Handbook

Erasmus+ Project MachineLearnAthon

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 European Union

Machine LearnAthon

Content

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union

1 Learning Logic and Pedagogical Rationale

1.1 Purpose and underlying problem the concept addresses

Machine learning (ML) has become a pervasive method across disciplines, including business, engineering, and the social sciences. While demand for ML skills rises, effective teaching remains difficult because ML learning is simultaneously (a) conceptually diverse (many paradigms and model families), (b) implementation-heavy (software tooling and coding practices are unavoidable), and (c) empirical and contextual (data selection, evaluation, and

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.

deployment decisions shape whether an ML solution is useful, safe, or misleading). In addition, ML outcomes are inherently sensitive to data issues such as bias and selection effects, which makes "responsible application" a core learning requirement rather than an optional add-on.

Traditional university ML courses often assume a relatively homogeneous student group with solid mathematical preparation and prior programming experience. In many target cohorts of an Erasmus+ higher-education consortium, the reality is the opposite: students come from multiple disciplines, and their prior exposure to mathematics, statistics, programming, and domain knowledge varies widely. The didactic concept described here is explicitly designed for this heterogeneity: it provides an inclusive learning pathway that does not require students to become ML engineers but does enable them to apply ML reliably and responsibly to real-world problems, and to collaborate effectively in interdisciplinary teams.

A systematic literature review (as described in the project paper) reinforces the need for structured didactic guidance: from an initially broad set of publications, only a small subset directly focuses on pedagogical strategies for teaching ML methods, highlighting a gap in established, evidence-based teaching formats for ML in higher education.

1.2 What is being taught: competence orientation and learning goals

The didactic concept follows a competence-oriented perspective: the goal is not merely that students can reproduce definitions or implement a single model, but that they can act in an ML project context understand the problem, select an approach, implement a solution, evaluate results critically, and reflect on limitations and risks.

The concept's overall objective is to teach ML to students with little or no prior programming knowledge and to make ML education accessible across disciplines. In operational terms, the course pursues six interlinked sub-goals:

1. **Data literacy improvement**

   Students learn to explore, interpret, and prepare data in ways that support sound ML modelling decisions. Data literacy includes understanding data quality, missingness, feature meaning, and the implications of preprocessing choices.

2. **Basic understanding of ML paradigms and widespread models**

   Students learn the basic categories of ML (e.g., supervised and unsupervised learning) and a curated set of widely used models. This conceptual basis is necessary so students can frame a real task into an appropriate ML paradigm and communicate their approach to others.

3. **Skill to employ ML models using No code solutions or Python**

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.

Students learn to operationalize ML using practical tools. The focus is not on "coding for its own sake," but on enabling students to implement, run, modify, and document ML workflows. The paper emphasizes the role of widely used Python tooling and the need to cover the ML pipeline end-to-end.

4. **Awareness of risks and limitations associated with ML**

   Students learn to interpret results critically, understand that performance metrics can mislead, recognize sources of bias and overfitting, and communicate uncertainty and limitations. This is essential for responsible ML application.

5. **Fostering cooperation in working groups (interdisciplinary and international)**

   Because real ML projects are rarely solo work, the concept emphasizes teamwork, including the coordination of different strengths (domain expertise vs. methodological expertise).

6. **Encouraging application-oriented thinking**

   Students learn to treat ML as a practical method serving a real problem, with constraints and trade-offs, rather than a purely theoretical exercise.

These learning goals translate into a fundamental design decision: ML should be taught as an applied, action-driven competence rather than as a sequence of disconnected theory topics. The primary educational question becomes: *How do we organize learning so that novices can meaningfully act in an ML problem-solving situation and gradually build capability?*

1.3 The central learning object: challenges as the organizing core

The didactic concept defines the **ML challenge** as the central learning object. A challenge is a structured, authentic (or authenticity-oriented) ML task derived from a real-world problem and paired with data and evaluation criteria. This differs from a conventional course where lectures define the syllabus and exercises follow afterward. Here, the challenge is the anchor; lectures and tutorials are derived from what students need to solve the challenge.

This choice is grounded in two core observations:

- **ML is best learned by doing, not by passive reception.** ML work requires decisions across data, modelling, and evaluation. These decisions cannot be fully understood without practice. The concept therefore emphasizes "hands-on" learning using challenges based on real-world problems and data.
- **Real-world challenges motivate learners and enable interdisciplinary collaboration.** When the problem is meaningful (and when it resembles industrial tasks), students experience relevance and urgency. This motivates effort and supports teamwork because domain knowledge becomes valuable, not peripheral. The paper highlights industrial problems and gamification elements as motivation drivers.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

In practice, defining the challenge as the core learning object also directly supports your proposal commitments: challenges can be designed in multiple difficulty levels, can be preceded by micro-courses, and can incorporate tool-introduction units. This allows you to differentiate learning pathways while keeping a stable conceptual frame across partners and institutions.

1.4 The pedagogical foundation: action orientation, constructivism, and problem orientation

The concept is explicitly grounded in three complementary didactic principles, each addressing a crucial requirement of ML teaching for heterogeneous cohorts.

1.4.1 Action orientation: learning as a complete action process

Action orientation frames learning as a process that mirrors professional practice. Rather than treating knowledge as isolated subject blocks, action-oriented learning organizes content around practical action structures and authentic tasks. The learning process should correspond to a complete action process, typically including clarifying goals, planning, realizing (implementation), presenting, and evaluating. These stages correspond closely to the lifecycle of an ML project, from problem framing through solution and validation to communication and reflection.

This principle is central because it transforms "learning ML" from memorizing algorithms into acquiring the ability to act competently in ML situations. For novices, action orientation also improves motivation: tasks are concrete and outcomes visible. In your proposal language, this connects directly with project-based learning and hackathon-style challenge formats.

1.4.2 Constructivism: active knowledge construction supported by tools

Constructivism emphasizes that learners build knowledge actively, integrating new information with prior experiences. In the ML context, this is crucial because students enter with different priors: some understand business domains well but lack programming; others may code but lack domain grounding; others may know statistics but not ML workflows.

Constructivist teaching therefore avoids a one-way transmission model. Instead, it provides "tools" and learning environments that allow students to make decisions, test approaches, and learn through guided activity. The course structure accordingly uses micro-lectures and tutorials that enable self-directed, active engagement students do not just hear about ML; they manipulate code, inspect data, observe consequences, and build practical understanding.

This is also why the micro-lecture format matters: micro-lectures can be arranged flexibly, allowing educators to adapt content to the specific challenge and the cohort's needs.

1.4.3 Problem orientation: authentic problems as the driver of learning

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Problem-oriented learning focuses on dealing with authentic problems. Students are given a task first, then they analyse, research, develop solutions, present results, and reflect on both the solution and the process. This aligns naturally with ML work, where the "problem" is rarely a neat textbook exercise and where success depends on iterative improvement and evaluation.

Problem orientation therefore complements both action orientation and constructivism: it provides the authenticity and complexity that makes active knowledge construction meaningful and ensures the learning process develops real competence in handling complex tasks.

1.5 Core didactic assumptions guiding design decisions

Building on the literature review and the three didactic principles, the concept explicitly states several assumptions that determine how teaching units are designed and assembled:

1. **ML is best taught hands-on using challenges based on real-world problems and data.**
2. **ML can be operationalized by students with low-to-intermediate methodological expertise, given tailored content selection.**
3. **Tailoring of theoretical methods and practical tools is best achieved via micro-lectures.**
4. **Interdisciplinary collaboration should be encouraged to combine methodological and domain expertise.**

These assumptions are not generic slogans; they have direct structural implications. For example, if ML is best taught via challenges, then a challenge catalogue becomes the primary curricular backbone. If tailoring is essential, then difficulty levels and modular content become mandatory, not optional.

1.6 Scope of content: what is covered and why it is curated

The concept proposes covering the most widespread ML categories and exemplary models across paradigms. This is a deliberate compromise between breadth and feasibility for novices: students should gain a conceptual map of ML and learn a representative subset that enables them to solve many practical problems.

At minimum, the proposed content scope includes:

- **Supervised learning:** classification and regression
- **Unsupervised learning:** clustering

The paper also indicates that such coverage sets the stage for more advanced directions such as semi-supervised learning, reinforcement learning, and AutoML, and explicitly references AutoML frameworks (e.g., AutoSklearn, TPOT) as empowering tools for non-experts when taught appropriately.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

In didactic terms, this content scope is not a traditional "syllabus list." Instead, it is a toolbox from which educators select what is needed for specific challenges, aligned with difficulty levels and student readiness.

1.7 How students learn: the micro-lecture + tutorial + challenge triad

1.7.1 Micro-lectures: just-in-time, modular concept delivery

Micro-lectures are short, reusable learning units that introduce challenge-relevant concepts at a manageable granularity. Their role is twofold:

- Provide foundational understanding of ML concepts, algorithms, and risks.
- Enable flexible re-organization of content for different challenges without redesigning the full course each time.

The concept explicitly argues that well-separated micro-lectures enable cohesive recombination of material and allow educators to adapt to an audience's skill level by varying task difficulty and adding tool introduction units where needed.

1.7.2 Tutorials: guided, hands-on implementation practice

Tutorials operationalize micro-lecture content. They include well-documented example code and tasks focused on code modification and incremental construction. Their goal is not only that students "run code," but that they understand how to:

- load and inspect data,
- implement baseline models,
- improve solutions iteratively,
- evaluate properly, and
- document and communicate results.

This design is constructivist: learners acquire knowledge by actively working with the tools and material, rather than passively receiving information.

1.7.3 Challenges: competence demonstration through authentic tasks

Challenges serve as the integrative mechanism that brings the ML pipeline together. A challenge includes the dataset, task definition, evaluation metric, and often a leaderboard mechanism (Kaggle-style). Within a challenge, students must unify what they learned:

- apply ML methods appropriately,
- use tools effectively,
- interpret results critically,
- collaborate in teams,
- and reflect on risks and limitations.

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.

A key didactic advantage is that challenges allow structured differentiation: Teacher can define multiple difficulty levels and tailor content requirements accordingly. This is how our concept differentiates from "standard Kaggle" challenges. This also supports inclusive participation and enables re-use across cohorts and universities.

1.8 Differentiation and inclusion: difficulty levels as a core mechanism

A defining feature of the concept is differentiating between several difficulty levels. The purpose is not to create elitism; it is to make the learning environment inclusive while still allowing advanced learners to stretch.

Difficulty levels can vary through:

- the degree of preprocessing provided (clean vs. messy data),
- the complexity of feature engineering required,
- the strictness of evaluation protocols (simple split vs. cross-validation),
- the openness of modelling choices (single baseline vs. multiple families),
- the extent of risk/ethics reflection required,
- or constraints like interpretability requirements.

