FUTURE
STEM HUB

AI Methodological Guiding for Teachers

FUTURE-STEM-HUB

Empowering Secondary School STEM Education with AI Training and Resources for Students and Educators / FUTURE-STEM-HUB

Project No.
2024-1-DE03-KA220-SCH-000247346

Co-funded by
the European Union

The project “Empowering Secondary School STEM Education with AI Training and Resources for Students and Educators / FUTURE-STEM-HUB” (ref. no: 2024-1-DE03-KA220-SCH-000247346) is co-funded by the Erasmus+ Programme of the European Union. It is coordinated by the University of Duisburg-Essen (Germany) and involves four other partner organisations: M&M Profuture Training (Spain), the Kütahya Provincial Directorate of the Ministry of National Education (Turkey), COOPETAPE – Cooperative of Education, CRL – the governing body of ETAP School (Portugal) and Tetra Solutions Ltd. (Bulgaria).

The project team members, representing all partner organisations, are developing the FUTURE-STEM-HUB AI Methodological Guide for Teachers. Their aim is to provide school educators with coherent, structured and practical methodological guidance on AI, as well as a digital e-toolkit containing lesson plans, templates and ready-to-use teaching materials that enhance teachers’ confidence and competence when integrating AI into school education.

Authors:
Mustafa Bilgin, University of Duisburg-Essen (Germany)
Monica Moreno, M&M Profuture Training (Spain)
Montserrat Renedo, M&M Profuture Training (Spain)
João Barroso, ETAP School (Portugal)
Angelina Presa, ETAP School (Portugal)
Silviya Georgieva, Tetra Solutions Ltd. (Bulgaria)
Borislava Zaharieva-Tomova, Tetra Solutions Ltd. (Bulgaria)
Yeliz Yurter, Kütahya MEM (Turkey)
Özcan Turan, Kütahya MEM (Turkey)

Editor:
Monica Moreno, M&M Profuture Training (Spain)

Index of illustrations

Disclaimer and safety

The use of the programming code provided in these materials, as well as access to external websites and third-party platforms, is at the user's own risk and responsibility. The project team accepts no responsibility for the content, functionality or availability of external links, such as Google Colab, Scratch or Machine Learning for Kids. As many of the interactive tasks require an active internet connection and data processing on cloud servers, users are responsible for complying with security standards and system requirements (such as up-to-date browser versions).

When using third-party tools to create AI models, personal or sensitive information should not be uploaded. As the educational materials are aimed at young people aged 15 to 18, registration on external services and participation in online activities should be carried out in accordance with the terms of use of the respective providers and, ideally, under the supervision and with the consent of a teacher or legal guardian. No liability is accepted for any data loss or damage to end devices resulting from the execution of the code examples provided.

Contents

FUTURE-STEM-HUB

Project Overview

FUTURE-STEM-HUB project aims to advance and facilitate the integration of Artificial Intelligence (AI) topics into STEM education at secondary schools by: 1) Providing educational materials introducing AI concepts and their societal implications; 2) Offering practical learning resources for students to explore AI using Python programming and 3) Equipping teachers with support for integrating AI into secondary school STEM training.

Empowering students to understand, question and responsibly use Artificial Integ.

Project Outputs

1. **Course 1: A Digital Primer: Artificial Intelligence Essentials (Introduction to AI through Interactive Educational Materials for Secondary School Students)**

2. **Course 2: Delving Deeper into AI with Python and Scratch (Advanced AI: Hands-on Learning Materials for Secondary School Students)**

3. **E-Toolkit for School Educators: Enhancing Artificial Intelligence Skills (AI Methodological Guidance for Secondary Teachers)**

Introduction

Welcome to the AI Methodological Guide for Teachers. This guide aims to provide school educators with a coherent, structured and practical methodological framework for integrating Artificial Intelligence (AI) into secondary school education. It includes a digital E-Toolkit containing lesson plans, templates and ready-to-use teaching materials designed to enhance teachers' confidence and competence when introducing AI concepts in the classroom.

The specific objectives of this guide are:

- To support secondary school teachers in integrating Artificial Intelligence (AI) concepts and applications into STEM teaching and learning.
- To enhance teachers' confidence and competence in using AI-related educational resources, methodologies and classroom activities.
- To provide practical, ready-to-use lesson plans, templates and teaching materials that facilitate the implementation of AI education in secondary schools.
- To foster pupils' AI literacy, critical thinking, problem-solving and digital skills through engaging and meaningful learning experiences.
- To promote the ethical, responsible and informed use of AI technologies whilst encouraging reflection on their impact on society, education and future careers.

The toolkit comprises 12 lesson plans based on the educational resources developed in Course 1 and Course 2, designed to support the delivery of in-class training activities using the project's AI learning materials.

Each lesson plan sets out clear learning objectives and intended outcomes, prerequisites and learning context, guidelines for the classroom environment, instructional resources, structured learning activities (including direct instruction and guided practice), assessment methods and opportunities for reflection.

The materials presented in this guide will also be further developed into an online course to be hosted on the FUTURE-STEM-HUB platform and made available via the project website: https://www.future-stem-hub.eu

How to Use This Toolkit

This toolkit has been designed to support teachers in introducing concepts of Artificial Intelligence (AI) and practical AI-related activities into secondary school classrooms. The materials combine a conceptual understanding of AI with hands-on experimentation using simple programming tools and interactive classroom activities.
The lesson plans included in this guide are based on the educational resources developed in Course 1 and Course 2 of the FUTURE-STEM-HUB project.

- Course 1 materials introduce the foundations of Artificial Intelligence, including basic concepts, real-life applications and ethical considerations relating to AI systems.

- Course 2 materials provide more advanced and practical learning activities that allow students to explore AI through hands-on experiences such as Python notebooks, rule-based decision systems and interactive programming environments.

Empowering students to understand, question and use Artificial Intelligence responsibly.

How to Use This Toolkit

Together, these resources support a gradual progression from understanding how AI works to experimenting with simplified AI models and decision systems.

Each lesson plan follows a consistent structure to make the materials easy to understand and implement in different educational contexts. Teachers can use the lessons individually or combine them into a short learning sequence depending on their curriculum, available time and students' prior knowledge.

The lesson plans can be used:

- as standalone classroom activities introducing specific AI concepts or computational ideas,
- as a sequence of lessons forming a short AI learning module, or
- as complementary activities integrated into subjects such as ICT, STEM, mathematics, or digital literacy.

Each lesson plan includes:

- Learning objectives describing the expected learning outcomes.
- Materials and resources required for the activity.
- Step-by-step classroom procedures that guide teachers through the activity.
- Practical exercises and student outputs demonstrating student understanding.
- Assessment suggestions, such as quizzes, observation checklists or self-assessment activities.
- Adaptations for diverse learners to support different learning needs and abilities.

How to Use This Toolkit

Teachers are encouraged to adapt the activities to their classroom context and use them as a starting point for further exploration of Artificial Intelligence and its impact on society.

Teachers can use the lesson plans on their own or incorporate them into existing STEM subjects. The activities are designed to be flexible and adaptable to different classroom contexts, technological resources and pupils' abilities. The toolkit encourages experimentation, discussion and collaborative learning when introducing AI concepts.

Empowering students to understand, question and use Artificial Intelligence responsibly.

