

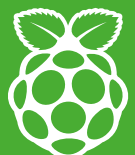
## Section 3: Computing topics

# Learning artificial intelligence at school with Scratch and LearningML

Juan David Rodríguez (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), Spain)

Rodríguez, J.D. (2021). Learning artificial intelligence at school with Scratch and LearningML. In Understanding computing education (Vol 1). Proceedings of the Raspberry Pi Foundation Research Seminar series.

Available at: [rpf.io/seminar-proceedings-2020](https://rpf.io/seminar-proceedings-2020)



Raspberry Pi

## Section 3: Computing topics

# Learning artificial intelligence at school with Scratch and LearningML

Juan David Rodríguez (Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), Spain)

### Introduction

Digital technologies are changing so quickly that people can become overwhelmed when dealing with new artifacts, both physical (hardware) and logical (software). However, in spite of this fast development, every new digital technology is based upon and can be explained by a set of well-established computer science (CS) principles.

For this reason, learning and teaching CS fundamentals to young people is crucial in order to help them become critical citizens able to understand and live comfortably in our increasingly digital society. Indeed, countries around the world have included in their educational curricula programming, robotic, and CS related content, intended to develop computational thinking (CT) skills (Wing, 2006) and achieve digital literacy.

Thanks to the creation of educational tools such as block-based programming platforms and the development of the CT concept, learning and teaching CS fundamentals is possible even at a very early age. A key feature of these tools is that they “engage and excite students in the first place” (Malan & Leitner, 2007, p. 6) and also allow students to solve problems that are important to them and their communities. This connection with young people’s ideas and interests is a powerful incentive to learn, providing an engaging and enjoyable way for

them to learn CS principles.

A top-down strategy is often taught when children are introduced to programming. First, the problem to be solved must be well understood; that is, it must be carefully analysed. Second, a set of rules able to solve the problem must be deduced, which is roughly speaking the algorithm. And finally, this set of rules must be turned into a computer program, by using a programming language to write the program code.

A broad range of problems can be solved by following this strategy. However, there are some kinds of problem which, while very easy to solve for a human, are very difficult to code as a computer program when a top-down strategy is followed. Recognising, in a set of pictures of cats and dogs, which are cats and which are dogs, is a clear example: any human can perform this recognition naturally, but obtaining by deduction a set of rules that enable you to build a computer program which solves this recognition problem is practically impossible.

Image recognition and classification, natural language understanding, sound recognition, and many other problems involving some kind of pattern extraction resist being solved using the traditional top-down strategy. Instead, a bottom-up strategy – inducing the rules that govern the problem from automatic data analysis – is followed when dealing with such tasks. And this way of attacking the problem takes us into the

field of artificial intelligence (AI) techniques.

Children are used to using computer applications capable of performing these kinds of tasks. They talk to their mobile phones to ask questions or search for information, use translation applications to translate into foreign languages, unlock their devices by showing their faces to the camera, and so on. Therefore, educational tools that allow children to design AI-based applications capable of dealing with these problems will help them to have a complete perspective about what can be done with a computer, and at the same time will allow students to solve problems that are of interest to them.

Furthermore, data is a prevalent concept in today's technological and economic development. Having an awareness of the central role that data plays in our lives and knowing how it is used to extract useful knowledge is a key factor in understanding digital society and, hence, achieving the digital literacy needed to understand the world we live in.

So, how are these problems solved? Can we teach children to solve them? At what age? These are questions we tried to ask with the development of LearningML, a tool intended to teach the fundamentals of machine learning (ML), the most widely used technique today for solving problems from data.

### Artificial intelligence and machine learning

ML is considered a subfield of AI, and the latter is a subfield of CS and almost as old as CS itself. In 1950, Alan Turing wrote a seminal paper entitled "Computing Machinery and Intelligence". Although the term AI does not appear in the paper, it is considered the birth of the field, since the big question it posed was: "Can machines think?" (p. 1)

A few years later, in 1956, McCarthy, Minsky,

Rochester, and Shannon led a workshop at Dartmouth College which aimed to gather a selected group of scientists to work "on the basis of conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it" (McCarthy, Minsky, Rochester, & Shannon, 2006, p. 1). Their proposal where the goal of the workshop was described was titled "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence", which was the first use of the term AI.