The micro-lecture format is crucial here: by combining challenges with modular micro-lectures and tool-introduction units, educators can adapt difficulty without redesigning the entire curriculum.

1.9 Social structure of learning: teams, interdisciplinarity, and international collaboration

ML practice is inherently collaborative. The concept therefore treats teamwork as a learning goal and a learning method. Students works alone or in the small groups, reflecting recommendations for action-based learning projects.
This group format has several didactic functions:

- It mirrors real ML projects where different expertise is needed (domain + methodology).
- It supports peer learning: students teach each other, which is particularly valuable in heterogeneous cohorts.
- It strengthens communication competence students must explain methods and justify decisions.
- It supports motivation and persistence, because teams create accountability and social reinforcement.

International and interdisciplinary collaboration is a natural extension: distributed groups, joint leaderboards, and cross-institution feedback cycles transform the course into a collaborative event format resembling hackathon, while retaining a structured academic framework.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

1.10 Motivation and engagement: gamification with purpose

The proposed approach highlights that gamification elements (particularly leaderboards) combined with real-world industrial problems significantly enhance student motivation and learning engagement. Within this didactic concept, gamification is not applied as superficial entertainment or competition, but as a structured pedagogical tool that supports learning processes, self-regulation, and reflective practice.

- **Immediate feedback:** Leaderboards and performance indicators provide students with continuous and transparent feedback on their progress over time. This immediate feedback loop allows learners to quickly understand the effects of their decisions, modelling choices, or analytical strategies, thereby reinforcing learning through timely reflection and adjustment.
- **Goal clarity:** Clearly defined metrics, milestones, and ranking criteria help students better understand what constitutes successful task completion. When carefully framed, rankings act as indicators of progress rather than measures of personal ability, supporting mastery-oriented learning goals while minimizing the risk of unhealthy or demotivating competition.
- **Sustained engagement:** Gamified elements encourage repeated interaction with learning tasks. Students are motivated to refine their solutions, test alternative approaches, and improve their results across multiple iterations. This supports the development of persistence, analytical thinking, and problem-solving skills, which are essential in data-driven and practice-oriented disciplines.
- **Social comparison and peer learning:** Leaderboards and team-based challenges enable constructive social comparison, fostering peer learning through shared discussions, collaborative problem-solving, and the presentation of diverse solution strategies. Within clearly defined ethical boundaries and rules, students learn not only from their own experimentation but also from observing and discussing the approaches of others, which strengthens collective knowledge-building and communication skills.

Because ML education can otherwise feel abstract, gamification helps maintain momentum particularly in cohorts that are initially anxious about coding or mathematics.

1.11 Delivery mode and accessibility: online materials, blended execution

The concept states that micro-lectures will be provided online and that online learning proved effective and suited for teaching ML in the reviewed literature.

This supports flexible implementation across partner universities:

- fully online - suitable for lifelong learning, professional training, or geographically dispersed learners, particularly beneficial for students balancing studies with family

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.

responsibilities, as micro-lectures can be accessed at any convenient time, allowing for greater flexibility in managing personal and academic commitments;
- blended - combining online micro-lectures with on-site seminars, workshops, and guided practical exercises;
- or on-site with online preparation, where digital materials support.

It also supports scalability and openness: if materials are designed as reusable micro-units, other institutions can adopt the concept with low overhead.

1.12 Course organization: exemplary timeline and action cycle integration

The concept proposes a course organization consisting of two parts:

1. **Part 1:** Learning ML basics and Python implementation skills through micro-lectures and tutorials.
2. **Part 2:** Hands-on application in which students work on real-world problems and real-world datasets, followed by presentations and feedback.

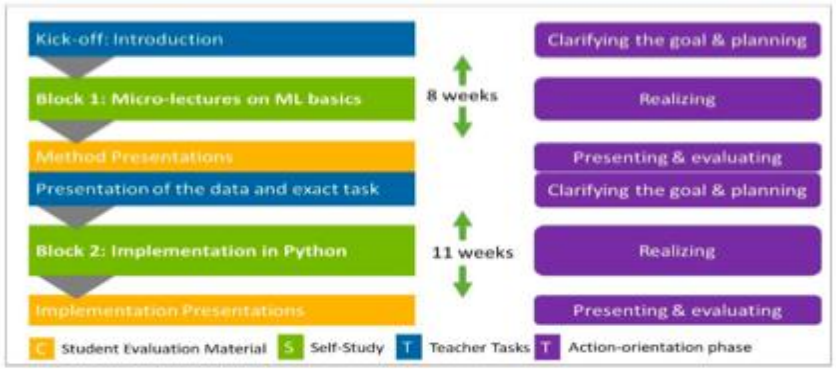


Figure 1. Timeline of the ML course

Following the action orientation approach, the students will be able to undergo the process of goal clarification, planning, realizing, presenting, and evaluating. During the kick-off, the outline and goal of the course will be presented to the students. The goal is to empower the students to solve a real-world use case with ML techniques. In the realization phase of the first part of the course, students are be provided with micro-lectures on relevant topics and tutorials for the practical implementation. Thus, they can learn the required methods through tools. The tutorials contain well-documented exemplary code and tasks about code modification. The combination of micro-lectures and tutorials ensures that the students both understand how the ML algorithms work and are able to implement them. The first part of the course will end with a presenting and evaluating phase, in which the students will receive feedback. The second part of the course will focus on the implementation of ML. The students will work on real world problems and finally present their results and obtain feedback.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

1.13 Assessment logic: performance, process, and reflection

In a challenge-based didactic approach, assessment should measure:

- **Technical competence:** the ability to design and implement a complete and functioning machine-learning pipeline, including model training and validation to achieve performance levels appropriate to the learners' experience and course objectives.
- **Methodological understanding:** the ability to justify modelling decisions, select appropriate evaluation metrics, interpret results correctly, and critically discuss model assumptions, limitations, and trade-offs.
- **Data literacy:** the quality of data exploration and preprocessing, including informed handling of missing values, feature selection or engineering, and reasoned decisions regarding data representation.
- **Team process and communication:** the level of contribution to collaborative work, effectiveness of teamwork, and clarity, structure, and coherence of oral or written presentations of results.
- **Risk awareness:** explicit reflection on ethical considerations, potential risks, biases, and limitations of machine-learning models, as well as awareness of responsible and context-appropriate application.

This framework aligns with the concept's intention to promote not only "technical model building," but also transparent, reliable, and responsible machine-learning practice.

1.14 Industry collaboration: authenticity, data, and realism

The concept places industrial datasets and problems at the heart of challenges. This serves several educational purposes:

- strengthens authenticity (problem orientation) - real industrial datasets and use cases expose students to problem settings that closely resemble professional practice. Learning activities are framed around realistic objectives, constraints, and decision-making contexts, which supports the development of applied analytical thinking rather than purely theoretical model construction,
- increases student motivation - working with data originating from real companies and industry scenarios enhances students' intrinsic motivation by demonstrating the practical relevance and societal value of machine-learning applications. Students are more engaged when they perceive a clear connection between course activities and real professional challenges,
- highlights constraints typical of real ML work (messy data, unclear objectives, stakeholder needs) - industry-based challenges make visible the typical limitations and complexities of applied ML work, such as incomplete or noisy data, evolving problem definitions, conflicting stakeholder requirements, and trade-offs between performance,

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

interpretability, and feasibility. This supports more realistic expectations of ML systems and discourages overly simplified or idealized solutions,
- and makes interdisciplinary collaboration meaningful - authentic challenges naturally require the integration of technical, domain-specific, and organizational perspectives. Students are encouraged to collaborate across roles and competencies, reflecting the interdisciplinary nature of real-world ML projects involving data scientists, domain experts, and decision-makers.

At the same time, the concept acknowledges that setting up real-world challenges is intensive and involves collaboration with companies, data anonymization and preparation, and detailed use-case descriptions. This is why the micro-lecture format and modular design are important: they reduce redesign effort and enable re-use across challenges and partners.

1.15 The core promise of the didactic concept: a reusable framework, not a one-off course

This didactic concept should be understood as a **framework that standardizes how to design, teach, and evaluate ML learning via challenges**, while still being flexible enough to support different domains, datasets, and cohorts.

In practical terms, the concept promises:

- **reusability** - learning materials are designed as modular micro-lectures, tutorials, and supporting resources that can be recombined across different courses, challenges, and datasets,
- **adaptability** - the concept allows instructors to tailor learning experiences by adjusting difficulty levels, selecting appropriate tools, and sequencing introductory and advanced units according to learners' prior knowledge,
- **inclusivity** - by combining self-paced online micro-lectures with guided challenge-based activities, the concept enables meaningful participation of learners with heterogeneous backgrounds, including differences in prior technical knowledge, programming experience, or disciplinary orientation,
- **authentic competence development** - learning is structured around an iterative action cycle that mirrors real machine-learning practice, including problem understanding, data exploration, modelling, evaluation, and reflection. The use of real or realistic problem contexts ensures that competence development goes beyond abstract model building and supports transferable, practice-oriented skills,
- and **responsible ML awareness** - consideration of risks, limitations, biases, and ethical implications of ML systems is explicitly integrated into learning goals, challenges, and assessment criteria.

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.

2 Structure of a MachineLearnAthon Challenge

This chapter defines the **internal structure of a MachineLearnAthon challenge**.
Each challenge included in the MachineLearnAthon catalogue follows this structure to ensure didactic consistency, transparency, and reusability across different domains, difficulty levels, and institutional contexts.

The structure described in this layer specifies:

- which components a challenge consists of,
- what type of information is included in each component,
- and how these components collectively support action-oriented, challenge-based machine learning education.

The internal structure operationalizes the didactic principles introduced in the previous chapter and translates them into a clear and repeatable structure for individual challenges.

2.1 Problem context and task definition

This section defines the **context and purpose** of the challenge.
It describes the real-world or realistic background in which the problem is situated, the practical or analytical goal to be achieved, and the relevance of the task in a professional or societal setting. The problem context frames the challenge as an authentic application of machine learning rather than a purely technical exercise.

The machine learning task type (e.g. classification, regression, clustering) is explicitly defined. At the same time, the formulation allows room for interpretation and refinement during the

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.

learning process, reflecting the iterative and exploratory nature of real-world ML problem solving.

2.2 Data and evaluation framework

This section defines the **technical basis** of the challenge. It specifies the dataset used in the challenge, which may be real, anonymized, or synthetically generated. The description includes the nature of the input variables and, where applicable in supervised learning tasks, the target variable. For unsupervised tasks such as clustering, no target variable is provided. The data description provides sufficient information to understand the structure of the problem without prescribing a single solution approach.

The evaluation framework defines how solutions are assessed. Evaluation metrics are selected to match the ML task type and the intended difficulty level. Beyond performance measurement, the evaluation framework serves as a learning instrument that supports discussion of trade-offs, limitations, and interpretation of results.