How to Use This Toolkit

Recommended use:

1. Read the methodological approach
2. Select a lesson plan based on topic
3. Follow the step-by-step implementation
4. Reflect and adapt

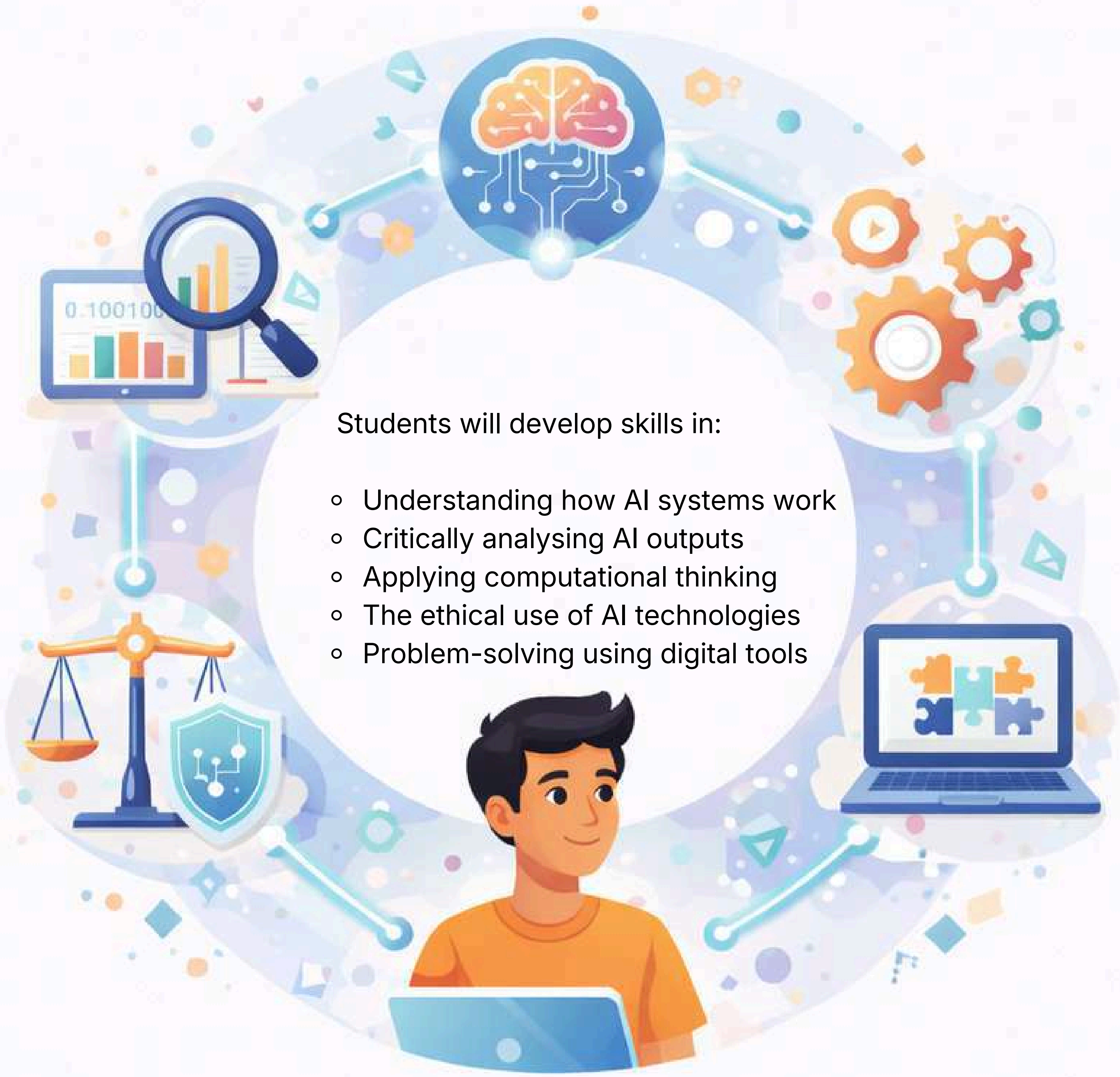
Structure of the FUTURE-STEM Lesson Plans

Each lesson plan follows a standard pedagogical structure to support teachers in delivering AI activities in the classroom:

- General Information
- Learning Objectives
- Skills Developed
- Prerequisites and Learning Context
- Classroom Environment
- Materials Required
- Step-by-Step Lesson Procedure
- Assessment and Reflection

This structure ensures consistency across all lesson plans and helps teachers to easily implement the activities in different classroom contexts.

AI Literacy Competences for Students



Students will develop skills in:

- Understanding how AI systems work
- Critically analysing AI outputs
- Applying computational thinking
- The ethical use of AI technologies
- Problem-solving using digital tools

Empowering students to understand, question and use Artificial Intelligence responsibly.

Methodological Approach for Teachers

This e-toolkit offers a practical and reflective introduction to the use of Artificial Intelligence in education. Through 12 lesson plans, participants will explore how to use our FUTURE-STEM resources to support curriculum planning, enhance student motivation, strengthen assessment practices, and promote ethical and responsible use in the classroom.

The lesson plans combine foundational concepts with hands-on activities using current AI tools, enabling educators to integrate AI as a complementary resource that enriches teaching and learning. This approach enables educators to integrate AI as a complementary resource that enriches teaching and learning whilst encouraging critical thinking about the role of technology in society.

The methodological approach adopted in this toolkit promotes active learning, collaboration and experimentation. Students are encouraged to explore AI concepts through guided activities, discussions and practical challenges that help them understand how AI systems work and how they influence everyday life.

There are many courses, workshops, seminars and other learning opportunities, both online and offline, that focus on the fundamentals of AI. There are also resources aimed at highly tech-savvy educators who have a background in AI concepts and the programming skills required to teach students how to code AI-based projects.

Empowering students to understand, question and use Artificial Intelligence responsibly.

However, when it comes to teachers who are themselves in the process of learning about AI, very little is available to help them translate what they are learning or already know into meaningful, student-led classroom activities. That is where the Hands-On FUTURE-STEM AI Lesson Plans e-toolkit comes in. Each lesson plan offers information and activity suggestions that teachers can use to ensure their students are given opportunities to engage in meaningful activities related to AI.

Hands-on learning activities help pupils move beyond theoretical explanations and develop a deeper understanding of how AI systems process data, generate predictions and influence decision-making. Through guided experimentation, pupils can test simple models, analyse results and reflect on the strengths and limitations of AI technologies.

Artificial Intelligence (AI) literacy in schools

AI literacy encompasses the technical knowledge, enduring skills and future-ready attitudes required to thrive in a world shaped by AI. It enables learners to engage with, create, manage and design AI, whilst critically evaluating its benefits, risks and ethical implications. This draft definition builds on existing definitions from the EU AI Act, the OECD, UNESCO and other organisations

Empowering students to understand, question and use Artificial Intelligence responsibly.

AI literacy encompasses the technical knowledge, enduring skills and future-ready attitudes required to thrive in a world shaped by AI. It enables learners to engage with, create, manage and design AI, whilst critically evaluating its benefits, risks and ethical implications. This draft definition builds on existing definitions from the EU AI Act, the OECD, UNESCO and other organisations

Recent research highlights the growing importance of developing AI literacy at school level. According to the AI Literacy Framework for Primary and Secondary Education developed by the European Commission and the OECD, AI literacy should help students understand how AI systems work, how they influence decision-making, and how individuals can interact with these systems responsibly in everyday life.

AI has become a widely used tool in everyday life, including for learning, personalised assistance and entertainment. Consequently, students must be able to understand how AI works, its impact and how to use AI ethically in order to be prepared for a society and economy in the age of AI.

However, the reality is that 49 per cent of 17- to 27-year-olds struggle to critically evaluate and identify AI's shortcomings, such as whether AI systems can invent facts (Merriman & Sanz Sáiz, 2024).

For this reason, integrating AI literacy into school education is becoming a key priority worldwide. As noted by Lidija Kralj, Independent Expert in AI and Data Education:
"Integrating AI literacy into education is essential to equip students with the critical thinking skills necessary to understand, interact with, and innovate using digital technologies, preparing them to contribute meaningfully to society"

Merriman, M., & Sanz Sáiz, B. (2024). How can we upskill Gen Z as fast as we train AI? Ernst & Young. https://www.ey.com/ en_us/about-us/corporate-responsibility/how-can-we-upskill-gen-z-as-fast-as-we-train-ai

Empowering students to understand, question and use Artificial Intelligence responsibly.

AI, Ethics and Critical Thinking

The intersection of AI, ethics and critical thinking is a vital area of focus as artificial intelligence becomes increasingly integrated into daily life, work and education. Although AI offers rapid data analysis and efficiency, there is a risk that it may undermine independent thought if users do not maintain a critical engagement with it, as highlighted by concerns that over-reliance on AI can affect analytical and decision-making skills.

Critical thinking remains a foundational skill in education, enabling students to make reasoned decisions, solve complex problems and evaluate diverse forms of information (Paul & Elder, 2019).