Since then, the field of AI has grown and its development has alternated between optimistic and pessimistic periods. After more than sixty years of research, many related subfields have emerged. Planning and problem solving, natural language processing, knowledge representation, expert systems, neural networks, machine learning, robotics and computer vision are some of the most successful and broadly used in current applications.

ML has existed as long as AI, although it has only been in recent years that it has flourished as the most successful subfield of AI. All ML algorithms need as much data as possible to produce useful outcomes. Data is the key. Therefore, greater computer power, together with the availability of very big storage systems, able to process and store large amounts of data, and fast and reliable network connections, have given rise to ML's recent success. The relevance of ML in AI today is so great that frequently when people say AI, they really mean ML, confusing the part with the whole.

So, what is ML in a nutshell? ML is the process of programming computers to optimise a performance criterion using example data or past experience. ML is used to create useful approximations for processes to solve tasks that we do not have algorithms for, but that we do have relevant data to learn patterns from

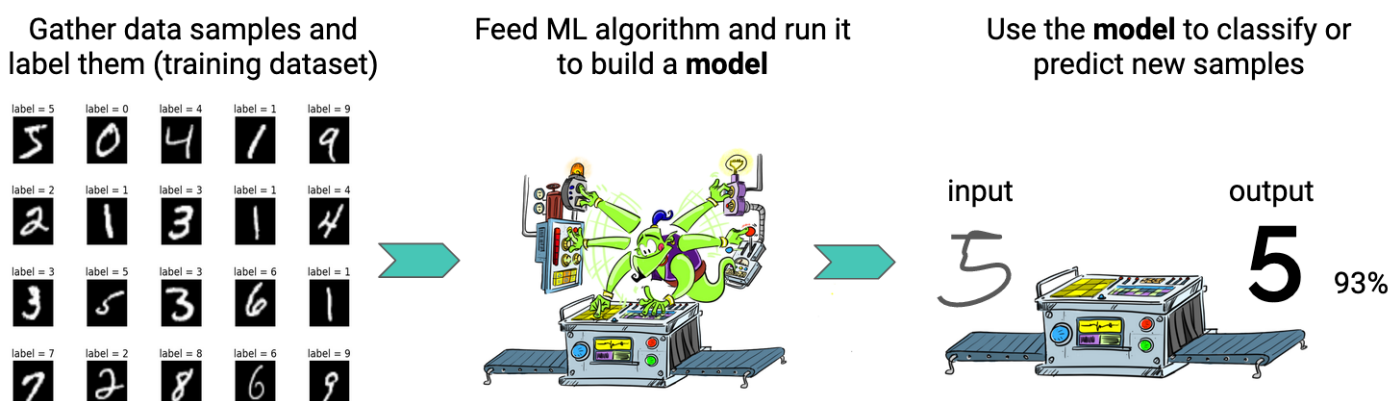


Figure 1. Supervised machine learning steps to build a ML model

(Alpaydin, 2020). The term learning is a metaphor which expresses the fact that the more data is fed in as an input to the algorithms, the more accurate their outcomes are.

Each of the numerous algorithms that make up the ML family belong to one of the following categories: supervised learning, unsupervised learning, and reinforcement learning. In all cases, the goal of ML is to build a model capable of classifying, predicting, or recognising things from a collection of data. In supervised learning, a dataset of known examples is manually labeled to build a model with which unlabeled data, similar to but different from those used in the training data set, can be recognised. In unsupervised learning, the built models are able to extract some patterns from a set of unlabeled data. Finally, in reinforcement learning, models are built by testing possible solutions; those that maximise some reward function are maintained while those that score badly according to that function are eliminated.

