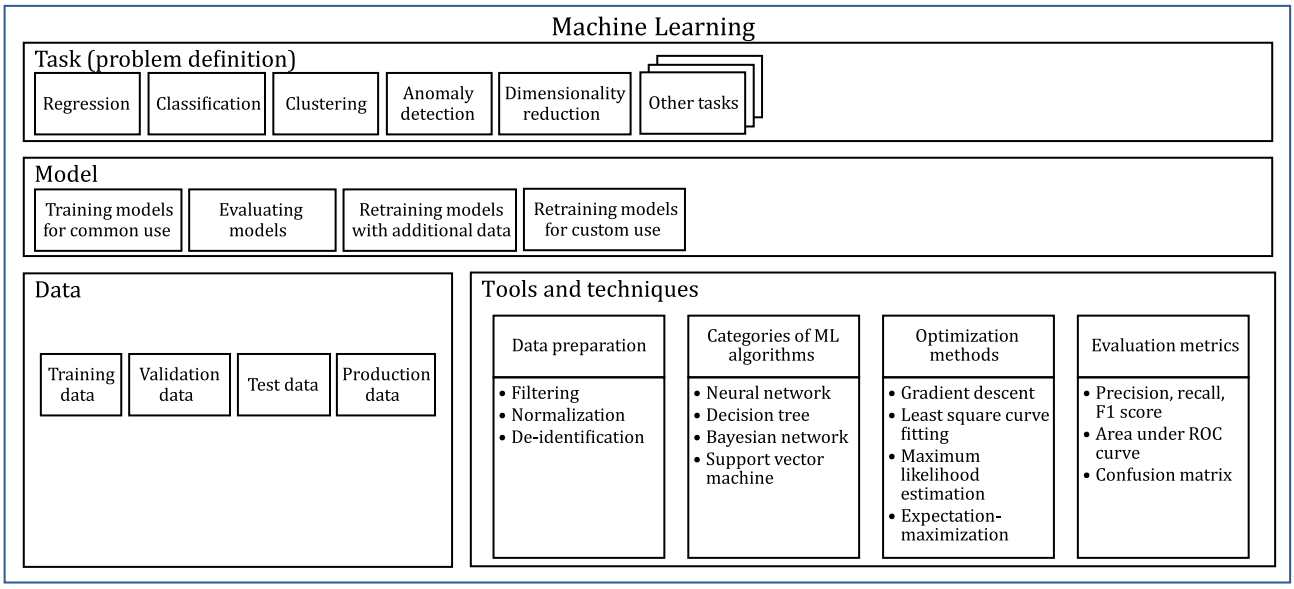

Figure 1 — Elements of an ML system

In Figure 1, the sub elements of model development and use can be considered as a layered approach, i.e. applications are built from models which are used to solve tasks. Model development and use in turn have a dependency on software tools and techniques and data.

A single ML system can be composed of several ML models used in combination. The system components can be described in terms of their input, output and their intent or function. The components can be tested independently.

ML models, when deployed, produce outputs such as predictions or decisions. A pre-trained model is an ML model already trained when it was obtained. In some cases, the developed model can be applied to a similar task, in a different domain. Transfer learning is a technique for modifying a pre-trained ML model to perform a different related task.

In this document, application refers both to the intended use of one or more ML models and to the concrete piece of software that implements that use. ML models are usually integrated with other software components to create applications. Applications using ML differ in the types of the input data they process and in the types of tasks they perform. In some applications, ML makes high-level predictions or decisions, while in other applications, ML provides answers to narrowly defined problems.

Differences in input data and tasks, as well as factors such as deployment options, accuracy and reliability, result in different application designs. AI applications can use proprietary custom designs or follow domain-specific design patterns.

Application logic is informed by the format of the input data, the output data, and potentially the transformation and the flow of data between the ML models in use. In all cases, the choice of ML algorithms and data preparation techniques is tailored to the application's tasks.

6.2 Task

6.2.1 General

The term "task" refers to actions required to achieve a specific goal. In ML, this implies identifying a problem to be solved using the ML model. One or more ML tasks can be defined for an ML application. Instead of solving a problem using a specific function represented as a set of steps and implemented in a software code, the defined problem is solved by applying a trained ML model to production data.

Effectively, the trained ML model implements a target function, which is an approximation of the hypothetical function that would have been written by a programmer to solve the problem.

An ML task setup involves defining the problem, the data format and the features.

The tasks described in the following subclauses are examples and are not exhaustive.