R. Paul, L. Elder. The Miniature Guide to Critical Thinking Concepts and Tools. Rowman & Littlefield (2019)
Pedagogical Coordinator for Computational Thinking and Artificial Intelligence at CEIBAL – Montevideo, Uruguay

Furthermore, artificial intelligence literacy frameworks emphasise that ethical considerations and critical thinking are central components of AI education. According to the OECD and European Commission's AI Literacy Framework, students must develop the ability to question AI-generated outputs, recognise potential bias or misinformation, and decide whether AI results should be accepted, revised or rejected.

AI literacy therefore combines technical understanding with critical evaluation and responsible attitudes, helping learners recognise both the opportunities and limitations of AI systems and understand their potential impact on society.

As Victor Koleszar said: "AI literacy should be part of every learner's journey, with educators playing a key role in making AI meaningful. By helping students understand and engage with AI in ways that reflect their local context, we can support more inclusive, future-ready education systems around the world."

Empowering students to understand, question and use Artificial Intelligence responsibly.

Lesson Plan Collections

INTRODUCTION

The following section presents the collection of Artificial Intelligence lesson plans developed as part of the FUTURE-STEM-HUB project.

Each module developed in these work packages has been adapted into a classroom-ready lesson plan designed to support teachers in delivering Artificial Intelligence activities with secondary school pupils.

Course 1 focuses on introducing fundamental AI concepts, real-life applications and ethical considerations, whilst Course 2 provides more advanced and hands-on learning activities using programming environments such as Python and Scratch.

Each lesson plan is designed to support secondary school teachers in introducing Artificial Intelligence concepts through structured and engaging classroom activities. The lessons combine conceptual understanding with hands-on experimentation using simple AI tools and programming environments such as Python notebooks, Scratch or other interactive platforms.

All lesson plans follow a common pedagogical structure to ensure consistency across the toolkit and to facilitate implementation in different educational contexts. Teachers may use the lessons as standalone activities or combine them into a short AI learning module depending on their curriculum and pupils' prior knowledge.

Overview of Lesson Plans

Lesson	Topic	Level	Teaching Approach	Duration	Competences Developed
Lesson 1	Introduction to Artificial Intelligence	Beginner	Discussion based learning	60 min	AI awareness, critical thinking
Lesson 2	Machine Learning & Neural Networks	Beginner–Intermediate	Visual explanations & examples	60 min	Analytical thinking, understanding ML basics
Lesson 3	AI Applications across Industries	Beginner	Case studies	60 min	Real-world connection, problem-solving
Lesson 4	Cognitive Computing & AI Technologies	Intermediate	Conceptual eploration & examples	60 min	Digital competence, AI literacy
Lesson 5	Ethical Considerations & Future of AI	Intermediate	Debate & Reflection	60 min	Ethics, critical thinking
Lesson 6	Introduction to Neural Networks (Story-based)	Beginner	Storytelling approach	45–60 min	Conceptual understanding, engagement

Overview of Lesson Plans

Lesson	Topic	Level	Teaching Approach	Duration	Competences Developed
Lesson 7	How Machines Learn	Beginner	Visual & conceptual learning	60 min	Logical thinking, data understanding
Lesson 8	Neural Networks in Practice	Intermediate	Guided coding activity	60–90 min	Practical skills, experimentation
Lesson 9	AI Escape Room (Calculator Challenge)	Intermediate	Gamification	60–90 min	Collaboration, problem-solving
Lesson 10	Scratch Meets Artificial Intelligence	Intermediate	Scratch programming	90 min	Computational thinking, creativity
Lesson 11	Python Game Programming	Advanced	Python (pygame/Arcade)	90–120 min	Coding, algorithmic thinking
Lesson 12	Object Recognition with Roboflow	Advanced	Computer vision project	90–120 min	AI application, data skills

Course 1: A Digital Primer: Artificial Intelligence Essentials

Empowering Secondary School STEM Education with AI
Training and Resources for Students and Educators
2024-1-DE03-KA220-SCH-000247346

Lesson Plan 1: Artificial Intelligence – From Automata to Modern AI Systems

1. General Information

Course and Module:	Course 1: A Digital Primer: Artificial Intelligence Essentials Module 1: Introduction to Artificial Intelligence
AI Tool / Technology Used:	Search engines, generative AI tools (e.g. ChatGPT),
Target Age Group:	15–18 years (Secondary education).
Subject Area:	ICT, Technology, STEM
Estimated Duration:	90 minutes (2 sessions)

2. Learning Objectives (Outcomes)

By the end of this course, students will be able to:

- Explain the basic definition of artificial intelligence.
- Summarize the historical development of artificial intelligence.
- Distinguish between expert systems, machine learning, neural networks, and deep learning.
- Give examples of the use of artificial intelligence in daily life.
- Discuss the ethical and social impacts of artificial intelligence.

This lesson focuses on building foundational AI literacy by introducing students to the origins, evolution, and key concepts of Artificial Intelligence. Through interactive learning activities and real-world examples, students develop a clear understanding of how AI systems have evolved, where they are used today, and how they influence everyday life and future technological developments.

3. Competences Developed

- Digital competence – Understanding the fundamental principles of Artificial Intelligence and recognising its applications in everyday digital environments.
- Critical thinking – Analysing the benefits, limitations and ethical implications of AI technologies.
- Problem-solving – Exploring how AI can be applied to solve real-world challenges across different sectors.
- Communication and collaboration – Participating in discussions, collaborative activities and ethical debates about AI.
- Creativity – Developing ideas and proposing innovative applications of AI through scenario-based activities.
- Ethical awareness – Reflecting on the social impact of Artificial Intelligence and recognising the importance of using AI responsibly, fairly and transparently.

4. Prerequisites and Learning Context

4.1 Prior knowledge required

Students should:

- Possess basic digital literacy skills, including the ability to use computers, tablets or mobile devices.
- Be familiar with searching for information using web browsers or search engines.
- Have basic experience participating in classroom discussions and collaborative activities.
- No prior knowledge of Artificial Intelligence is required, as this lesson introduces the topic from a foundational level.

4.2 Learning Context

This lesson is designed as the introductory session of Course 1: A Digital Primer: Artificial Intelligence Essentials and provides the conceptual foundation for all subsequent lessons.
Students explore how Artificial Intelligence has evolved over time, discover its presence in everyday life and begin to understand its growing influence on education, society and future careers. The lesson also introduces key ethical considerations that will be revisited throughout the course.

5. Classroom Environment

The classroom should support interactive and collaborative learning, with flexible seating arrangements that allow for group work and discussions. Access to a projector or smart board is recommended for visual presentations, videos, and demonstrations. If available, internet-connected devices can be used by students to explore AI tools and conduct guided research activities. A safe and inclusive environment should be maintained to encourage open dialogue, especially during ethical discussions.

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:
- Prepare the presentation slides.
- Check internet access for online demonstrations.
- Prepare any videos or digital resources used during the lesson.
- Print worksheets or matching cards if required.
- Familiarise themselves with the discussion prompts and ethical scenarios.

5.2 Classroom Setup

Students should work individually during introductory activities and in pairs or small groups during discussions and collaborative tasks.
The classroom should allow easy interaction between groups while enabling the teacher to facilitate whole-class discussions.

5.3 Techinical requeriments

Hardware
- Computer connected to a projector or interactive whiteboard.
- Student devices (optional).

Software
- Modern web browser.
- Presentation software.
- Internet access.
- Optional access to generative AI tools such as ChatGPT or similar platforms for demonstration purposes.

6 Material Needed

6.1 Digital Resources

- Presentation slides.
- Short introductory AI video.
- Online timeline of AI history.
- Optional demonstration using a generative AI application

6.2 Physical Material (optional)

- AI Timeline worksheet.
- Matching activity cards.
- Ethical discussion cards.
- Reflection worksheet.

7. Lesson Procedure (Step-by-Step Flow)

7.1 Main Activity

Section 1: Warm-up and Theoretical Foundations (10 Minutes)

Discussion Question: Can machines think?
Students discuss decision-making technologies in daily life (recommendation systems, navigation, voice assistants)

Discussion: "Is the phone in your hand intelligent, or just a very fast calculator?"
Definition of the phone: The OECD definition is given in a simplified way: "A system that perceives its environment, makes predictions, and makes decisions in line with its goals."

Game: 3 people are selected from the class. One is the "Inquirer", one is the "Human", and one is the "Atlas (AI)".