As an example, Figure 1 shows the steps needed to build a model aimed at recognising handwritten numbers by means of supervised

ML. The model, represented by a machine, is being built by a ML algorithm, represented by a genie, which analyses the training dataset to iteratively improve the model. When the ML algorithm is finished, an independent model capable of recognising new handwritten numbers is ready to be used in a software application.

### Artificial intelligence education in K12

AI education in K12 is not new. The first efforts to make AI programming tools accessible to children took place in the early 1970s, with the Logo programming language (Solomon et al., 2020), and continued through the 1980s. However, AI education suffered a cold period from the 1990s until 2012, when educators, AI researchers, and the general public changed their view about AI due to the big success achieved by ML in solving problems such as image recognition, language translation, transcription of speech, game playing, and natural language processing (Kahn & Winters, 2020).

This vigorous rebirth of AI education becomes pertinent in Moreno-Guerrero et al. (2020), where

a scientific mapping of 379 publications, dated from the birth of AI in 1956 to the present was carried out to analyse the importance and the high profile that AI has acquired in the scientific literature in the Web of Science categories related to the field of education.

Why this growing interest in teaching AI at school? AI has erupted in society, creating new applications and possibilities while also introducing some ethical problems. Whether they are conscious of it or not, children use software applications based on AI on a daily basis: product recommendation systems, predictive writing, face recognition, and many more. However, few people understand how these technologies work, yet as argued earlier, this is a must if we want to educate conscientious and critical citizens of the future.

Therefore, governments around the world, worried about the benefits and risks AI poses, are developing policies, strategic plans, and other initiatives around this subject. Some policy foresight reports suggest that in the coming years AI will change learning, teaching, and education. The speed of technological change will be very fast, and it will create pressure to transform educational practices, institutions, and policies (Pedro, Subosa, Rivas, & Valverde, 2019; Tuomi, 2018). These reports also suggest that AI-related jobs are growing dramatically and hence, there is a rising demand for AI-literate workers.

Although programming and CT content has been incorporated into primary and secondary curricula in most developed countries, AI content is often not included, or is treated very superficially. CT is a cognitive ability while programming is an instrumental competence (Moreno-León, Robles, Román-Gonzalez, & Rodríguez-García, 2019). Indeed, programming, together with unplugged activities, is the activity widely used to develop CT. We propose that

hands-on AI projects can also contribute to CT development (Rodríguez-García, León, González, & Robles, 2019). In fact, AI could add some new dimensions to the existing CT framework, as proposed in Van Brummelen, Shen, & Patton, (2019).

For instance, when gathering and labelling a dataset intended to build a model by means of supervised ML, students should strive to find a representative set of examples from which a good model is obtained when the ML algorithm is applied. This activity helps them to get a deeper insight into the problem being solved. For example, when a text recognition model on a given topic is to be built, a set of sentences with the necessary vocabulary for that topic must be chosen to build a sufficiently precise ML model. Furthermore, a classification task and an evaluation of the model has to be performed and, if the resulting model does not perform well enough, the training dataset must be improved by adding or removing some data. Therefore, the solution is found through iteration, which is a way to gain a progressive understanding of the problem that is being solved.

### The LearningML platform

LearningML is an educational platform intended to teach ML fundamentals in an easy and enjoyable way (Rodríguez-García, Moreno-León, Román-González & Robles, 2020). It has been developed taking “low floor, high ceiling and wide walls” as the main design principle (Resnick et al, 2009, p. 63). That is, we have tried to build a tool which is very easy to get started with and allows users to get some results from the very beginning (low floor), but also allows students to build more complex projects over time (high ceiling), and is able to support different kinds of projects (wide walls).