6.2.2 Regression

Regression tasks comprise predicting a continuous variable by learning a function that best fits a set of training data. In a regression task, the trained regression model represents a custom space. When the trained model is applied to a new production data instance, the instance is projected into the custom space defined by the trained regression model.

Regression is mainly used to predict numerical values of a real-world process based on previous measurements or observations from the same process. Use cases for regression include:

— predicting stock market price;

— predicting the age of a viewer of streaming videos;

— predicting the amount of prostate-specific antigen in the body based on different clinical measurements.

6.2.3 Classification

Classification tasks comprise predicting the assignment of an instance of input data to a defined category or class. Classification can be binary (i.e. true or false), multi-class (i.e. one of several possibilities) or multilabel (i.e. any number out of several possibilities). For example, classification can be used to predict whether an object in an image is a cat or a dog, or even from a completely different species. The classes are typically from a discrete and unordered set, such that the problem cannot be formalised as a regression task. For example, a medical diagnosis of a set of symptoms can be {stroke, drug overdose, seizure}, there is no order to the class values, and there is no continuous change from one class to another.

Use cases for classification include:

— Document classification and email spam filtering, where documents are grouped into several classes. A spam filter for instance uses two classes, namely “spam” and “not spam”;

— Classifying the species of a specimen. For example, an ML classification model can predict the species of a flower when provided with data that specifies the sepal length and width, and the petal length and width;

— Image classification. Given a set of images (e.g. of furniture), an ML system can be used to recognize and name the objects shown in those images.

6.2.4 Clustering

Clustering tasks comprise grouping input data instances. Unlike classification tasks, the classes are not predefined in clustering tasks but are determined as part of the clustering process. Clustering can be used as a data preparation step to identify homogenous data which can then be used as training data for supervised machine learning. Clustering can also be used to detect outliers or anomalies by identifying input data instances that are not like other samples. Example applications of clustering tasks include the sorting or organizing of files.

6.2.5 Anomaly detection

Anomaly detection comprise identifying input data instances that do not conform to an expected pattern. Anomaly detection can be useful for applications such as detecting fraud or unusual activities.

For anomaly detection, the ML model predicts whether an input data instance is typical for a given distribution.

6.2.6 Dimensionality reduction

Dimensionality reduction consists of reducing the number of attributes or dimensions per sample while retaining most of the useful information.

Dimensionality reduction can promote a dataset's most useful features and thereby mitigate computation costs.

Dimensionality reduction alleviates the various less-than-ideal effects of keeping too many features, collectively known as "the curse of dimensionality". Dimensionality reduction is also useful for data exploration and model analysis.

Methods for dimensionality reduction are unsupervised, supervised or semi-supervised[1].

6.2.7 Other tasks

There exist many other tasks which have different purposes and expected outputs. These tasks can be specific to a given application. Examples of other tasks include semantic segmentation of text or images, machine translation, speech recognition or synthesis, object localisation and image generation.

In planning, the task is to optimise a sequence of actions from an agent or agents through observing the environmental state.

Despite their diversity, a number of concepts have been formulated to draw connections between some of these other tasks. Structured prediction, corresponding to tasks in which the expected output of the model is a structured object as opposed to a single value, is one such concept.

Structured prediction requires computational methods that can account for regularities in the output, either by explicitly modelling them or by jointly predicting the whole structure with a model that internally models the regularities.

Use cases for structured prediction include:

— constructing a parse tree for a natural language sentence;

— translating a sentence in one language into a sentence in another language;

— predicting protein structure;

— semantic segmentation of an image.

6.3 Model

ISO/IEC 22989:2022, 3.2.11, defines an ML model as a mathematical construct that generates an inference or prediction, based on input data or information. The ML model comprises a data structure and software to process the structure, both determined by a chosen ML algorithm. The model is configured with inputs and outputs essential to solving the given problem.

The model is populated (also known as "trained") to represent the relevant statistical properties of the training data. Effectively, through the training process the model "learns" how to solve the problem for the training data with the goal to apply this acquired knowledge to a real-world application.

ML models produce results that are approximations of optimal solutions. ML algorithms utilise statistical optimisation methods to perform this approximation. The resultant mapping from the inputs to the outputs of the model reflects the patterns learned from the training data. Patterns can relate to correlations, causal relationships or categories of data objects. ML models are the result of the training data used. Thus, if the data used is incomplete, or reflects inherent societal bias, then the model performance will reflect this as well. Therefore, care should be taken with the datasets used for