Process: The Inquirer asks questions to the two people behind the curtain (What is your favorite food? What is 154 x 23?).

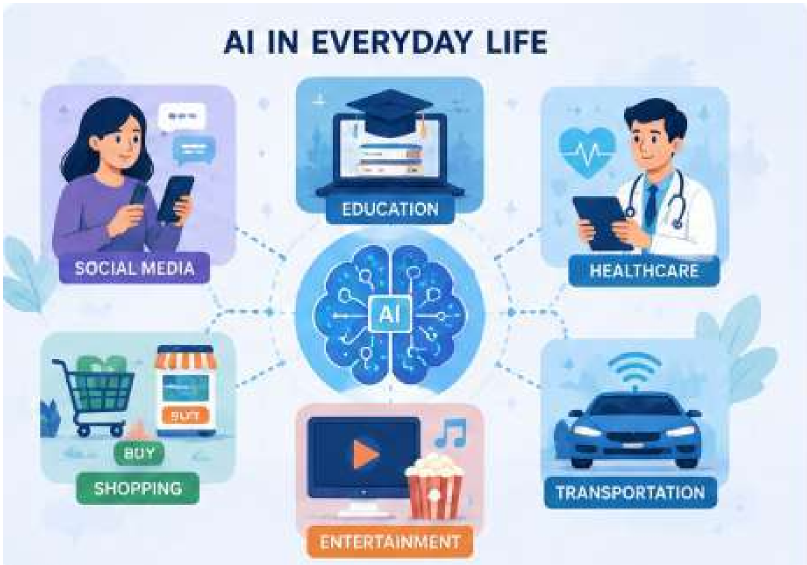
Outcome: To experience Alan Turing's question "Can machines think?" and the Turing Test.

Section 2. What is Artificial Intelligence? (10-15 minutes)

The teacher encourages the students use AI to answer the question. Artificial intelligence is defined as systems that perceive, analyze, and make decisions about the environment.

Emphasis: Artificial intelligence = algorithm + data + model.

Section 3. Historical Development (15 minutes)
The teacher encourages the students use AI to find information about the topic listed below:
1. Mechanical automata in ancient times
2. Charles Babbage and the Difference Engine
3. Alan Turing and the Turing Test
4. Dartmouth Conference (1956)
5. Deep Blue defeating Garry Kasparov (1997)



Section 4. Types of Artificial Intelligence Activity (25 minutes)

Match the AI Types with their real-life samples
1. Netflix recommendations a. Expert System / Search Algorithms
2.Chess program b. Deep Learning
3.Voice assistants c. Machine Learning
4.Face recognition d.Machine Learning + Natural Language Processing

Matching activity:

Students match real-life examples with the appropriate AI type:
- Netflix recommendations = Machine Learning
- Chess program =Expert System / Search Algorithms
- Voice assistants = Machine Learning + Natural Language Processing
- Face recognition = Deep Learning

Young Engineers Activity:

"Young Engineers, today we found an old chest in our lab. Inside was a prototype assistant from the 1950s named Atlas. Atlas's memory is a little muddled. He can't remember whether he's just a 'calculator' or a 'thinking intelligence'. We need to take a time travel from Alan Turing to the present day and help Atlas update his 'brain'. Are you ready?"

Student Worksheet

HOW TO UPDATE ATLAS!

Engineer Name: ....

A. Intelligence Test: Rule or Learning?

Examine the following situations. Write [R] if the system only follows the given rules, [L] if it learns from the data and improves itself.
( ) A refrigerator beeping when the door is left open.
( ) Spotify creating a new playlist for you based on your favorite songs.
( ) Traffic lights turning red every 60 seconds.
( ) An autonomous vehicle noticing a pedestrian on the road and stopping.

B. Atlas's Time Tunnel

Arrange the following events from oldest to newest according to the information:
( ) AlphaGo defeating the Go champion.
( ) Alan Turing's "Computing Machinery and Intelligence" paper.
( ) IBM Watson winning the quiz show.
( ) The term "Artificial Intelligence" being used for the first time.

C. Critical Thinking: What if Atlas Came to the Classroom?

If Atlas were a student sitting next to you in class, how would you know he was a robot? (Write a hint):
....

Section 5. Ethical Discussion (20 min):

Discussion: Is artificial intelligence an opportunity or a threat?
Students are divided into two groups randomly. Group 1 defends the idea "AI is an opportunity". Group 2 defends "AI is a treat". The teacher gives 5 minutes to collect data for the students. Later the group discussion is performed under the moderation of the teacher.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Practical Exercise / Student Output (10 minutes)

Expected Student Outputs
By the end of the lesson, each student (or pair of students) should produce:

- A completed AI Knowledge Check demonstrating their understanding of the fundamental concepts introduced during the lesson.
- A completed Exit Ticket reflecting on the benefits, risks and personal learning outcomes related to Artificial Intelligence.
- Active participation in the class discussion by sharing ideas about the ethical and social implications of AI.

8.2 Informal Success Criteria

Students successfully achieve the lesson objectives if they can:
Explain the basic concept of Artificial Intelligence using their own words.

- Describe the purpose of the Turing Test.
- Explain the difference between Artificial Intelligence, Machine Learning and Deep Learning.
- Recognise Artificial Neural Networks as the core structure used in Deep Learning.
- Identify Deep Blue as one of the historical milestones in the development of Artificial Intelligence.
- Participate actively in discussions about the opportunities and challenges of AI.

Note for Teachers

Encourage students to explain their reasoning rather than simply providing correct answers. Use open-ended questions to stimulate discussion, promote critical thinking and create a classroom environment where different perspectives are respected. Remind students that understanding how AI works is more important than memorising technical terms, and encourage curiosity, experimentation and collaborative learning throughout the lesson.

8.3 Assessment

Assessment is formative and takes place throughout the lesson.
It combines:

- the AI Knowledge Check;
- teacher observation;
- student self-assessment (Exit Ticket).

AI Knowledge Check
Answer the following questions:

1. What is Artificial Intelligence?

....

2. What does the Turing Test measure?

....

3. What is Machine Learning?

....

4. What type of structure does Deep Learning use?

………………………………………………………………………………………

5. Why is Deep Blue considered an important milestone in the history of Artificial Intelligence?

………………………………………………………………………………………

Teacher Observation Checklist

During the lesson, the teacher observes whether students:

- Participate actively in discussions and collaborative activities.
- Correctly explain the fundamental concepts introduced during the lesson.
- Distinguish between Artificial Intelligence, Machine Learning and Deep Learning.
- Recognise historical milestones such as the Turing Test and Deep Blue.
- Express informed opinions about the ethical implications of Artificial Intelligence.
- Collaborate respectfully with classmates and justify their ideas using examples.

These observation points may be recorded using a simple checklist prepared by the teacher.

Student Self-Assessment (Exit Ticket)

At the end of the lesson, students complete a short self-assessment individually.

Write one benefit of Artificial Intelligence.

………………………………………………………………………………………

Write one potential risk of Artificial Intelligence.

………………………………………………………………………………………

The most surprising thing I learned today was…

………………………………………………………………………………………

Question of the Day:

Do you think Atlas could have feelings, or is it simply processing information and performing mathematical calculations? Explain your answer.

....

The Exit Ticket may be completed on paper, in a shared digital document or through an online form. It provides students with an opportunity to reflect on their learning while helping teachers identify misconceptions and plan future lessons.

9. Adaptations for Diverse Learners

Additional Support

- Provide a printed or digital glossary with the key concepts introduced during the lesson (e.g., Artificial Intelligence, Machine Learning, Deep Learning, Neural Networks, Turing Test, Expert Systems).
- Use visual supports such as an AI timeline, concept maps or illustrated examples to reinforce students' understanding.
- Provide partially completed worksheets or guided note-taking templates to help students organise key ideas.
- Allow students to work in pairs or small groups during discussions and matching activities.

For Students with Lower Digital Confidence

- Pair students with more experienced classmates during digital activities.
- Provide step-by-step instructions for accessing online resources or AI demonstrations.
- Use familiar real-life examples (e.g., voice assistants, recommendation systems, navigation apps) before introducing more abstract AI concepts.
- Reduce the number of concepts introduced at one time and reinforce learning through short recap activities.