Among these principles, the first — developing a tool that is very easy to use and get started with — was the most relevant for us. Hence,

only a standard web browser is needed to run LearningML. In addition, the use of any cloud AI service (such as Google AI or IBM Watson) has been avoided, since they require users to create an account and deal with API keys, which can be a very easy task for a software developer, but can be an obstacle for children and teachers. In addition, although there are free usage plans available for these cloud services, they are exposed to possible changes in their terms and conditions and that availability may change in the future. Therefore, all the complex ML algorithms have been built into the code of LearningML to run locally in the web browser. There is no need to register in order to start building ML models and coding applications, although some extra functionality, such as the option to save projects in the cloud and to share or reinvent shared works, can be accessed when you create a LearningML account.

Although our research has shown (Rodríguez-García, Moreno-León, Román-González & Robles, 2021) that the tool is very suitable for children between 10 and 16 years old, LearningML can also be helpful for undergraduate students and professionals who need to learn ML fundamentals because of the expansion of ML-based tools in their fields.

LearningML is composed of three elements: the website, the ML editor, and the programming editor.

### The website

The website<sup>8</sup> is devoted to hosting the platform and offering content designed to help users learn how to use the tool, as well as learn about ML and AI. Guided activities, video tutorials, a manual, a curated list of resources about ML/AI, and a blog with ML/AI related news can be found on the website.

<sup>8</sup> <https://learningml.org>

<sup>9</sup> <https://learningml.org/editor>

<sup>10</sup> <https://www.gnu.org/licenses/agpl-3.0.en.html>

### The ML editor

The ML editor<sup>9</sup> is the tool where the user can build ML models for image or text recognition. This tool demonstrates clearly how supervised ML works. The main screen is divided into three sections, one for each phase of supervised ML.

In the first section the user creates some buckets corresponding to the different classes of data that have to be recognised. Then a set of example images or text must be added to these buckets depending on the class they belong to. This process, where the user gathers and labels a dataset, is known as *training*.

Once the dataset has enough samples, the learning phase can be run. Since it is an educational application, a few examples in each bucket (>10) are sufficient to obtain a working model. In the learning phase a simple neural network is used as the ML algorithm for text recognition, while a pre-trained neural network, known as mobilenet (Howard et al., 2017), is used together with a simple neural network as ML algorithms for image recognition.

Finally, when the learning phase has finished, a ML model, able to recognise new text or images similar but different to those used in the training dataset, is available for evaluation. In the evaluation phase we can test if the model works and is able to recognise almost all the test data. If the model does not perform well enough, more data examples can be added to the training dataset and a new model can be built by running the learning phase once again. This iterative process can be repeated until a good model is obtained.

This software is released under the *GNU Affero General Public License*<sup>10</sup>, a free software license allowing anyone to study, modify, or contribute to the project. An instance of the application is also accessible at no cost.