2.3 Difficulty-level differentiation within the challenge

Each MachineLearnAthon challenge is designed with three levels of difficulty: Beginner, Intermediate, and Advanced. These levels are used to describe the expected degree of familiarity with programming, machine learning tools, and ML concepts, rather than to prescribe specific tutorials or technical solutions.

The purpose of difficulty levels is to make challenges accessible to participants with different backgrounds while allowing progressive engagement with machine learning. The level of a challenge reflects how intensively participants are expected to interact with programming, ML models, and computational tools.

The description of each difficulty level therefore includes an explanation of why the challenge is considered to belong to that level, in relation to the others.

2.3.1 Beginner level

The Beginner level is intended for participants with little or no prior experience in programming or machine learning. At this level, contact with programming languages and ML models is minimal or introductory.

Typical characteristics of the Beginner level include:

- no prior requirement for programming experience,
- no-code or low-code interaction instead of full programming-level model implementation,
- emphasis on understanding the problem context and results rather than on implementation details,

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

- use of environments where programming concepts are introduced gradually or abstracted.

A challenge is considered Beginner level when it can be completed without sustained coding activity and when the primary learning focus lies on understanding ML concepts, workflows, and outcomes rather than on programming itself.

2.3.2 Intermediate level

The Intermediate level is intended for participants who already have basic familiarity with programming concepts and who are ready to engage more actively with ML implementation.

Typical characteristics of the Intermediate level include:

- regular but manageable interaction with code,
- use of simple ML models and workflows,
- basic experimentation with data and model parameters,
- familiarity with environments such as Jupyter Notebooks or integrated development environments is assumed or developed quickly.

A challenge is considered Intermediate level when participants are expected to modify code, run simple ML experiments, and work with programming environments without these being entirely new to them.

2.3.3 Advanced level

The Advanced level is intended for participants who actively seek to deepen their machine learning competence and who are already comfortable with programming.

Typical characteristics of the Advanced level include:

- frequent and autonomous interaction with code and ML models,
- use of more complex or less constrained ML workflows,
- ability to work independently in programming environments,
- focus on methodological decisions, model comparison, and critical evaluation.

A challenge is considered Advanced level when programming is not a barrier to participation and when the primary learning objective is to explore machine learning methods, assumptions, and limitations in greater depth.

By defining difficulty levels in terms of **expected engagement with programming and machine learning**, the MachineLearnAthon concept avoids rigid technical prescriptions while still providing clear orientation for participants and educators. This approach supports inclusive participation and flexible course design while preserving a common understanding of challenge complexity.

2.4 Micro-lectures as modular theoretical building blocks

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Micro-lectures provide focused conceptual input that supports understanding of the methods and concepts required to address the challenge. They are modular in nature and can be combined flexibly depending on the challenge and difficulty level.

This section clarifies which theoretical topics are relevant for the challenge, such as specific ML paradigms, model concepts, evaluation metrics, or known risks and limitations. Micro-lectures are referenced as supporting elements rather than reproduced in full, emphasizing their role as targeted conceptual scaffolding.

2.5 Tutorials as guided skill-construction units

Tutorials are used to develop hands-on skills that are required for autonomous challenge work. They focus on applying ML tools and workflows in a guided manner and enable step-by-step construction of practical competence.

Typical tutorial content includes activities such as:

- exploring and understanding datasets,
- implementing baseline models,
- modifying model parameters,
- comparing model performance,
- interpreting and documenting results.

This component ensures a structured transition from theoretical understanding to independent problem solving.

2.6 Programming languages, tools, and execution environments

MachineLearnAthon challenges are designed to be technologically flexible while remaining didactically consistent. The challenge-based approach deliberately separates learning objectives and outcomes from specific programming languages, tools, or execution environments.

For data analytics and machine learning activities, most tutorials are based on Python, which serves as a common reference language due to its widespread adoption and extensive ecosystem. However, the use of Python is not a strict requirement. Challenges are defined by tasks, data, and evaluation criteria, and solutions are assessed based on results and reasoning rather than on a prescribed implementation language.

To support learners with different technical backgrounds, tutorials are provided across multiple environments, including:

- **No-code and low-code environments**, such as *Orange Data Mining*, enabling ML experimentation without programming.
- **Interactive notebook environments**, such as *Jupyter Notebooks*, supporting exploratory analysis and reproducible workflows.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

- **Cloud-based notebook environments**, such as *Google Colab*, providing browser-based access without local installation.
- **Integrated Development Environments**, such as *Visual Studio Code*, supporting structured and extended ML projects.

In addition, tutorials introduce container-based environments using *Docker*, demonstrating reproducible execution of ML workflows, including running Jupyter Notebooks and integrating Docker with development tools. These tutorials are optional and primarily serve learners interested in professional ML workflows and environment management.

Overall, the MachineLearnAthon concept does not prescribe a single technical stack. Instead, it provides a curated set of tools and environments that support different levels of abstraction while maintaining comparable challenge outcomes.

2.7 Challenge platforms and execution environments

This component describes the execution and submission framework for challenges.

Challenges may be implemented using:
- dedicated ML challenge platforms (e.g. Kaggle),
- cloud-based notebook platforms,
- or learning management systems with integrated submission functionality.

Regardless of the selected platform, the execution framework must support:
- structured submission of results,
- transparent and reproducible evaluation,
- and feedback mechanisms.

Gamification elements, such as leaderboards, may be included where appropriate and are embedded within a pedagogically framed learning environment.

2.8 Example of course-level assessment using a challenge

This component provides an illustrative example of how a MachineLearnAthon challenge *may* be used as part of course-level assessment. It does not define mandatory assessment rules and does not prescribe a single evaluation model. Instead, it demonstrates how challenge results can be interpreted and assessed within a broader educational context.

The challenge-based approach deliberately separates challenge execution from formal assessment decisions. While challenges define tasks, data, and evaluation metrics, the way in which results contribute to grading or certification remains flexible and dependent on local curricular requirements.

An example assessment approach may consider multiple dimensions, such as:

- the achieved model performance relative to the defined evaluation metric,

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

- the correctness and coherence of the implemented ML workflow,
- the quality of result interpretation and methodological justification,
- the ability to reflect on limitations, risks, and assumptions,
- and the clarity of documentation or presentation of results.

Depending on course design, these aspects may be assessed individually or in combination and may be weighted differently. For example, performance metrics may be emphasized in technically oriented courses, while interpretation and reflection may play a stronger role in interdisciplinary or business-oriented settings.

3 Using MachineLearnAthon challenges in educational practice

The MachineLearnAthon concept is intentionally modular. Challenges represent the central learning units, while micro-lectures and tutorials serve as supporting elements that can be selected and combined depending on course objectives, learner profiles, and available teaching time.

3.1 Selection and combination of challenges and learning materials

MachineLearnAthon challenges can be used individually or combined to form a complete course. Depending on the scope of the course and the intended learning outcomes, one or several challenges may be selected. Challenges may also be combined across different machine learning tasks or difficulty levels to create a coherent learning pathway.

Micro-lectures and tutorials are not tied to a fixed order. Instead, they can be freely combined to support the selected challenges. This flexibility allows learning paths to be adapted to the background and needs of participants for example, by placing greater emphasis on conceptual understanding, hands-on experimentation, or deeper methodological exploration.

All micro-lectures and tutorials provided in video form are freely available through the MachineLearnAthon YouTube channel, supporting self-paced learning and reuse across institutions. The datasets used in tutorials are available for download on the MachineLearnAthon project website, ensuring easy and consistent access to all required learning materials.

3.2 Use of challenge datasets and evaluation approaches

For challenge-based evaluation, the preferred approach within the MachineLearnAthon framework is the use of Kaggle. Kaggle supports structured submissions, transparent evaluation, and leaderboard-based feedback, which enables comparison of results across student teams, institutions, and countries. This approach supports motivation, engagement, and international collaboration.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union
Machine LearnAthon

In addition to this preferred approach, alternative evaluation strategies are possible when local requirements or constraints apply.

If a challenge is not evaluated using the Kaggle platform, evaluation can be carried out locally. In such cases, special care should be taken to prevent unintended access to solution files or evaluation data.

There are two practical ways to proceed:

- **Request evaluation files from the MachineLearnAthon team.**

  If teachers wish to run their own evaluation or adapt an existing challenge without using Kaggle, they should contact the MachineLearnAthon project team by email. Upon request and explanation of the use case, the team can provide the necessary datasets and evaluation files in a controlled form to support local assessment.

- **Create a custom data split from the training data.**

  Alternatively, teachers may use the training dataset provided with the challenge and create their own training-test split for local evaluation. In this case, the original test set is not used, and evaluation is performed entirely on the locally defined data split.

Both approaches allow challenges to be reused in local teaching settings while reducing the risk of solution leakage and maintaining the integrity of the assessment process.

3.3 Integration into course-level assessment

Challenges can be used as formative learning activities, summative assessment components, or a combination of both. Depending on course design, challenge results may contribute to grades through model performance, documentation, interpretation, or reflective analysis.

The MachineLearnAthon concept does not impose a single assessment model. Instead, it provides structured challenges and comparable evaluation logic that can be aligned with local assessment regulations and pedagogical preferences.

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.

4 Example Challenge Catalogue Entry

4.1 Bid2Win: Tender Outcome Prediction Challenge

Keywords: Public Procurement, Price Prediction, Machine Learning, Historical Data, OpenTender

Machine learning task: Regression

Target variable: Final_Price_EUR

1. Problem context and task definition

Public procurement refers to formal purchasing procedures conducted by governments and public institutions. In each procurement procedure, multiple suppliers submit bids, and the final awarded price is one of the most important outcomes of the process.

The goal of this challenge is to **predict the final awarded price of a public procurement tender** using historical procurement data. Such prediction tasks are relevant for procurement analytics, market analysis, and policy evaluation, and they reflect real-world complexity caused by differences in legislation, economic conditions, market structures, and bidder behaviour across countries.

The task focuses on predicting the **lowest final bidding price**, expressed in euros (Final_Price_EUR), using information available at the tender level. All prices in the dataset are already converted to a common currency, ensuring comparability across countries.

This challenge serves as an **introductory applied machine learning task**, particularly suitable for learners encountering ML concepts for the first time in a business or public-sector context.

2. Data and evaluation framework

The dataset contains historical public procurement records collected from multiple countries. It includes both numerical and categorical variables describing tender characteristics, procedural attributes, buyer information, and estimated and final prices.

Key characteristics of the dataset:

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union

Machine LearnAthon

- multi-country and multi-year coverage,
- heterogeneous procurement categories (CPV codes),
- combination of numerical indicators and high-cardinality categorical variables,
- standardized monetary values in EUR.

The dataset includes information on public procurement procedures collected from multiple countries. It contains categorical and numerical features describing each tender, including tender characteristics (country, size, supply type, procedure type, CPV classification, and year), procedure and funding indicators (EU-funded status, framework agreements, GPA applicability, lots, electronic auctions), buyer-related attributes (buyer type, NUTS region, country), as well as estimated and final prices.