Fast Finishers

- Research an additional milestone in the history of Artificial Intelligence and explain why it was significant.
- Find another example of AI used in everyday life and present it to the class.
- Compare two AI systems (e.g., ChatGPT and a voice assistant) and identify similarities and differences.
- Write a short paragraph discussing one potential future application of Artificial Intelligence and one ethical challenge it may create.

10. Resources / Attachments

The following resources support the implementation of this lesson:

Digital Resources

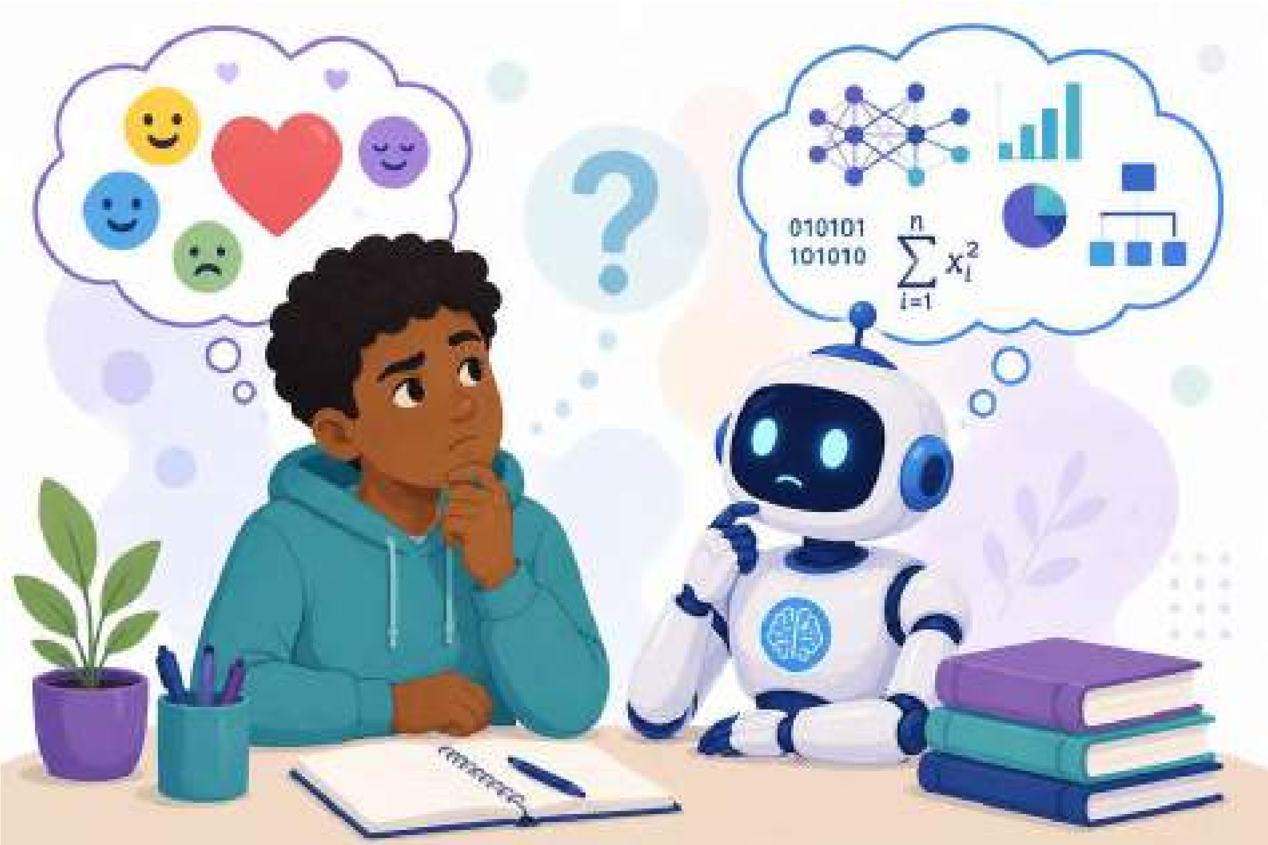
- Presentation slides: Artificial Intelligence – From Automata to Modern AI Systems.
- Search engines for guided research activities.
- Generative AI tools (e.g., ChatGPT) for teacher demonstration, where appropriate.
- Short introductory video on Artificial Intell0nce (optional).

Classroom Materials

- AI Timeline activity.
- AI Concepts Matching Activity.
- Knowledge Check questions.
- Exit Ticket.
- Whiteboard or interactive whiteboard.
- Student worksheets (if printed).

Teacher Resources

Teachers are encouraged to adapt the lesson using age-appropriate videos, online resources and real-life examples of Artificial Intelligence that are relevant to their students and local curriculum.



Lesson Plan 2: AI Core Concepts: Machine Learning, Deep Learning and Neural Networks

1. General Information

Course and Module:	Course 1: A Digital Primer: Artificial Intelligence Essentials Module 2: AI Core Concepts – From Machine Learning to Neural Networks
AI Tool / Technology Used:	Teachable Machine (Google) – simple image/pose classification model
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Computer Science / STEM
Estimated Duration:	60 minutes (optional extension up to 90 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Explain in simple terms the differences between AI, Machine Learning (ML), Deep Learning (DL) and Neural Networks (NN).
- Describe and give one example of supervised, unsupervised and reinforcement learning.
- Identify the basic components of a simple neural network (input, hidden layers, output, weights, bias, activation).
- Train and test a simple image based model using Teachable Machine with a small set of examples.
- Reflect on how data quality and diversity affect the accuracy and fairness of AI models.

This lesson focuses on building foundational AI literacy by helping students understand the core concepts behind Artificial Intelligence and Machine Learning, and by engaging them in training and testing a simple image-based model.

3. Competences Developed

- Digital competence: Using a web based AI tool (Teachable Machine), handling images via webcam or files.
- Problem solving: Experimenting with training data, testing predictions and adjusting inputs to improve results.
- Critical thinking: Discussing limitations, bias and over reliance on imperfect AI models.
- Collaboration: Working in pairs to design, train and document a simple AI model.
- Creativity: Designing meaningful classes (gestures, objects, emotions) and exploring real world applications.

4. Prerequisites and Learning Context

4.1 Prior knowledge required

- Basic digital skills: using a browser, webcam and file uploads.
- Very simple prior understanding of "what AI is" (Modules 1 or previous introductory lesson), but no technical ML knowledge is required.
- No prior coding experience is necessary; the tool hides the underlying code.

4.2 Learning Context

This lesson is part of Course 1 "A Digital Primer: Artificial Intelligence Essentials", Module 2 – AI core concepts.
It connects conceptual explanations (AI vs ML vs DL vs NN, types of ML) with a hands-on activity using Teachable Machine, helping students see how these ideas appear in real tools.
The lesson can be used as:
I A follow-up to an introductory AI literacy session (e.g., Aylin & Alara – Module 1).
I A bridge towards more technical modules in Course 2 (Python, Scratch-based AI).

5. Classroom Environment

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:
- Ensure that all computers have a modern browser and working webcam (or that students can upload images if webcams are not available).
- Test access to Teachable Machine: https://teachablemachine.withgoogle.com/
- Decide on the recommended project type (Image Project) and a simple three-class example (e.g., Rock–Paper–Scissors hands, three facial expressions, or three objects available in the classroom).

u Prepare a short slide deck or visual support summarising:
u AI vs ML vs DL vs NN
u Supervised / unsupervised / reinforcement learning (with 1–2 examples each).