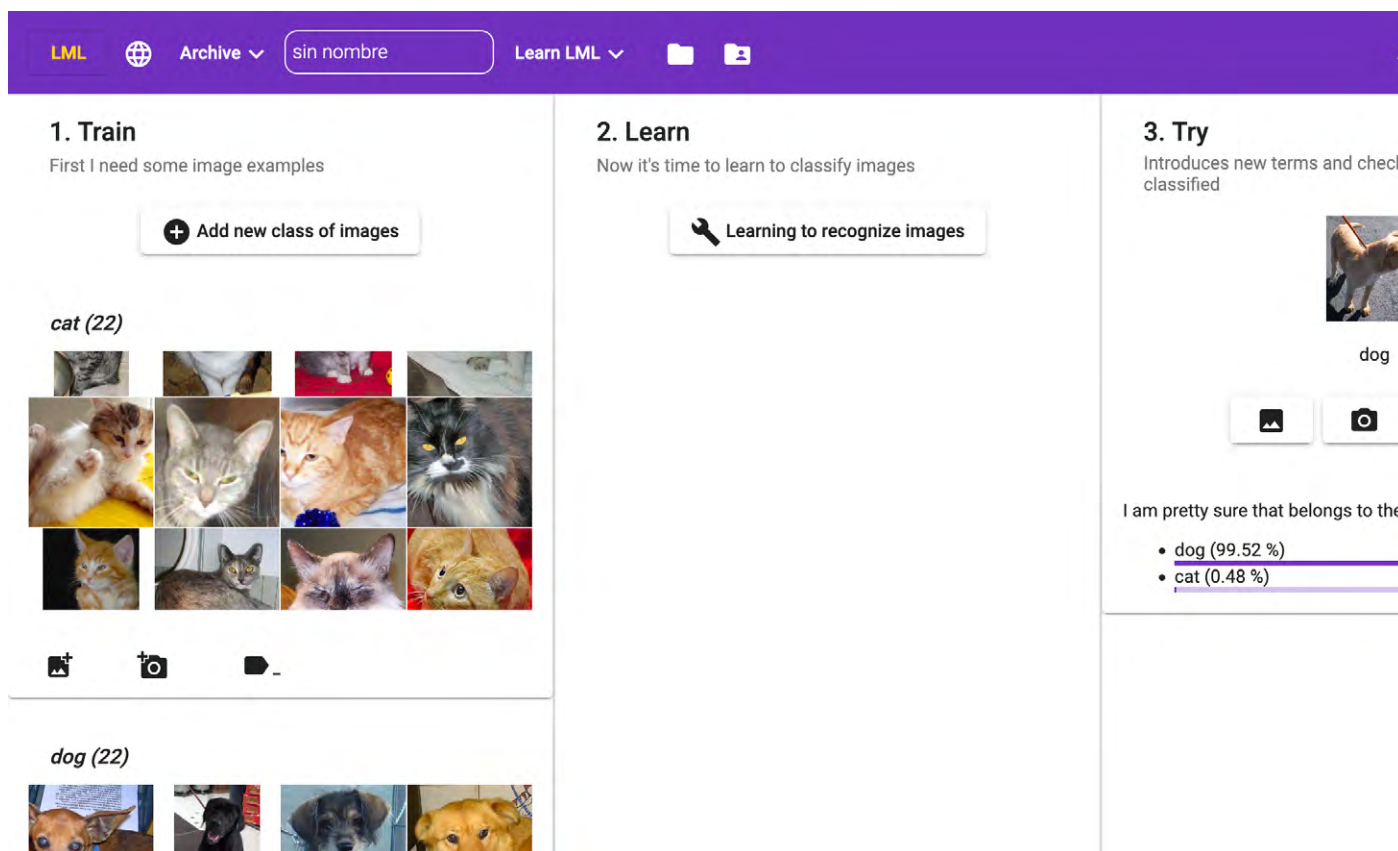


Figure 2. LearningML. The ML editor.

## The programming editor

The programming editor<sup>11</sup> is a Scratch (Resnick et al., 2009) modification in which new blocks have been created which use the model built with the ML editor. Therefore, text and image recognition features can be added to Scratch programs. To achieve this, the following new blocks are available: a reporter able to classify text/images, a reporter that returns the confidence level of the classification given by the ML model, a stack block intended to add new image or text examples to the training dataset, and a stack block which allows the user to run the learning phase in order to build a new model.

We have been able to develop our ML extension

of Scratch thanks to the free *Apache license 2.0*<sup>12</sup> under which this software is released, since it allows modification of the code under the conditions imposed by the license. Our ML Scratch fork has also been released under the same license.

## Conclusion

LearningML is an educational platform aimed at teaching and learning AI fundamentals by doing. It is being developed to be as user-friendly as possible but, at the same time, to allow the creation of a wide range of AI-based applications, from the simplest to the most complete and complex applications. We aim to provide teachers and students with a