This dataset and associated competition uses data from the Open Tender platform https://opentender.eu/all (Government Transparency Institute) licensed under Creative Commons BY-NC-SA 4.0 for educational and non-commercial research purposes only.
Neither the dataset nor any derived models or insights or code will be used in commercial products or services or internal business processes.

The competition has been developed within the framework of the Erasmus+ MachineLearnAthon project, see website https://dss.lfo.tu-dortmund.de/. Both the dataset and the competition are to be used strictly for educational academic and research activities. Any form of commercial exploitation direct or indirect including but not limited to incorporation into paid services or products or consulting or proprietary systems is expressly prohibited under the terms of the license. The use of this data and competition materials must fully comply with the non-commercial clause of the Creative Commons BY-NC-SA 4.0 license. All content (including data) on this website is licensed under Creative Commons BY-NC-SA 4.0. If you require a different license email info@govtransparency.eu.

The challenge uses a **regression-based evaluation logic**, focusing on how well predicted prices match actual final prices. The model's predictive performance will be evaluated using standard regression metrics, with R-squared (R^2) as the primary evaluation measure. R^2 indicates how well the model explains the variation in Final_Price_EUR. Higher values mean the model captures underlying patterns in procurement pricing more effectively.

3. Difficulty level: Beginner

This challenge is classified as **Beginner level**.

The difficulty level is defined based on the **expected level of interaction with programming and machine learning tools**, not on mathematical complexity.

This challenge is considered Beginner-level because:

- no prior programming experience is required,
- the dataset is pre-processed and structured for immediate use,
- learners can complete the challenge using **no-code or low-code tools**,
- the focus is on understanding the problem, running models, and interpreting results rather than on implementing algorithms from scratch.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The challenge is particularly suitable for **business students or public administration students** who typically work with tools such as spreadsheets or web-based applications and are being introduced to machine learning concepts for the first time.

4. Relevant micro-lectures (theoretical background)

The following micro-lectures support this challenge:

- Introduction to machine learning
- Data preparation (for tabular data)
- Data Preprocessing
- Data visualization
- Feature Engineering
- Regression problems and target variables
- Simple and multivariate linear regression
- Regression analysis quide
- Advanced Concepts in Linear Regression: Assumptions, Dummy Variables, and Interactions
- Hyperparameter Tuning
- Introduction to XGBoost Algorithm
- Random Forest
- LightGBM
- Forecasting Methods
- Interpreting model performance and prediction errors
- Evaluation metrics
- Interpretability I-IV
- Overfitting
- No code
- Installing Python
- Low code I – Model creation
- Low code II – Model evaluation
- Fairness in Machine Learning

Micro-lectures are provided as **short video units** and are available through the public **MachineLearnAthon YouTube channel**. They are intended as conceptual preparation and can be watched independently of the challenge execution.

5. Supporting tutorials (practical preparation)

This challenge is supported by tutorials that demonstrate how to work with the dataset using different levels of technical abstraction.

Tutorials using similar datasets are:

- Docker desktop instalation guide
- Low code regression Part1

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

- Low code regression Part2
- Low code regression Part3
- Low code regression Part 4
- No code regression Part1
- No code regression Part2
- No code regression Part3
- Low code regression evaluation

The tutorials intentionally avoid complex programming constructs and emphasize workflow understanding.

6. Programming languages, tools, and environments

The challenge is designed to be **tool-flexible**, but the provided tutorials use **Python-based environments**.

Supported approaches include:

- **No-code solution:**
  - *Orange Data Mining (drag-and-drop workflows, suitable for complete beginners)*
- **Low-code solution:**
  - *PyCaret used within Jupyter Notebooks, enabling model training with minimal coding*

To ensure reproducibility and avoid dependency issues across different systems, PyCaret-based tutorials are also provided in a **Docker-based environment**. This allows learners to run the tutorials without dealing with library version conflicts or local installation problems.

7. Platforms and execution environment

This challenge is primarily designed to be evaluated using a **Kaggle-style challenge platform**. Kaggle provides a standardized and transparent environment for challenge-based machine learning tasks and supports fair comparison of results across students, institutions, and countries.

The use of Kaggle follows a common workflow that is consistent across all MachineLearnAthon challenges.

Access to challenge data

On the Kaggle challenge page, participants have access to the following files:

- **Training dataset**
  This dataset contains all input features as well as the target variable. It is used to explore the data, train models, and validate approaches.

- **Test dataset**

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.

This dataset contains the same input features as the training data but does not include the target variable. It is used to generate predictions that are submitted for evaluation.

- **Sample submission file**
  This file shows the required format of the submission and serves as a template for creating prediction outputs.

All files can be downloaded directly from the Kaggle challenge page.

Submission format

Predictions are submitted to Kaggle in the form of a **CSV file**.

Each submission file must contain exactly two columns:

1. **Procurement_ID**
   A unique identifier for each procurement procedure (uppercase "ID").

2. **Final_Price_EUR**
   The predicted final procurement price in euros (uppercase "EUR").

The structure of the submission file must match the sample submission provided on the Kaggle page. Only correctly formatted submissions are accepted by the platform. After submission, Kaggle automatically evaluates the predictions using the predefined evaluation metric and assigns a score.

Leaderboards and evaluation

Kaggle uses a **leaderboard-based evaluation system**, which allows participants to compare their results with others.

Two types of leaderboards are typically used:

- **Public leaderboard**
  This leaderboard evaluates submissions on a subset of the test data and provides immediate feedback during the challenge. It allows participants to track their progress and experiment with different approaches.

- **Private leaderboard**
  This leaderboard evaluates submissions on a hidden subset of the test data and is revealed at the end of the challenge. It determines the final ranking and helps prevent overfitting to the public evaluation data.

This two-stage evaluation approach supports fair comparison of solutions and encourages robust modelling rather than optimization for a single score.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

8. Learning focus and intended outcomes

This challenge is designed as an introductory, application-oriented machine learning task. Its primary focus is on:

- understanding and framing a real-world business problem in a data-driven context,
- working with structured, tabular datasets commonly encountered in business and public-sector analytics,
- applying machine learning models using no-code or low-code tools, without requiring advanced programming skills,
- interpreting model outputs and performance metrics in a meaningful and business-relevant way.

The emphasis of the challenge lies on learning how machine learning can be applied in practice, rather than on mastering algorithmic details or implementing models from scratch. The dataset is intentionally prepared to support this learning goal. It consists of predominantly structured numerical and categorical variables and is not intended for advanced natural language processing tasks, such as text mining of free-form descriptions, nor for entity extraction from names, addresses, or unstructured documents. Likewise, the challenge does not focus on complex data imputation strategies or large-scale data integration problems. While minor data cleaning steps may be required, extensive preprocessing, feature extraction from text, or domain-specific data engineering are deliberately kept out of scope.

9. Example of course-level assessment using a challenge

The evaluation of challenges is based on a combination of model performance and submission quality. The primary goal of this approach is to encourage experimentation while ensuring fair and transparent assessment.

For each student, model performance is evaluated using the **primary evaluation metric defined on the Kaggle platform** (e.g. R^2 for regression tasks). Rather than using this metric to calculate exact grades, it is used to **place submissions into performance tiers**.

To support fair and transparent evaluation, three reference performance levels are identified:

- a basic reference, corresponding to a minimal working solution that successfully runs and produces a valid Kaggle submission,
- a tutorial reference, corresponding to the score achieved by following the provided tutorial workflow step by step,
- an advanced reference, corresponding to strong performance, defined either by the instructor or based on the private Kaggle leaderboard results (for example, top-performing submissions).

Based on where a submission's Kaggle score falls relative to these reference levels, it is assigned to one of the predefined performance tiers (Basic, Tutorial, Improved, or Advanced). The exact number of points awarded within a tier is then determined by the overall quality of the submission rather than by small differences in the Kaggle score.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Submissions are grouped into four performance tiers:

- Basic tier (up to 25 points): minimal working solution,
- Tutorial tier (up to 50 points): tutorial-level performance,
- Improved tier (up to 75 points): performance beyond the tutorial baseline,
- Advanced tier (up to 100 points): performance close to or exceeding the advanced reference.

Within each tier, the final number of points is determined based on the **quality of the submission**, not on small differences in the performance metric. Quality considerations may include reproducibility, clarity of the workflow, basic data handling, model justification, and interpretation of results.

Each tier allows for internal differentiation (for example, very weak to very strong submissions within the same tier), ensuring that students are rewarded not only for achieving a certain performance level, but also for how thoughtfully and transparently the solution was developed.

4.2 Bid2Save: Savings Prediction Challenge

Keywords: Public Procurement, Saving Classification, Machine Learning, Prediction, Historical Data, OpenTender, Classification Models

Machine learning task: Classification

Target variable: saving_bin

1. Problem context and task definition

Public procurement refers to structured purchasing procedures conducted by governments and public institutions. In each procurement process, contracting authorities publish an estimated price, suppliers submit bids, and a final awarded price is determined at the end of the tendering procedure. An important analytical question in public procurement is whether a given tender achieves meaningful savings, that is, whether the final awarded price is sufficiently lower than the initial estimated price. Addressing this question is highly relevant for procurement analytics, strategic planning, and policy evaluation, as it provides insights into market competitiveness and the effectiveness of procurement design.

The goal of this challenge is to classify public procurement tenders based on whether they result in significant savings. Instead of predicting a continuous price, the task is formulated as a binary classification problem, where the target variable **saving_bin** labels each tender as **saving** (sufficient reduction from the estimated price) or **no_saving** (no meaningful savings achieved).

The classification relies exclusively on **information available before** the award decision. The dataset covers procurement procedures from multiple countries and categories, reflecting

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

differences in legislation, economic conditions, market structures, and bidder behaviour, which introduce realistic complexity into the task.

This challenge serves as an **introductory applied machine learning classification task** in a public-sector context, emphasizing robust and generalizable models rather than solutions tailored to a single country or procurement setting.

2. Data and evaluation framework

The dataset contains historical public procurement records collected from multiple countries. It includes both numerical and categorical variables describing tender characteristics, procedural attributes, buyer information, and estimated and final prices.

Key characteristics of the dataset:

- multi-country and multi-year coverage,
- heterogeneous procurement categories (CPV codes),
- combination of numerical indicators and high-cardinality categorical variables,
- standardized monetary values in EUR.

The dataset includes information on public procurement procedures collected from multiple countries. It contains categorical and numerical features describing each tender, including tender characteristics (country, size, supply type, procedure type, CPV classification, and year), procedure and funding indicators (EU-funded status, framework agreements, GPA applicability, lots, electronic auctions), buyer-related attributes (buyer type, NUTS region, country), as well as estimated and final prices.

