MachineLearnAthon Tool List

This document provides an overview of tools recommended for MachineLearnAthon participants. The selection supports different levels of technical experience and learning goals.

For participants with **no programming experience** or those who prefer not to learn coding, we recommend using a **no-code environment** such as *Orange Data Mining*. It allows users to build machine learning workflows visually by connecting components without writing code—ideal for beginners who want to explore ML concepts intuitively.

For participants who wish to **learn basic programming skills** and possibly extend them later, we recommend using **Python**, one of the most popular languages in data science and machine learning. Python is known for its

- simple and readable syntax,
- extensive open-source library ecosystem (e.g., *scikit-learn*, *pandas*, *numpy*),
- strong community support and abundant educational resources,
- cross-platform compatibility, and
- widespread use across academia and industry—making it highly valuable for future career development.

There are several environments where Python can be run effectively: interactive notebooks such as **Jupyter Notebook**, cloud-based platforms like **Google Colab**, or integrated development environments such as **Visual Studio Code**. Each offers unique advantages depending on user preferences regarding setup effort, flexibility, and project complexity.

For more advanced learners interested in professional workflow management or reproducible research setups, we additionally introduce **Docker** as an optional tool. Docker enables containerized execution of ML workflows—for example running Jupyter Notebooks within isolated environments—and helps students understand modern deployment practices used in industry.

The following table introduces the purpose, advantages, and limitations of each tool presented in the MachineLearnAthon.

Category	Tool / Environment	Purpose / Use Case	Advantages (Pros)	Limitations (Cons)
No-Code ML Environment	**Orange Data Mining**	Enables visual creation of ML pipelines through drag-and-drop components; suitable for classification, regression, clustering tasks.	Intuitive interface; no programming required; ideal for beginners or non-technical disciplines; immediate visualization of workflows and results.	Limited flexibility for advanced customization; fewer libraries compared to Python ecosystem.

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2022-1-DE01-KA220-HED-000086932.

Co-funded by the Erasmus+ programme of the European Union

Programming Language & Core Ecosystem	**Python**	Serves as common foundation across tutorials for data preprocessing, model training, evaluation, and visualization.	Readable syntax; extensive open-source libraries covering all ML tasks; strong community support; widely used in academia and industry—ideal for employability.	Requires basic coding knowledge; setup may vary depending on environment configuration.
Interactive Notebook Environment	**Jupyter Notebook**	Local interactive notebooks supporting code execution alongside documentation and visualization. Ideal for exploratory analysis and step-by-step learning.	Combines text, code, figures in one document; supports reproducible workflows; highly flexible for experimentation.	Requires local installation unless containerized or cloud-hosted; performance depends on system setup.
Cloud-Based Notebook Environment	**Google Colab**	Browser-based notebook platform enabling Python execution without local installation or configuration effort. Used frequently for team collaboration or quick prototyping of challenges.	Free access with GPU/TPU options; easy sharing via Google Drive links; no need for software installation—students can start immediately from any device.	Internet connection required; limited persistent storage space compared to local setups.
Integrated Development Environment (IDE)	**Visual Studio Code (VS Code)**	Supports structured development of larger ML projects beyond single notebooks—especially useful	Modular design supporting multiple languages and version control systems (Git); clear project organization structure aiding	Slightly steeper learning curve than notebook interfaces; requires manual environment configuration initially.

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2022-1-DE01-KA220-HED-000086932.

Co-funded by the Erasmus+ programme of the European Union

		for more advanced students or professional applications.	scalability of codebases.	
Container-Based Environment	**Docker** (optional)	Demonstrates reproducible execution environments by containerizing ML workflows such as Jupyter Notebooks or API deployment examples in challenges focused on production readiness.	Ensures portability across systems ("runs everywhere"); introduces students to real-world DevOps practices relevant in industry contexts; supports dependency isolation between projects.	Primarily suited for advanced learners interested in workflow automation—requires additional setup effort not necessary for introductory topics.

The creation of these resources has been (partially) funded by the ERASMUS+ grant program of the European Union under grant no. 2022-1-DE01-KA220-HED-000086932.

Co-funded by the Erasmus+ programme of the European Union