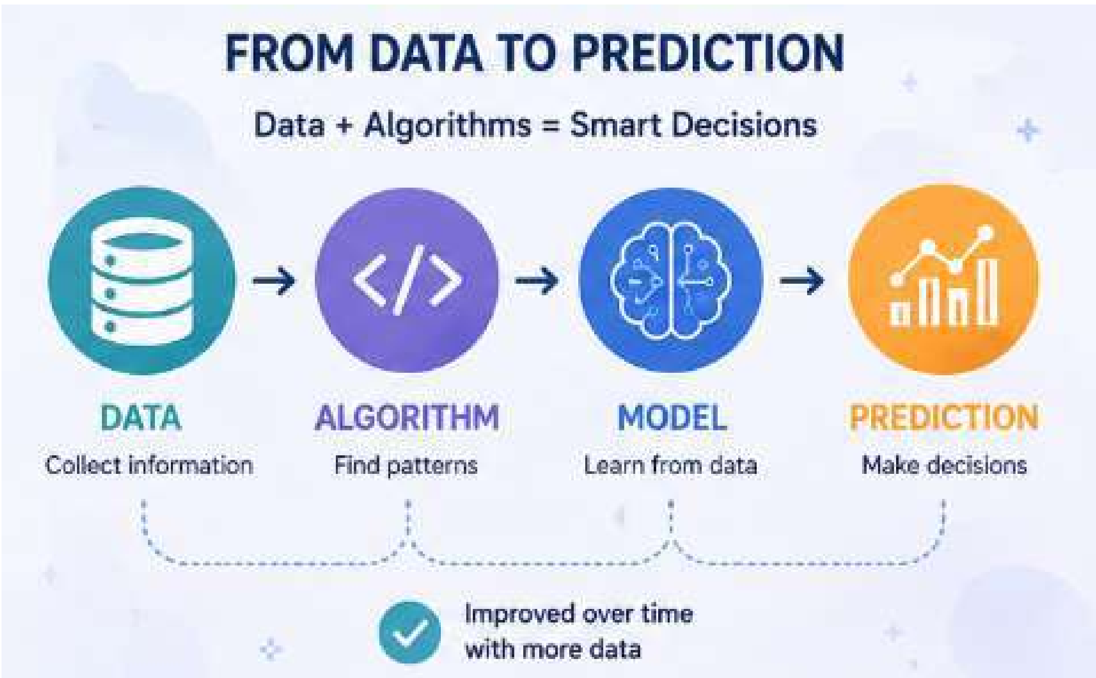
- Check that school internet is stable enough for webcam-based training or image uploads.
- (Optional) Prepare a teacher demo model in Teachable Machine to show expected results and common issues.

5.2 Classroom Setup

Students work mainly in pairs, sharing one device, to promote collaboration and discussion.
Pair programming roles are used:

- Driver: operates the mouse/keyboard and interacts with Teachable Machine.
- Navigator: reads instructions, suggests examples and helps with decisions.

Roles are swapped at least once during the activity (after training the first version of the model).



5.3 Technical requirements (for in-class activities)

Hardware

Computers or laptops with webcam (preferred) or ability to upload image files.

Projector or interactive whiteboard for teacher explanations and demos.

Software

Python Modern web browser (Chrome, Edge, Firefox, etc.).

Teachable Machine (Image Project):

https://teachablemachine.withgoogle.com/

Internet connection

Required throughout the lesson for accessing Teachable Machine and, optionally, for showing video examples or additional resources.

Backup plan for classes without webcams

If webcams are not available or reliable, students can:

- Upload existing images provided by the teacher (e.g. from a shared folder or open image datasets).
- Use simple icons or drawings saved as image files instead of live camera input.
- Work with a pre prepared Teachable Machine project where images are already uploaded and focus on testing, adding new images, and retraining the model.

6. Materials Needed

6.1 Digital Resources

- Module 2 – AI Core Concepts: Machine Learning, Deep Learning and Neural Networks (student-facing text).
- Slide deck or short handout with:
- Definitions of AI / ML / DL / NN.
- Types of ML: supervised, unsupervised, reinforcement (with examples from music apps, recommendation systems, games).
- Teachable Machine link and short step-by-step screenshots (optional).

6.2 Physical Materials (optional)

- Printed summary sheet of key terms (AI, ML, DL, NN, supervised, unsupervised, reinforcement, neural network parts).
- Printed exit ticket or reflection sheet (Section 9.3).

7. Step-by-Step Lesson Procedure

The timing below is indicative for a 60 minute lesson. Teachers may shorten or extend individual parts depending on student needs and available time.
Warm-Up / Introduction (10 minutes)

Teacher actions

Ask the opening question: “Where do you think AI is used in your daily life?”
Possible prompts: music apps (Spotify), video platforms (YouTube, Netflix), social media feeds, navigation apps, voice assistants.

Collect a few examples from students and quickly connect them to:

- recommendation systems,
- image and speech recognition,
- prediction and decision making.

Present the goal of the lesson in simple language:
"Today we will understand the core ideas behind AI: what machine learning, deep learning and neural networks are, and we will create a small AI model ourselves using Teachable Machine."

Key terms (introduced in simple language)

- Artificial Intelligence (AI) – machines performing tasks that usually require human intelligence (learning, recognising patterns, making decisions).
- Machine Learning (ML) – AI systems that learn patterns from data and make predictions instead of following only fixed rules.
- Deep Learning (DL) – a powerful type of ML using neural networks with many layers to handle complex data (like images or sound).
- Neural Network (NN) – a network of simple units (neurons) connected with weights, inspired by the brain.

Class organization: Whole class discussion, then transition to students working in pairs at computers.

7.2 Main Activity – AI in Action (60 minutes total)

Part 1 – Mini input: AI, ML, DL, NN and types of learning (10 minutes)

Teacher input
Using slides or the module text, briefly explain:
AI vs ML vs DL vs NN

- AI: “teaching machines to think and solve problems.”
- ML: “learning from examples (data) instead of only rules.”
- DL: “using neural networks with many layers for complex tasks (images, speech).”
- NN: input layer, hidden layers, output layer; weights, bias, activation function.

Three types of ML (with one example each):

- Supervised learning – learning from labelled data (e.g., spam/not spam in email, cat/dog images).
- Unsupervised learning – finding groups or patterns without labels (e.g., grouping music listeners by habits for playlists).
- Reinforcement learning – learning by trial and error with rewards and penalties (e.g., game bots learning to win, self-driving cars).

Keep explanations short and focus on intuitive examples, not formulas.

Check-in question (whole class):

- Can someone give an example of supervised learning in real life?”
- “Where might reinforcement learning be useful in a game?”

Part 2 – Teachable Machine Demo (5 minutes)

Teacher actions

- Open https://teachablemachine.withgoogle.com/ on the projector.
- Select Image Project → “Standard image model”.
- Show the structure:

Class 1, Class 2, Class 3 (e.g., “Rock, Paper, Scissors” or three facial expressions).
Demonstrate quickly:
Recording a few images per class with the webcam or uploading example images.
Clicking “Train model” and then testing live with the webcam.

- Emphasise that students will now create their own small models in pairs.

Part 3 – Student Activity: Train your own model (25 minutes)

Objective: Students work in pairs to create, train and test a simple image classification model with three classes using Teachable Machine.
Suggested class themes (teacher chooses one for the whole class or lets pairs decide):

- Rock / Paper / Scissors (hand gestures).
- Happy / Neutral / Sad (facial expressions).
- Three classroom objects (e.g., pen / notebook / bottle).
- Student tasks (in pairs)
- Open Teachable Machine and create an Image Project.
- Rename the three classes according to the chosen theme.
- For each class, collect multiple examples:
- 10–20 images per class (different angles, lighting, background and people if possible).
- Train the model by clicking the "Train" button.
- Test the model using the live webcam preview:
- Try each class multiple times.
- Observe which predictions are correct and where the model is confused.
- Make at least one change to improve the model:
- Add more diverse examples for a class that performs poorly.
- Remove noisy images or extremely similar duplicates.
- Retrain and test again.

Guiding questions while they work
"Does your model perform better when you add more varied images?"
"What happens if the background changes or the lighting is worse?"

Teacher role

Circulate between pairs:
Support technical issues (camera permissions, slow training).
Encourage students to think about data diversity and bias (e.g., only one person's hand vs many hands).
Remind pairs to swap roles (Driver/Navigator) after the first training round.

Optional extension (for fast finishers)

Add a fourth class (e.g., "Nothing / Background" or another gesture/object) and retrain the model.
Compare accuracy before and after adding the new class.

7.3 Reflection / Discussion (10 minutes)

Teacher-led whole-class discussion
Use these questions to structure the reflection:
"What did your model do well? In which situations did it fail?"
"How did the amount and diversity of training data influence the accuracy?"
"If we only used images from one person or one skin tone, would the model work equally well for everyone?"
"Where could a tool like Teachable Machine be useful in real life (education, games, accessibility, etc.)?"