This dataset and associated competition uses data from the Open Tender platform https://opentender.eu/all (Government Transparency Institute) licensed under Creative Commons BY-NC-SA 4.0 for educational and non-commercial research purposes only.
Neither the dataset nor any derived models or insights or code will be used in commercial products or services or internal business processes.

The competition has been developed within the framework of the Erasmus+ MachineLearnAthon project, see website https://dss.lfo.tu-dortmund.de/. Both the dataset and the competition are to be used strictly for educational academic and research activities. Any form of commercial exploitation direct or indirect including but not limited to incorporation into paid services or products or consulting or proprietary systems is expressly prohibited under the terms of the license. The use of this data and competition materials must fully comply with the non-commercial clause of the Creative Commons BY-NC-SA 4.0 license. All content (including data) on this website is licensed under Creative Commons BY-NC-SA 4.0. If you require a different license email info@govtransparency.eu.

3. Difficulty level: Beginner

This challenge is classified as **Beginner level**.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The difficulty level is defined based on the **expected level of interaction with programming and machine learning tools**, not on mathematical complexity.

This challenge is considered Beginner-level because:

- no prior programming experience is required,
- the dataset is pre-processed and structured for immediate use,
- learners can complete the challenge using **no-code or low-code tools**,
- the focus is on understanding the problem, running models, and interpreting results rather than on implementing algorithms from scratch.

The challenge is particularly suitable for **business students or public administration students** who typically work with tools such as spreadsheets or web-based applications and are being introduced to machine learning concepts for the first time.

4. Relevant micro-lectures (theoretical background)

The following micro-lectures support this challenge:

- Introduction to ML
- Classification I
- Classification II
- Classification III
- Data preprocessing
- Data preparation (for tabular data)
- Data visualization
- Random Forest
- LightGBM
- Introduction to XGBoost Algorithm
- Evaluation metrics
- Imbalanced Datasets
- Hyperparameter Tuning
- Interpretability in machine learning I – IV
- Low code I – Model creation
- Low code II – Model evaluation
- No code
- Fairness in Machine Learning

Micro-lectures are provided as **short video units** and are available through the public **MachineLearnAthon YouTube channel**. They are intended as conceptual preparation and can be watched independently of the challenge execution.

5. Supporting tutorials (practical preparation)

This challenge is supported by tutorials that demonstrate how to work with the dataset using different levels of technical abstraction.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Tutorials using similar datasets are:

- Docker desktop instalation guide
- Low code classification Part1
- Low code classification Part2
- Low code classification Part3
- Low code classification Part 4
- No code classification Part1
- No code classification Part2
- No code classification Part3
- Low code classification evaluation

The tutorials intentionally avoid complex programming constructs and emphasize workflow understanding.

6. Programming languages, tools, and environments

The challenge is designed to be **tool-flexible**, but the provided tutorials use **Python-based environments**.

Supported approaches include:

- **No-code solution:**
  - *Orange Data Mining (drag-and-drop workflows, suitable for complete beginners)*
- **Low-code solution:**
  - *PyCaret used within Jupyter Notebooks, enabling model training with minimal coding*

To ensure reproducibility and avoid dependency issues across different systems, PyCaret-based tutorials are also provided in a **Docker-based environment**. This allows learners to run the tutorials without dealing with library version conflicts or local installation problems.

7. Platforms and execution environment

This challenge is primarily designed to be evaluated using a **Kaggle-style challenge platform**. Kaggle provides a standardized and transparent environment for challenge-based machine learning tasks and supports fair comparison of results across students, institutions, and countries.

The use of Kaggle follows a common workflow that is consistent across all MachineLearnAthon challenges.

Access to challenge data

On the Kaggle challenge page, participants have access to the following files:

- **Training dataset**

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

This dataset contains all input features as well as the target variable. It is used to explore the data, train models, and validate approaches.

- **Test dataset**
  This dataset contains the same input features as the training data but does not include the target variable. It is used to generate predictions that are submitted for evaluation.

- **Sample submission file**
  This file shows the required format of the submission and serves as a template for creating prediction outputs.

All files can be downloaded directly from the Kaggle challenge page.

Submission format

Predictions are submitted to Kaggle in the form of a **CSV file**.

Each submission file must contain exactly two columns:

1. **Procurement_ID**
   A unique identifier for each procurement procedure (uppercase "ID").

2. **saving_bin**
   Binary indicator of whether a procurement tender achieved meaningful savings, labelled as *saving* or *no_saving*.

The structure of the submission file must match the sample submission provided on the Kaggle page. Only correctly formatted submissions are accepted by the platform. After submission, Kaggle automatically evaluates the predictions using the predefined evaluation metric and assigns a score.

Leaderboards and evaluation

Kaggle uses a **leaderboard-based evaluation system**, which allows participants to compare their results with others.

Two types of leaderboards are typically used:

- **Public leaderboard**
  This leaderboard evaluates submissions on a subset of the test data and provides immediate feedback during the challenge. It allows participants to track their progress and experiment with different approaches.

- **Private leaderboard**
  This leaderboard evaluates submissions on a hidden subset of the test data and is revealed at the end of the challenge. It determines the final ranking and helps prevent overfitting to the public evaluation data.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

This two-stage evaluation approach supports fair comparison of solutions and encourages robust modelling rather than optimization for a single score.

8. Learning focus and intended outcomes

This challenge is designed as an introductory, application-oriented machine learning task. Its primary focus is on:

- understanding and framing a real-world business problem in a data-driven context,
- working with structured, tabular datasets commonly encountered in business and public-sector analytics,
- applying machine learning models using no-code or low-code tools, without requiring advanced programming skills,
- interpreting model outputs and performance metrics in a meaningful and business-relevant way.

The emphasis of the challenge lies on learning how machine learning can be applied in practice, rather than on mastering algorithmic details or implementing models from scratch. The dataset is intentionally prepared to support this learning goal. It consists of predominantly structured numerical and categorical variables and is not intended for advanced natural language processing tasks, such as text mining of free-form descriptions, nor for entity extraction from names, addresses, or unstructured documents. Likewise, the challenge does not focus on complex data imputation strategies or large-scale data integration problems. While minor data cleaning steps may be required, extensive preprocessing, feature extraction from text, or domain-specific data engineering are deliberately kept out of scope.

9. Example of course-level assessment using a challenge

The evaluation of challenges is based on a combination of model performance and submission quality. The primary goal of this approach is to encourage experimentation while ensuring fair and transparent assessment.

For each student, model performance is evaluated using the **primary evaluation metric defined on the Kaggle platform** (e.g. Accuracy for classification tasks). Rather than using this metric to calculate exact grades, it is used to **place submissions into performance tiers**.

To support fair and transparent evaluation, three reference performance levels are identified:

- a basic reference, corresponding to a minimal working solution that successfully runs and produces a valid Kaggle submission,
- a tutorial reference, corresponding to the score achieved by following the provided tutorial workflow step by step,
- an advanced reference, corresponding to strong performance, defined either by the instructor or based on the private Kaggle leaderboard results (for example, top-performing submissions).

Based on where a submission's Kaggle score falls relative to these reference levels, it is assigned to one of the predefined performance tiers (Basic, Tutorial, Improved, or Advanced).

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The exact number of points awarded within a tier is then determined by the overall quality of the submission rather than by small differences in the Kaggle score.

Submissions are grouped into four performance tiers:

- Basic tier (up to 25 points): minimal working solution,
- Tutorial tier (up to 50 points): tutorial-level performance,
- Improved tier (up to 75 points): performance beyond the tutorial baseline,
- Advanced tier (up to 100 points): performance close to or exceeding the advanced reference.

Within each tier, the final number of points is determined based on the **quality of the submission**, not on small differences in the performance metric. Quality considerations may include reproducibility, clarity of the workflow, basic data handling, model justification, and interpretation of results.

Each tier allows for internal differentiation (for example, very weak to very strong submissions within the same tier), ensuring that students are rewarded not only for achieving a certain performance level, but also for how thoughtfully and transparently the solution was developed.

4.3 Sales Forecasting

Keywords: Time Series Forecasting, Sales Prediction, Manufacturing Industry, Supply Chain Optimization, Demand Planning

Machine learning task: Time-Series Forecasting (Regression) & Clustering

Target variable: Sales Quantity

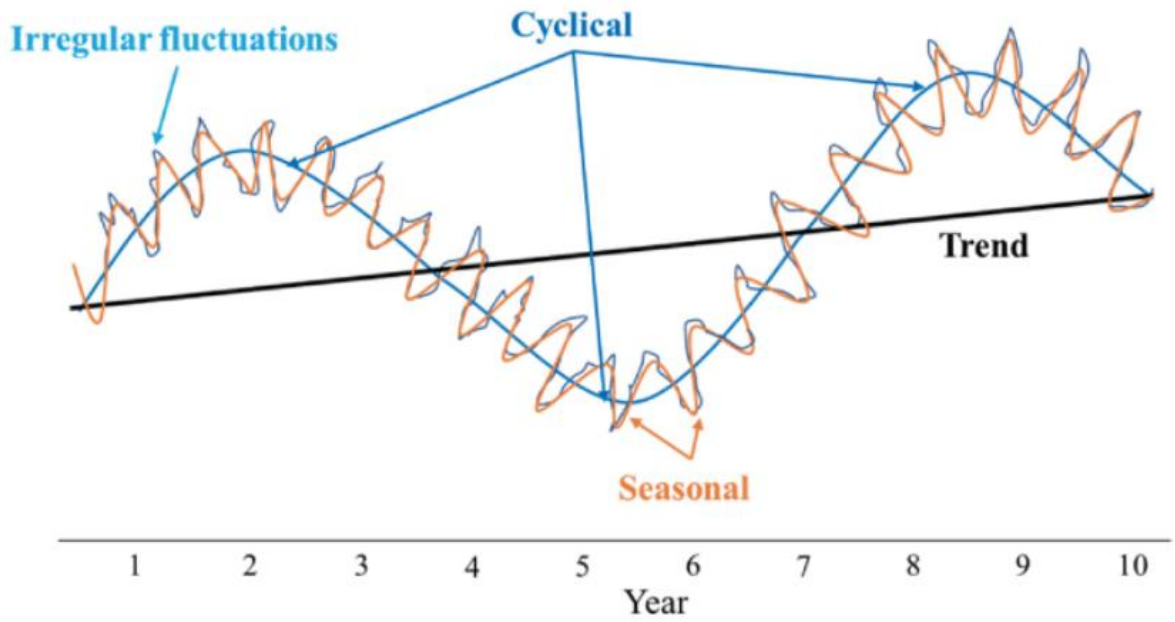
1. Problem context and task definition

Sales forecasts are indispensable for companies, especially in the manufacturing industry. They enable companies to make well-founded decisions. They therefore often serve as the basis for planning processes along the supply chain. Sales forecasts help to plan production capacities, optimize stock levels and avoid supply bottlenecks. To avoid planning errors, the accuracy of sales forecasts is of great importance. However, demand patterns can be very complex. They depend on various factors, such as the time series pattern, e.g. trend, seasonal or sporadic, the company's own marketing or the current purchasing power of customers.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon



Time series components [1]

In addition, nowadays more and more data is available for creating forecasts. Machine learning algorithms are able to generate precise forecasts from large volumes of data.