<sup>11</sup> <https://learningml.org/scratch>

<sup>12</sup> <https://www.apache.org/licenses/LICENSE-2.0>

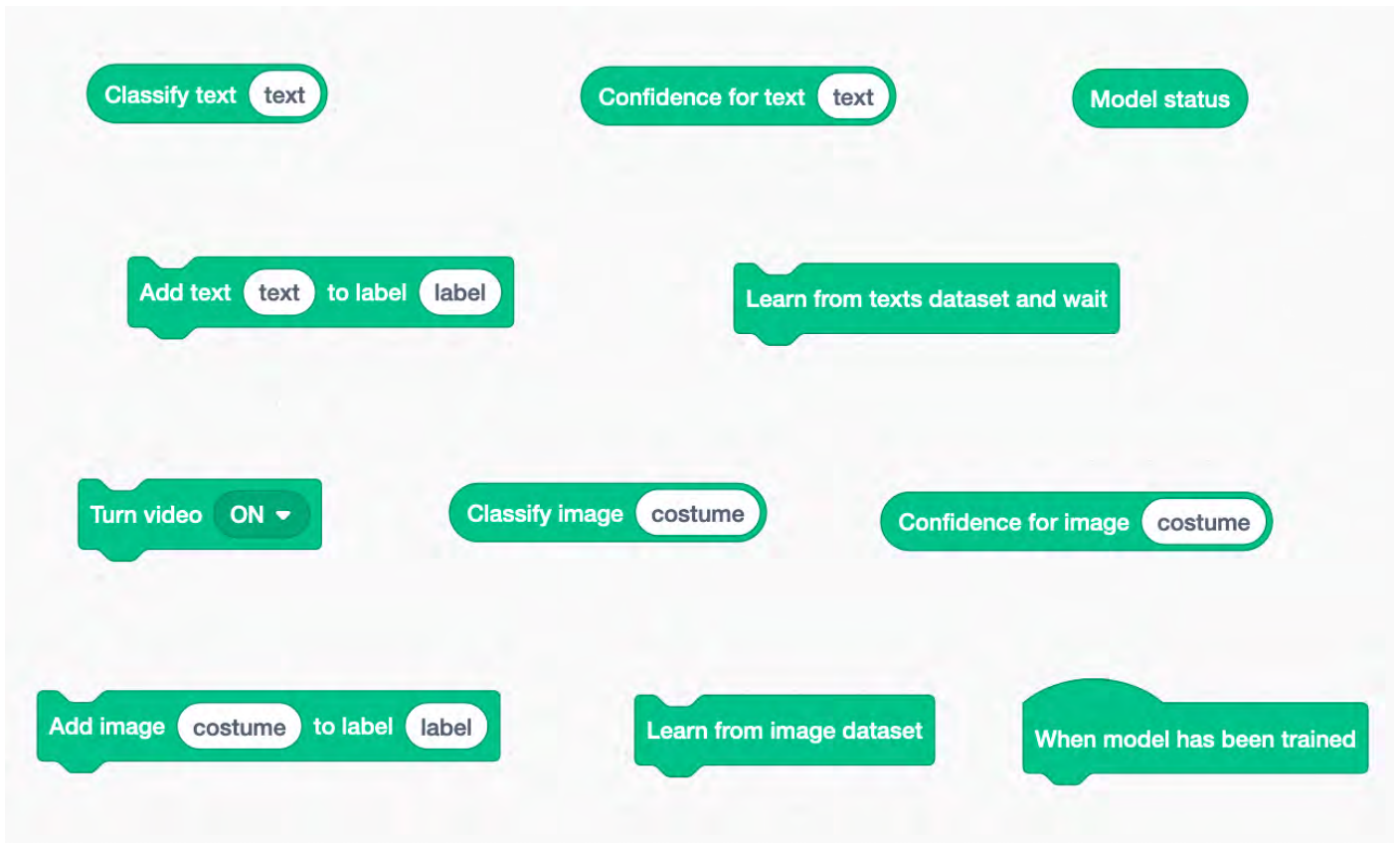


Figure 3. LearningML. ML Blocks added to Scratch in the programming editor.

powerful and engaging tool that helps them to develop CT skills by combining traditional programming activities with ML model building, to encourage the development of some new concepts, practices, and perspectives such as classification, training, and evaluating.