Key takeaway

Highlight 2–3 main ideas:

Machine learning models learn from examples, not from explicit rules.
Data quality and diversity are essential for fair and accurate AI.
Neural networks and deep learning are powerful, but they are not magic; they can make mistakes and reflect the bias of their training data.

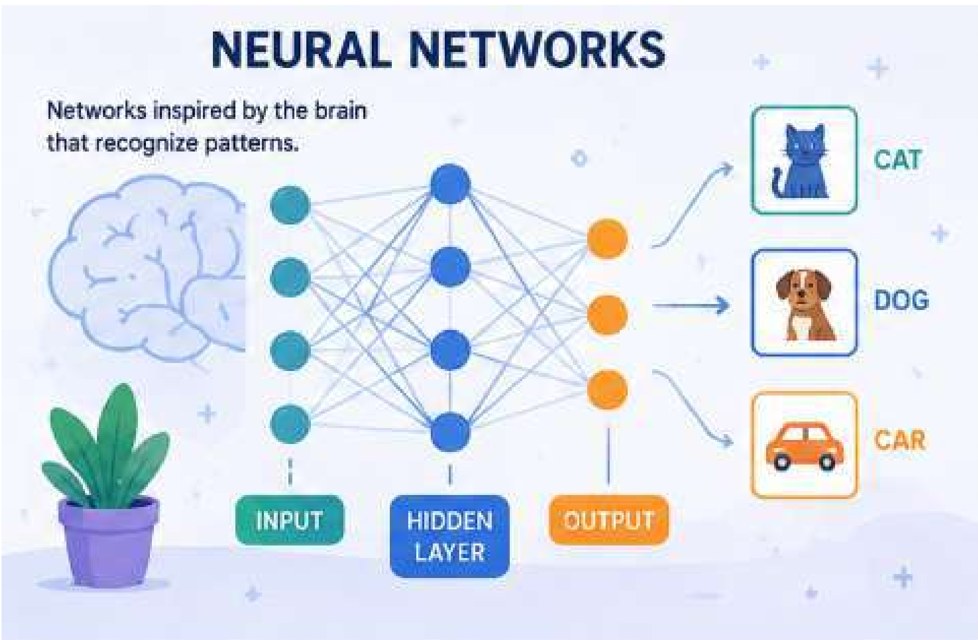
Note for Teachers:

Students may initially confuse Machine Learning, Deep Learning and Neural Networks. Encourage them to focus on understanding the relationships between these concepts rather than memorising definitions. Use simple analogies, visual representations and real-world examples to reinforce understanding, and remind students that AI systems learn from data provided by humans, making the quality and fairness of that data essential. Encourage questions and collaborative discussion throughout the lesson.

8.Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.



8.1 Practical Exercise / Student Output (10 minutes)

Expected Student Outputs

By the end of the lesson, each pair should produce:

- A trained Teachable Machine image model with at least three classes and a visible live test demonstrating correct predictions for most cases.
- A short-written note (3–4 sentences) answering:

What classes did you use in your model?

What did you change to improve the model (more images, different angles, etc.)?

In which situations did the model make mistakes?

This written note can be done on paper or in a shared digital document.

8.2 Informal success criteria

The model runs and can correctly recognise the three classes in most test attempts.

Students can explain in simple language:

What their model learned from the data.

Why adding more varied examples can improve performance.

Students can give at least one realistic use case for similar models in the real world.

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.

It combines:

- a short quiz,
- teacher observation,
- student self-assessment (exit ticket).

Quick Quiz (5 questions)
Use or adapt the following multiple-choice questions derived from Module 2.

What is the primary goal of Artificial Intelligence (AI)?
a) To perform tasks requiring human-like intelligence, such as learning, decision-making or pattern recognition.
b) To replicate the human brain exactly.
c) To make computers faster and more powerful.
d) To analyse and store large amounts of data.
Correct answer: a).

How does Machine Learning differ from traditional programming?
a) ML uses predefined rules to make decisions, while traditional programming learns from data.
b) ML learns patterns from data and makes decisions, while traditional programming follows step-by-step rules written by humans.
c) ML uses only numerical data, while traditional programming uses images and text.
d) There is no difference between ML and traditional programming.
Correct answer: b).

Which of the following is an example of supervised learning?
a) Grouping music listeners into categories based on their preferences.
b) Teaching a robot to navigate a maze through trial and error.
c) Training a spam filter using labelled examples of “spam” and “not spam” emails.
d) Finding clusters of similar customer behaviours without using labels.
Correct answer: c).

What is the main difference between supervised and unsupervised learning?
a) Supervised learning uses labelled data, while unsupervised learning does not.
b) Supervised learning finds hidden patterns, while unsupervised learning predicts outcomes.

c) Supervised learning uses numerical data, while unsupervised learning uses text data.
d) There is no difference; they are the same.
Correct answer: a).

What is Deep Learning?
a) A method of teaching machines using only labelled data.
b) A type of AI that uses Artificial Neural Networks (ANNs) to process large amounts of complex data.
c) A tool for analysing simple datasets with limited variables.
d) A form of reinforcement learning for solving reward-based problems.
Correct answer: b).

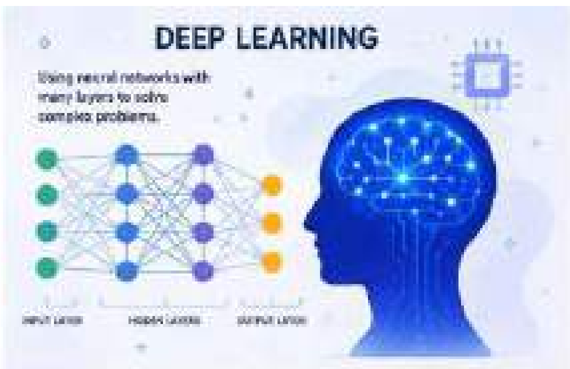
The quiz can be done on paper, in a shared document or using online tools (e.g., Kahoot, Google Forms).

Teacher Observation Checklist

During the main activity, the teacher observes each pair of students and checks whether they:

- Can access and use Teachable Machine without major difficulties.
- Create and name at least three classes in their model.
- Collect multiple examples per class and understand why variety matters.
- Test the model and identify at least one situation where it fails.
- Can explain one example of AI/ML from everyday life and connect it to supervised, unsupervised or reinforcement learning.

These observation points can be recorded using a simple checklist (see Annex 1 of this lesson plan for a ready to use template).



Student Self-Assessment (Exit Ticket)

At the end of the lesson, students complete a short self-assessment (on paper or digitally):

"In my own words, AI is…"
"One difference between traditional programming and machine learning is…"

Rate yourself from 1 to 5:
"I can explain the difference between AI, ML, DL and neural networks."
1 □ 2 □ 3 □ 4 □ 5 □

"I can describe at least one type of machine learning (supervised, unsupervised or reinforcement) with an example."
1 □ 2 □ 3 □ 4 □ 5 □

"One limitation or risk I noticed about AI in our activity today is…"

9. Adaptations for Diverse Learners

Additional support

- Provide a printed or digital glossary with key terms (AI, ML, DL, NN, supervised, unsupervised, reinforcement).
- Offer a partially prepared Teachable Machine project with classes already created; students only need to add images and train.
- Allow students to work in groups of three if some need extra guidance.

For students with lower digital confidence

- Pair them with more experienced classmates.
- Prepare step-by-step screenshots of the Teachable Machine workflow.
- Focus on one simple scenario (e.g., rock/paper/scissors) instead of allowing free choice.

Fast finishers

- Add an extra class and retrain the model; compare overall performance.
- Experiment with changing backgrounds, lighting or different people to test robustness.
- Ask them to write a short paragraph describing a realistic application of their model and potential ethical concerns (e.g., bias, misuse).

10. Resources / Attachments

- Module 2 – AI Core Concepts: Machine Learning, Deep Learning and Neural Networks (Course 1 educational material).
- Teachable Machine – https://teachablemachine.withgoogle.com/
- Printable documents (optional, as annexes):
- Teacher Observation Checklist (pairs).
- Student Exit Ticket / Self-Assessment.

Annex 1 – Teacher Observation Checklist

Use this checklist to observe student pairs during the lesson. Tick (✓) when the criterion is met.