The aim of this challenge is to calculate a sales forecast for a company using machine learning methods. The dataset provides information about the sales of a production company located in the Ruhr area in Germany. The dataset is split chronologically. The training data contain the documentation of sales from December 1st, 2012, until June 30th, 2018. The test dataset contains the months July, August and September 2018. The task is to create monthly sales forecasts for this period. To forecast, please divide the training data into proper training data, validation data and test data.

To improve your sales prediction, we suggest to previously apply cluster methods to group the time series.

2. Data and evaluation framework

The dataset documents individual sales transactions from December 2012 to June 2018. Each record represents one sale and includes both categorical descriptors of the product and numerical information about the quantity sold. The date variable indicates the time of sale, though all timestamps have been slightly shifted to preserve privacy.

The feature **X1** identifies the product category, with five distinct products labeled P1 to P5. The features **X2** through **X7** are hierarchical anonymized categorical variables that describe additional aspects of the production or customer context. **X7** represents the exact product designation. Each value of **X7** maps uniquely to one value of **X6**. Each value of **X6** maps uniquely to one value of **X5**. **X4** represents subcategories of **X2**. These variables represent different dimensions of the manufacturing process or market segmentation but are kept anonymous for confidentiality reasons.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union

Machine LearnAthon

The variable **SalesQuantity** records the number of units sold per transaction and is numerical. **SalesUnit** specifies the measurement unit associated with each sale and can take one of several categorical values: PC (piece), M (meter), M² (square meter), or KG (kilogram). This attribute reflects differences in how products are quantified depending on their physical nature or intended use in production processes.

Participants should:

- Split the dataset into training, validation, and test sets,
- Apply appropriate preprocessing (e.g., handling missing values, date-based feature extraction),
- Build time series or regression-based forecasting models,
- Predict monthly sales values for July–September 2018.

Evaluation metric: Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

The RMSE penalizes large deviations more strongly than absolute error measures such as MAE — making it particularly suitable for assessing forecast accuracy in business contexts where extreme errors can lead to significant financial consequences.

Participants must also report their model's RMSE on the training data as an internal performance check.

3. Difficulty level: Advanced

The difficulty level of this challenge is **advanced,** as the dataset contains only a limited number of features and provides little contextual information about them. Consequently, extensive feature engineering is not required. However, participants should possess basic knowledge of **time series forecasting techniques**, including data splitting, trend and seasonality analysis, and evaluation using appropriate error metrics such as RMSE.

4. Relevant micro-lectures (theoretical background)

- Installing Python
- Introduction to ML
- Data Visualization
- Data Preparation
- Clustering
- Feature Extraction
- Time Series Forecasting
- Forecasting Methods
- Hyperparameter Tuning

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

- Overfitting
- Imbalanced Datasets
- Random Forest
- XGBoost
- LightGBM
- Evaluation metrics

5. Supporting tutorials (practical preparation)

- Introduction to Python tutorial
- Tutorial on Time-Series Forecasting – Part I
- Tutorial on Time-Series Forecasting – Part II

6. Programming languages, tools, and environments

Python libraries such as:

- pandas, numpy for preprocessing,
- scikit-learn or statsmodels for modeling,

7. Platforms and execution environment

The challenge is hosted on Kaggle — providing data access via:

Access to challenge data

https://www.kaggle.com/competitions/sales-forecasting-new

Submission format

CSV file containing columns in exact order: Date, X1, Forecast, SalesUnit

Leaderboards and evaluation

Leaderboards will rank participants by test-period RMSE scores calculated over July–September 2018 forecasts.

8. Learning focus and intended outcomes

Participants will learn how to:

- Build end-to-end forecasting models using real industrial datasets,
- Handle mixed-type categorical + numerical features,
- Engineer date-based features capturing seasonal effects,
- Interpret results within operational planning contexts.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Outcome: Practical experience applying ML methods to real-world demand forecasting problems relevant across manufacturing industries.

9. Example of course-level assessment using a challenge

For course-level assessment: Students receive an unseen dataset representing new products or later months beyond September 2018. They must build complete pipelines including preprocessing steps (encoding categorical variables), model selection/tuning based on validation RMSE results, generate forecasts for unseen periods, compute RMSE metrics on training/test sets, visualize predictions vs actuals, and discuss potential sources of error such as seasonality shifts or promotional effects.

Assessment emphasizes methodological soundness over leaderboard rank — focusing on reproducibility of results and understanding of forecasting dynamics under industrial uncertainty.

4.4 Blueberry Production Classification Competition

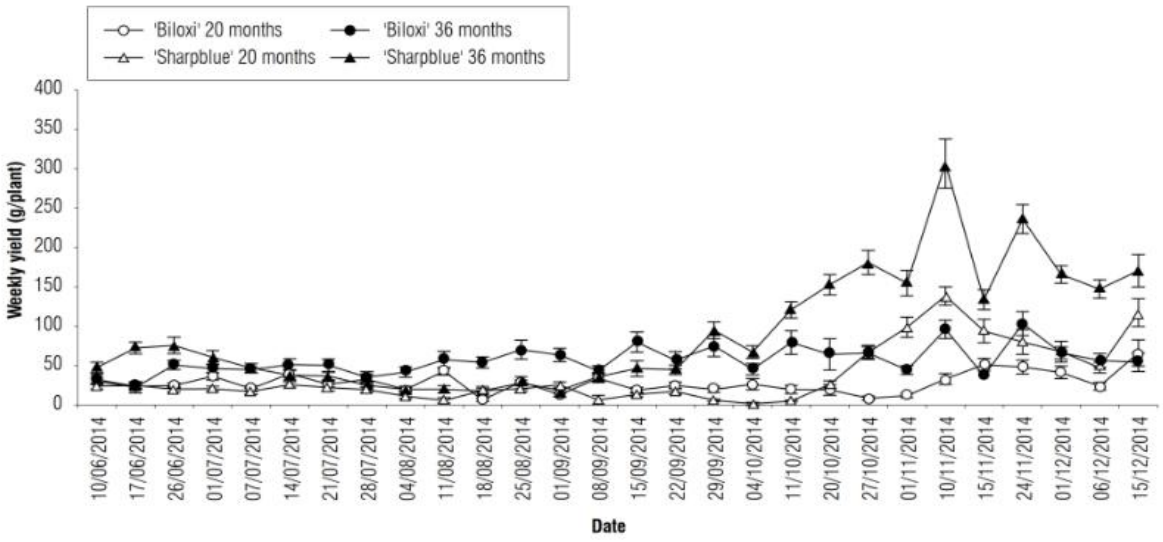
Keywords: Blueberry production, crop yield classification, machine learning, agricultural analytics, precision agriculture

Machine learning task: Classification (AutoML frameworks FLAML and MLJAR)

Target variable: The exact target variable is the **blueberry production class,** which is a categorical variable with three labels: **low, medium, and high** production.

1. Problem context and task definition

Blueberry production is an important economic activity for South American countries such as Peru, Argentina, and Chile, which supply global markets. Accurate estimation of crop yield is crucial for production planning, meeting export deadlines, and avoiding economic losses. However, yield estimation is challenging because it depends on many factors, including



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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union
Machine LearnAthon

irrigation, plant nutrition, soil conditions, vegetation indices (NDVI), and weather. As farms grow larger, manual estimation across multiple sectors becomes inefficient and prone to errors.

Figure 1. Weekly yield per plant for two blueberry cultivars. Figure taken from M. E. Cortes, P. A. Mesa, C. M. Grijalba, and M. M. Perez. (2016). Yield and Fruit Quality of the Blueberry Cultivars Biloxi and Sharpblue in Guasca, Colombia.

This competition addresses this challenge by using artificial intelligence to support agronomists in predicting blueberry production levels. Based on historical agronomic and environmental data, participants are required to build a supervised machine learning model that classifies production into three categories: low, medium, and high. The goal is to provide a reliable decision-support tool that improves yield estimation, planning, and overall farm management.

2. Data and evaluation framework

The dataset used in this challenge consists of agronomic, environmental, and management data collected from blueberry farms during the years **2021–2022**. Each record corresponds to a specific **week of the year** and captures multiple factors that influence blueberry growth and yield. The target variable is the **production class**, defined as **low**, **medium**, or **high** production.

The input features include indicators related to irrigation, plant nutrition, crop health, and weather conditions. Key variables include water supply (H2O), vegetation indices such as **NDVI**, evaporation (EVO), irrigation sheet (LR), nutrient levels (**N, P, K**), and weather attributes like temperature, humidity, wind velocity, rainfall, and solar radiation. Additionally, plant growth and yield characteristics are provided, such as stems per meter, total buds, buds per bunch, maturation level, BRIX degrees, bud size percentages, and aborted fruit percentage. Together, these variables offer a comprehensive view of factors affecting blueberry production.

Participants are expected to preprocess the data appropriately, including handling missing values and scaling numerical features where necessary, before training supervised classification models to predict the production class.

Evaluation metric: Accuracy

Model performance is evaluated using **classification accuracy**, which measures the proportion of correctly predicted production classes relative to the total number of samples. Accuracy is defined as:

$$\text{Accuracy}=\frac{1}{N}\sum_{i=1}^{N}1(\hat{y}_i=y_i)$$

where $N$is the total number of samples, $y_i$is the true production class, $\hat{y}_i$is the predicted class, and $1(\cdot)$is an indicator function that equals 1 when the prediction is correct and 0 otherwise. This metric provides a clear and interpretable measure of how well the model classifies blueberry production levels.

3. Difficulty level: Beginner

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The difficulty level of this challenge is beginner, as the dataset is well-structured and consists mainly of numerical features collected from agronomic, environmental, and management sources. The task focuses on a straightforward supervised classification problem with three clearly defined target classes: low, medium, and high production.

While some basic preprocessing steps such as handling missing values, feature scaling, and data splitting are required, extensive feature engineering is not necessary. Participants are expected to have foundational knowledge of machine learning classification techniques, model training and validation, and performance evaluation using accuracy. This makes the competition well suited for beginners who want to gain practical experience applying machine learning to real-world agricultural data.