---

## References

- Alpaydin, E. (2020). *Introduction to machine learning*. MIT Press.
- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada* (Vol. 1, p. 25).
- Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andretto, M., Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.
- Kahn, K., & Winters, N. (2020). Constructionism and AI: A history and possible futures. *Constructionism 2020*, pp 238-243. (Available at: <http://www.constructionismconf.org/wp-content/uploads/2020/05/C2020-Proceedings.pdf>)
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227.
- McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (2006). A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence, August 31, 1955. *AI magazine*, 27(4), 12.
- Moreno-Guerrero, A. J., López-Belmonte, J., Marín-Marín, J. A., & Soler-Costa, R. (2020). Scientific Development of Educational Artificial Intelligence in Web of Science. *Future Internet*, 12(8), 124.
- Moreno-León, J., Robles, G., Román-González, M., & Rodríguez-García, J. D. (2019). No es lo mismo: Un análisis de red de texto sobre definiciones de pensamiento computacional para estudiar su relación con la programación informática: Not the same: a text network analysis on computational thinking definitions to study its relationship with computer programming. *Revista Interuniversitaria de Investigación En Tecnología Educativa*, N° 7. <https://doi.org/10.6018/riite.397151>
- Pedro, F., Subosa, M., Rivas, A., & Valverde, P. (2019). Artificial intelligence in education: Challenges and opportunities for sustainable development.
- Resnick, M., Silverman, B., Kafai, Y., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., & Silver, J. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60. <https://doi.org/10.1145/1592761.1592779>
- Rodríguez-García, J. D., León, J. M., González, M. R., & Robles, G. (2019, November). Developing computational thinking at school with machine learning: an exploration. In *2019 International Symposium on Computers in Education (SIIE)* (pp. 1-6). IEEE.
- Rodríguez-García, J. D., Moreno-León, J., Román-González, M., & Robles, G. (2020). LearningML: A Tool to Foster Computational Thinking Skills Through Practical Artificial Intelligence Projects. *Revista de Educación a Distancia*, 20(63).
- Rodríguez-García, J. D., Moreno-León, J., Román-González, M., & Robles, G. (2021). Evaluation of an Online Intervention to Teach Artificial Intelligence With LearningML to 10-16-Years-Old Students. In SIGCSE '21 ACM SIGCSE Technical Symposium, March 17–20, 2021, Toronto, Canada. ACM, New York, NY, USA. Preprint [https://www.researchgate.net/publication/344744220\\_Evaluation\\_of\\_an\\_Online\\_Intervention\\_to\\_Teach\\_Artificial\\_Intelligence\\_With\\_LearningML\\_to\\_10-16-Year-Old\\_Students](https://www.researchgate.net/publication/344744220_Evaluation_of_an_Online_Intervention_to_Teach_Artificial_Intelligence_With_LearningML_to_10-16-Year-Old_Students)
- Solomon, C., Harvey, B., Kahn, K., Lieberman, H., Miller, M. L., Minsky, M., Papert, A. & Silverman, B. (2020). History of Logo. *Proceedings of the ACM on Programming Languages*, 4(HOPL), 1-66.
- Tuomi, I. (2018). The impact of artificial intelligence on learning, teaching, and education. *Luxembourg: Publications Office of the European Union*.
- Turing, A. M. (2004). Computing machinery and intelligence (1950). *The Essential Turing: The Ideas that Gave Birth to the Computer Age*. Ed. B. Jack Copeland. Oxford: Oxford UP, 433-64.
- Van Brummelen, J., Shen, J. H., & Patton, E. W. (2019, June). The Popstar, the Poet, and the Grinch: Relating Artificial Intelligence to the Computational Thinking Framework with Block-based Coding. In *Proceedings of International Conference on Computational Thinking Education* (pp. 160-161).
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.





**Raspberry Pi**

[www.raspberrypi.org](http://www.raspberrypi.org)

 [@raspberrypi](https://www.facebook.com/raspberrypi)

 [@raspberrypi](https://www.instagram.com/raspberrypi)

 [@Raspberrypi](https://twitter.com/Raspberrypi)

 [raspberrypi](https://www.youtube.com/raspberrypi)