4. Relevant micro-lectures (theoretical background)

- Installing Python
- Introduction to ML
- Data Visualization
- Data Preparation
- Feature Extraction
- Hyperparameter Tuning
- Overfitting
- Classification I
- Classification II
- Classification III
- Evaluation metrics

5. Supporting tutorials (practical preparation)

- Tutorial: AutoML frameworks FLAML and MLJAR

6. Programming languages, tools, and environments

The challenge can be effectively addressed using Python, which provides a rich ecosystem for data analysis and machine learning. Commonly used libraries include:

- pandas and NumPy for data loading, cleaning, preprocessing, and numerical operations,
- scikit-learn for building, training, and evaluating classification models, as well as feature scaling and model selection,
- Matplotlib and Seaborn for exploratory data analysis and visualization of feature distributions and model results.

7. Platforms and execution environment

The challenge is hosted on **Kaggle**, which provides an integrated and user-friendly platform for data science competitions. Kaggle offers direct access to the dataset, competition rules, and evaluation metrics through its web interface.

Access to challenge data

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

https://www.kaggle.com/competitions/blueberry-production-classification-competition/overview

Submission format

The submission must be provided as a CSV file with a header. For each record in the test dataset, participants are required to submit a prediction for the TARGET variable, which represents the blueberry production class.

The file should contain exactly two columns, in the following order:

- **ID**: the unique identifier of each test sample
- **TARGET**: the predicted production class (low, medium, or high)

The column names and their order must be strictly followed for the submission to be accepted.

Leaderboards and evaluation

Participants are ranked on the leaderboard based on their model's performance on the hidden test dataset. Evaluation is conducted using classification accuracy, which measures how correctly the model predicts the blueberry production classes (low, medium, high). Higher accuracy scores indicate better model performance, and the leaderboard is updated automatically after each valid submission.

8. Learning focus and intended outcomes

Participants will learn how to:

- Build end-to-end machine learning **classification models** using real agricultural datasets,
- Work with multivariate numerical features from agronomic, environmental, and weather data,
- Apply data preprocessing techniques such as feature scaling and train–validation splitting,
- Evaluate and interpret classification results in the context of crop production decision-making.

Outcome: Practical experience in applying machine learning techniques to real-world agricultural problems, with a focus on supporting yield estimation and improving farm management decisions.

9. Example of course-level assessment using a challenge

For course-level assessment, students are provided with an unseen dataset representing new weeks, fields, or growing conditions beyond the original 2021–2022 data. They are required to build a complete machine learning pipeline, including data preprocessing (handling missing values and feature scaling), model selection, and hyperparameter tuning based on validation accuracy.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Students must generate production class predictions for the unseen data, evaluate model performance on training and test sets, and present results using confusion matrices or accuracy scores. They are also expected to visualize key features and discuss possible sources of error, such as changes in weather patterns, irrigation practices, or crop health indicators.

Assessment focuses on the correctness and reproducibility of the methodology, clarity of analysis, and understanding of how agronomic and environmental factors influence blueberry production, rather than solely on leaderboard ranking.

4.5 Fiber Content in Wheat and Barley Grains Crops

Keywords: Arabinoxylan prediction, Crop quality modeling, Agricultural machine learning, Weather-based regression, Cereal fibre analysis

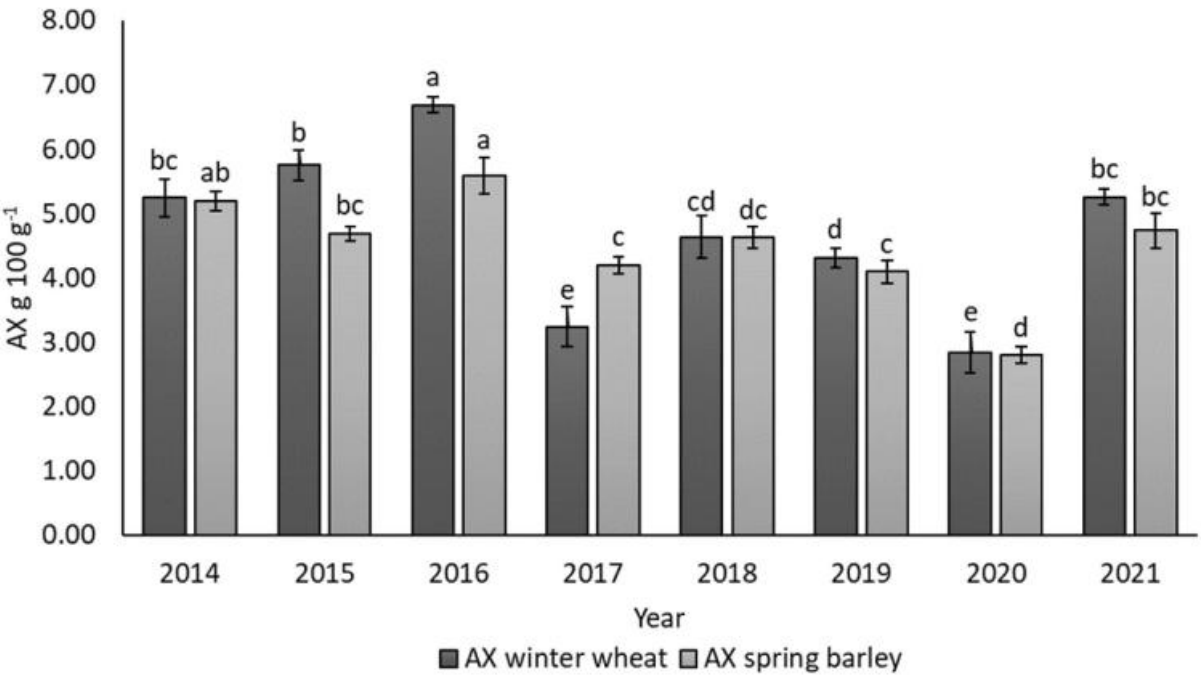
Machine learning task: Regression

Target variable: The target variable **is AX_CONTENT**, which represents the arabinoxylan content measured in **g/100g.**

1. Problem context and task definition

Arabinoxylan plays a central role in determining the nutritional quality of cereal crops such as wheat and barley, which are staple foods in human diets. According to Karge et. al. (2023), the levels of arabinoxylan are seasonal; winter and spring might influence the final quality of the fibre in both crops Figure 1.

Accurate estimation of arabinoxylan levels enables farmers and agronomists to make informed decisions regarding crop management practices, fertilizer application, and production strategies aimed at improving fibre quality and consumer health. This challenge focuses on



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.

predicting the level of arabinoxylan in cereal crops. Participants are tasked with developing machine learning models that estimate arabinoxylan content using a four-year historical dataset incorporating fertilizer treatments, crop systems, and temperature conditions. The goal is to design robust predictive models capable of capturing complex interactions between environmental factors and agricultural practices that influence fibre accumulation in wheat and barley.

Figure 1. The AX content as an average of N treatments in winter wheat and spring barley flours in different years.

2. Data and evaluation framework

The dataset contains agronomic and environmental data collected over several growing seasons and is provided in three files: train.csv, test.csv, and sample_submission.csv. Each record represents a single crop-year observation. The target variable, **AX_CONTENT**, denotes the arabinoxylan content measured in g/100g.

Key features include **YEAR, CROPPING_SYSTEM**, and **TREATMENT**, which describe temporal and management-related aspects of crop production. Weather conditions are captured through average monthly temperatures from April to August and precipitation variables, including both total and month-wise rainfall. The variable **BARLEY_YIELD** represents crop productivity in tons per hectare.

Participants should:

- Split the training data into training and validation sets,
- Perform appropriate preprocessing (e.g., handling missing values, encoding categorical features),
- Build regression-based machine learning models,
- Predict **AX_CONTENT** for the test dataset.

Evaluation metric:

Area Under the Receiver Operating Characteristic Curve (AUC-ROC)

The AUC-ROC evaluates the model's ability to discriminate between target outcomes across thresholds and provides a robust measure of predictive performance.

3. Difficulty level: Beginner

The difficulty level of this challenge is beginner, as the dataset contains a moderate number of well-defined features with clear agronomic and environmental meanings. Participants are not expected to perform highly complex feature engineering or advanced modeling techniques. However, they should have a basic understanding of regression modeling, categorical encoding, data preprocessing, and model evaluation. The task involves predicting a continuous agronomic outcome (arabinoxylan content) based on inputs such as cropping system, treatment, and seasonal weather data, requiring familiarity with standard machine learning workflows

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.

including data splitting, handling mixed data types, and evaluating predictive performance using appropriate metrics.

4. Relevant micro-lectures (theoretical background)

- Installing Python
- Introduction to ML
- Data Visualization
- Data Preparation
- Feature Extraction
- Hyperparameter Tuning
- Overfitting
- Regression analysis guide
- RandomForest
- Introduction to XGBoost Algorithm
- LightGBM
- Simple and multivariate linear regression
- Advanced Concepts in Linear Regression: Assumptions, Dummy Variables, and Interactions
- Evaluation metrics

5. Supporting tutorials (practical preparation)

- Tutorial: No code regression Part I
- Tutorial: No code regression Part II
- Tutorial: No code regression Part III
- Tutorial: xAI with LIME

6. Programming languages, tools, and environment

The challenge can be effectively addressed using **Python** as the primary programming language. Commonly used libraries include:

- **pandas** and **numpy** for data loading, cleaning, and preprocessing,
- **scikit-learn** for building and evaluating regression-based machine learning models,
- **matplotlib** or **seaborn** for exploratory data analysis and visualization.

Participants are encouraged to work in standard data science environments such as **Jupyter Notebook**, **Google Colab**, or local **Python IDEs**, which provide sufficient support for experimentation, model development, and result analysis.

7. Platforms and execution environment

The challenge is hosted on **Kaggle**, which provides direct access to the dataset, submission interface, and an integrated execution environment through Kaggle Notebooks for model development and evaluation.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Access to challenge data

https://www.kaggle.com/competitions/fiber-content-in-wheat-and-barley-grains-cropss/overview

Submission format

CSV file containing a header with the following columns in exact order: **ID, TARGET,** where **TARGET** represents the predicted probability for each corresponding **ID** in the test set.

Leaderboards and evaluation

Leaderboards rank participants based on their AUC-ROC score achieved on the test dataset, reflecting the model's ability to accurately predict arabinoxylan content across evaluation thresholds.

8. Learning focus and intended outcomes

Participants will learn how to:

- Build end-to-end regression models using real agricultural and environmental datasets,
- Handle mixed categorical and numerical features common in agronomic data,
- Analyse the impact of weather and management practices on crop quality,
- Interpret model predictions to support data-driven farming decisions.

Outcome: Practical experience in applying machine learning techniques to predict crop fibre content and improve agricultural quality assessment.

9. Example of course-level assessment using a challenge

For course-level assessment, students are provided with an unseen dataset representing new growing seasons or crop conditions not included in the training data. They are required to build a complete machine learning pipeline, including data preprocessing (handling missing values and encoding categorical variables), feature selection, and model training and tuning based on validation performance.

Students must generate predictions for arabinoxylan content on the unseen data, evaluate model performance using the specified metric, visualize predicted versus actual values, and analyse the influence of environmental and management factors. Assessment emphasizes methodological rigor, reproducibility, and interpretability of results rather than leaderboard ranking, with a focus on understanding how agronomic and climatic variability affects crop fibre prediction.

4.6 Trumpf

Keywords: Production Data, Classification, Random Forest, XGBoost, LightGBM

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Machine learning task: Classification

Target variable: binary encoded version of "id_13013_RGT_erfolgreich_1try_mean"

1. Problem context and task definition

In industrial sheet metal production, accurately determining whether an extracted part can be successfully removed is crucial for maintaining product quality and process efficiency. Faulty extractions cause delays and increase costs due to rework or material waste. This challenge aims to develop a machine learning model that predicts the probability of successful extraction based on geometric properties, material characteristics, and cutting technology features.

The dataset originates from **TRUMPF**, a leading global supplier of technologies for sheet metal processing. Participants are tasked with building a binary classifier to predict whether an extraction will succeed or fail, enabling early detection of potential issues in manufacturing workflows.

2. Data and evaluation framework

The dataset contains **15,293 samples** with **84 anonymized features**, including both numerical and categorical variables (categorical features are postfixed with `__nom`). These describe part geometry (e.g., perimeter, area), position data, material properties, and cutting parameters.

Participants must:

- Preprocess data (encoding categorical variables, scaling numerical ones),
- Create the binary target label from `id_13013_RGT_erfolgreich_1try_mean`,
- Train and validate classification models,
- Optimize performance using recall as the main metric.

Evaluation metric:
The primary metric is **Recall**, defined as:

$$Recall = \frac{TP}{TP+FN}$$

where TP denotes true positives (correctly predicted successful extractions) and FN denotes false negatives (missed successful extractions). Recall is prioritized because correctly identifying failed extractions is more critical than predicting successes.

3. Difficulty level: Intermediate

This challenge requires intermediate-level skills in data preprocessing, feature engineering for mixed-type datasets (numerical + categorical), model training/validation for imbalanced classes, and metric-driven optimization.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union

Machine LearnAthon

4. Relevant micro-lectures (theoretical background)

- Installing Python
- Introduction to ML
- Data Visualization
- Data Preparation
- Feature Extraction
- Feature Selection
- Classification II
- Classification III
- Hyperparameter Tuning
- Overfitting
- Imbalanced Datasets
- Random Forest
- XGBoost
- LightGBM
- Evaluation metrics

5. Supporting tutorials (practical preparation)

- Introduction to Python tutorial
- Tutorial: AutoML frameworks FLAML and MLJAR

6. Programming languages, tools, and environments

Python is recommended — particularly libraries such as:

- **scikit-learn** for modeling,
- **pandas** & **numpy** for preprocessing,
- **matplotlib** & **seaborn** for visualization,
- Optional: **XGBoost**, **LightGBM**, or TensorFlow/Keras for advanced models.

R could also be used as an alternative environment.

7. Platforms and execution environment

The challenge is hosted on Kaggle — providing data access via:

Access to challenge data:

https://www.kaggle.com/datasets/6d881d7584d1beb7ab8d9418320a82c5dfd3221066f885e308d7e8b2322fb738

Submission format

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Submit a CSV file containing two columns: Index, Prediction

Leaderboards and evaluation

Submissions are evaluated based on *Recall*. The leaderboard ranks participants by their ability to correctly identify successful extractions while minimizing missed positives.

8. Learning focus and intended outcomes

Participants will learn how to:

- Handle mixed-type industrial datasets,
- Apply preprocessing pipelines suitable for real-world manufacturing data,
- Build robust binary classifiers capable of generalizing across unseen parts,
- Interpret results within a production-quality assurance context.

The outcome emphasizes practical ML deployment in manufacturing — turning predictive insights into actionable process improvements.

9. Example of course-level assessment using a challenge

For course-level assessment: Students receive an unseen dataset representing new parts from another production batch. They must design an end-to-end pipeline covering:

1. Data cleaning & encoding,
2. Feature selection/extraction,
3. Model training & validation using recall optimization,
4. Performance reporting including confusion matrix interpretation.

Grades reflect not only leaderboard performance but also code clarity, reproducibility of results, justification of chosen methods, and understanding of industrial implications in sheet metal processing quality control.

4.7 Graph neural networks for molecular property prediction

Keywords: Graph Neural Networks (GNNs), Molecular Property Prediction, Molecular Graphs, Boiling Point Prediction, Chemical Engineering, Regression

Machine learning task: Prediction of molecular boiling points.

Target variable: The normal boiling point of the molecule (in Kelvin).

1. Problem context and task definition

Boiling point prediction is a critical task in chemical and petrochemical engineering, particularly for the design and optimization of separation processes such as distillation.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Accurate predictions enable efficient solvent selection, reactor design, and process safety assessment while reducing reliance on costly and time-consuming experimental measurements. This challenge focuses on predicting molecular boiling points directly from chemical structure using graph neural networks.

2. Data and evaluation framework

Molecules are represented as graphs derived from SMILES strings. Each molecule is converted into a 2D topological molecular graph using RDKit, where atoms correspond to nodes and chemical bonds to edges. Node features include atomic number, hybridization state, and formal charge, while edge features include bond type. Model performance is evaluated using the Root Mean Squared Error (RMSE) metric.

3. Difficulty level: Intermediate-Advanced

The challenge requires advanced programming knowledge to set-up and develop graph and descriptor-based models.

4. Relevant micro-lectures (theoretical background)

Several micro lectures are relevant to this challenge, including Neural Networks (I-VIII), Overfitting, Introduction to ML, Data Visualization, Data Preparation, Feature Extraction.

5. Supporting tutorials (practical preparation)

Several tutorials can help students in practical preparation, including the tutorials on low code and no code regression.

6. Programming languages, tools, and environments

Python, PyTorch, R

7. Platforms and execution environment

The challenge is hosted and executed on Kaggle, providing a standardized environment for data access, model training, and submission.

Access to challenge data

The dataset is accessible directly through the Kaggle competition page.

Submission format

Participants must submit a CSV file containing predictions for the test set with the following columns: SMILES, Boiling point/K

Leaderboards and evaluation

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The challenge's success will be quantitatively measured using the **Root Mean Squared Error (RMSE)** metric to evaluate the performance of the regression models. Participants' models will be assessed based on their ability to correctly the boiling points in the test dataset. All results will be tracked on a live **Kaggle leaderboard**, providing real-time feedback and a competitive environment to compare model performance among participants. This leaderboard structure encourages iterative improvement and fosters the development of the most accurate and robust machine learning solution.

8. Learning focus and intended outcomes

The primary learning objective is to develop robust machine learning models capable of predicting molecular boiling points from chemical structure. Participants gain practical experience with molecular graph representations, graph neural networks, and regression evaluation metrics, addressing a real-world industrial problem in chemical engineering.

9. Example of course-level assessment using a challenge

For course-level assessment, students are evaluated on their ability to design complete modeling pipelines rather than solely achieving a high leaderboard score. A separate dataset is provided during the assessment, requiring students to apply preprocessing, graph construction, model selection, and performance evaluation using RMSE. This ensures students can generalize GNN-based molecular property prediction methods to unseen data.

4.8 Digitizing Hand-Drawn Unit Operations

Keywords: Process Flow Diagrams (PFDs), Hand-Drawn Symbols, Unit Operations, Chemical Engineering, Machine Learning, Digitization

Machine learning task: C**lassification** of hand-drawn unit operation symbols.

Target variable: The **category** (class) of the unit operation symbol (e.g., PFR reactor, valve, absorption column, CSTR reactor, compressor, pump, turbine, storage).

1. Problem context and task definition

Process Flow Diagrams (PFDs) are essential documents in the chemical process industry, containing vital information like process topology and major unit operations. However, PFDs (also called Engineering Drawings or EDs) are sometimes still drawn by hand, especially in early development stages, leading to a large amount of legacy EDs that are not machine-readable. The task is to develop robust machine learning models to accurately classify hand-drawn unit operation symbols into their respective categories, which is a crucial step for digitizing these legacy PFDs. Unit operations are the building blocks of chemical processes, used to transform raw materials or separate and purify components (e.g., distillation, filtration, mixing, reactions).

2. Data and evaluation framework

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

Participants will have access to a **diverse dataset** of hand-drawn unit operation symbols for training and validation. The models will be evaluated based on their ability to correctly classify symbols in a **test dataset**.

3. Difficulty level: Intermediate-Advanced

The challenge requires advanced programming knowledge to set-up and develop computer vision-based models.

4. Relevant micro-lectures (theoretical background)

Several micro lectures are relevant to this challenge, including Computer vision lectures (I-VII), Neural Networks (I-VIII), Overfitting, Introduction to ML, Data Visualization, Data Preparation, Classification (I-III).

5. Supporting tutorials (practical preparation)

Several tutorials can help students in practical preparation, including the tutorial on computer vision and the Low code classification tutorials.

6. Programming languages, tools, and environments

Python , PyTorch , R

7. Platforms and execution environment

The challenge is accessible via Kaggle.

Access to challenge data

The challenge is accessible via Kaggle.

Submission format

Submission format (csv, excel) with predicted classes per image.

Leaderboards and evaluation

The challenge's success will be quantitatively measured using the **Accuracy** metric to evaluate the performance of the classification models. Participants' models will be assessed based on their ability to correctly classify the hand-drawn unit operation symbols in the test dataset. All results will be tracked on a live **Kaggle leaderboard**, providing real-time feedback and a competitive environment to compare model performance among participants. This leaderboard structure encourages iterative improvement and fosters the development of the most accurate and robust machine learning solution.

8. Learning focus and intended outcomes

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union Machine LearnAthon

The primary learning outcome is to develop robust **machine learning models** capable of accurately classifying hand-drawn unit operation symbols. This provides practical experience in applying image classification techniques to address a real-world industrial need for digitizing legacy engineering drawings. Participants will gain familiarity with the visual representation and categories of **unit operations**—the building blocks of chemical processes—such as reactors, valves, and compressors. By successfully completing the challenge, they will have contributed to bridging the gap between historical **hand-drawn PFDs** and modern, machine-readable data representations. This work makes essential process information more accessible and efficient for chemical engineers and operators.

9. Example of course-level assessment using a challenge

For the course-level assessment, the challenge focuses on testing a student's ability to develop robust model pipelines, not just achieving a high score. Students will receive a **different dataset** of unit operation symbols during the exam. The assessment emphasizes the entire model development life cycle, including preprocessing, model selection, and utilizing evaluation metrics like Accuracy. This ensures students can generalize their modeling techniques to a novel problem, proving mastery of the core concepts behind digitizing engineering documents.

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.

Neither the European Commission nor the project's national funding agency DAAD are responsible for the content or liable for any losses or damage resulting of the use of these resources.

Co-funded by the European Union

Machine LearnAthon