Pair Names	Accesses Teachable Machine and sets up an Image Project	Creates and names at least three classes	Collects multiple images per class (with some variation)	Trains the model and tests each class	Identifies at least one situation where the model makes mistakes	Can explain one real life example of AI / ML	Notes
Pair 1:	☐	☐	☐	☐	☐	☐	
Pair 2:	☐	☐	☐	☐	☐	☐	
Pair 3:	☐	☐	☐	☐	☐	☐	
Pair 4:	☐	☐	☐	☐	☐	☐	

General observations:

Annex 2 – Student Exit Ticket

Name (optional):
Date:

1. In my own words, AI is...
2. One difference between traditional programming and machine learning is...

3. Rate yourself from 1 to 5:
"I can explain the difference between AI, machine learning, deep learning and neural networks."
1 □ — 2 □ — 3 □ — 4 □ — 5 □
"I can describe at least one type of machine learning (supervised, unsupervised or reinforcement) with an example."
1 □ — 2 □ — 3 □ — 4 □ — 5 □

Our Teachable Machine model
What three classes did your model learn to recognise?

What did you change to improve your model (for example, more images, different angles, new class)?
Limitation / risk
One limitation or risk I noticed about AI in our activity today is:

Thank you!

Lesson Plan 3:
AI Applications across Industries

1. General Information

Course and Module:	Course 1: A Digital Primer: Artificial Intelligence Essentials Module 3: AI Applications across Industries
AI Tool / Technology Used:	None
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Computer Science / AI literacy
Estimated Duration:	60 minutes (optional extension: 90-120 min)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Identify and explain how AI is applied across 5 key industries: education, healthcare, finance, social media and entertainment, and automotive.
- Analyse specific AI applications and understand their real-world impact on efficiency, decision-making, and user experience.
- Evaluate the benefits and potential limitations of AI applications in different sectors.
- Apply critical thinking to assess how AI could be implemented or improved in familiar contexts, particularly in education.
- Synthesise their understanding through case study analysis and collaborative discussion.

3. Competences Developed

- Digital competence: Understanding how Artificial Intelligence is applied across different industries and recognising its impact on everyday life and professional environments.
- Critical thinking and analytical skills: Analysing the benefits, limitations and ethical implications of AI applications in sectors such as healthcare, education, finance, entertainment and transport.
- Communication and collaboration: Working effectively in groups to discuss case studies, exchange ideas and present conclusions based on evidence.
- Problem-solving: Exploring how AI technologies address real-world challenges and proposing innovative solutions for familiar contexts.
- Research and information literacy: Locating, interpreting and evaluating information about AI applications from different sources while distinguishing reliable information from misconceptions.

4. Prerequisites and Learning Context:

4.1 Prior knowledge required:

Students should have:

- Basic understanding of what artificial intelligence is (from Modules 1 and 2).
- Familiarity with everyday digital technologies and platforms (smartphones, social media, online learning tools).
- Basic reading comprehension and group discussion skills.
- No advanced technical knowledge of AI systems is required, as the lesson focuses on understanding applications rather than technical implementation.

4.2 Learning Context

- This lesson is part of Module 3 within the FUTURE-STEM-HUB AI Essentials course. It builds upon foundational AI concepts introduced in Modules 1 and 2 by demonstrating how these concepts are applied in real-world scenarios across diverse industries.
- The lesson connects abstract AI concepts to students' everyday experiences—from personalised learning platforms they may use at school to healthcare technologies that help doctors diagnose illnesses. By exploring these practical applications, students develop a more concrete understanding of AI's impact on society and gain insights into potential career paths in AI-related fields.
- This module prepares students for subsequent lessons that will explore AI ethics, societal implications, and future developments in the field.

5. Classroom Environment

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:

- Read through Module 3 educational materials thoroughly to understand the content and examples provided.
- Prepare the digital presentation slides or visual aids showing key AI applications in each industry.
- Print or share digitally the case study (The Future of Learning – Alex's Journey with AdaptLearn) and reflective questions.
- Ensure access to devices if students will read materials digitally or conduct additional research.
- (Optional) Prepare the quiz for formative assessment at the end of the lesson or beginning of the next session.

5.2 Classroom Setup

Students work in groups of 4-6 (depending on the class size) to encourage collaborative discussion and diverse perspectives.

The classroom should be arranged to facilitate group discussions, with desks or tables grouped together.

A projector or interactive whiteboard is needed for the teacher's presentation and whole-class discussions.

The teacher should be able to move freely between groups to monitor discussions and provide support.

5.3 Technical requirements (for in-class activities)

Hardware: Projector or interactive whiteboard for teacher presentation; (optional) tablets or laptops if students will access digital materials or conduct research.

Software: PDF reader or web browser to access Module 3 materials; presentation software for teacher's slides.

Internet connection: Optional, but beneficial if students will explore AI applications online or watch supplementary videos.

Note: This lesson does not require coding or technical software. The focus is on understanding concepts through reading, discussion, and analysis.

6. Materials Needed

6.1 Digital Resources:

- Module 3 educational materials (Course 1 - AI Applications across Industries) – provided as reading material for students.
- Case study: 'The Future of Learning – Alex's Journey with AdaptLearn' (included in Module 3 materials).
- Teacher presentation slides showing key AI applications in each industry (to be prepared by teacher based on Module 3 content).
- (Optional) Supplementary videos or infographics about AI in different industries.
- (Optional) Access to platforms mentioned in the module (Khan Academy, Duolingo, etc.) for demonstration purposes.

6.2 Physical Materials:

- Case study handout with reflective questions (printed or digital)
- (Optional) Quiz for formative assessment (printed or digital) – included as Annex 1.
- Flipchart paper and markers for group presentations (if preferred over digital presentation).

7. Step-by-Step Lesson Procedure

This lesson uses a combination of teacher-led introduction, collaborative group exploration, case study analysis, and whole-class discussion to help students understand how AI is applied across different industries and its real-world impact.

7.1. Warm-Up / Introduction (10 minutes)

Teacher actions

Set the context (2 min):
The teacher introduces the lesson by reminding students that AI is not just a theoretical concept — it is actively transforming many industries and aspects of daily life. The teacher explains that today's lesson will explore how AI is used in education, healthcare, finance, social media and entertainment, and the automotive industry.

Warm-up question (5 min):
The teacher posesthe question: "Think about your daily life. Where do you encounter AI without even realising it?"
Students are encouraged to share examples (e.g., social media feeds, music recommendations, smart home devices, navigation apps).
The teacher notes examples on the board and groups them by industry if applicable.

Mini input: Overview of AI across industries (2–3 min):

The teacher briefly presents the five industries that will be explored: Education, Healthcare, Finance, Social Media & Entertainment, and Automotive. For each industry, the teacher gives one example.

Class organisation

Students work in groups of three or more depending on the classroom and available devices.

7.2. Main Activity (30–35 minutes) – Industry Exploration

This section is divided into three parts: teacher introduction of key concepts, group exploration of assigned industries, and group presentations with class discussion.

Part 1 – Teacher Introduction: Key AI Applications (5 minutes)

Before students begin their group work, the teacher provides a structured overview of the types of AI applications found across industries:

- Personalisation: AI tailoring experiences to individual users (e.g., adaptive learning, personalised playlists)
- Automation: AI performing repetitive tasks (e.g., grading, administrative work, manufacturing)
- Prediction and Analysis: AI analysing data to make predictions (e.g., diagnosing diseases, detecting fraud, forecasting demand)
- Decision Support: AI helping humans make better decisions (e.g., treatment recommendations, investment advice)
- Safety and Security: AI monitoring systems to identify risks (e.g., cyberbullying detection, predictive vehicle maintenance)

Part 2 – Teacher instructions: how students will work (2–3 minutes)

The teacher explains that students will explore one industry in depth and identify specific AI applications in that sector.

Part 3 – Group Exploration: Industry Deep Dive (15 minutes)

Aim of this activity: To allow students to explore AI applications in one specific industry in detail through collaborative group work.

Class organisation:

- Divide the class into 5 groups (if the class in big you can spilt students in more groups).
- Assign each group 1 industry: Education, Healthcare, Finance, Social Media & Entertainment, or Automotive.
- Each group should search the Internet and make a list of at least 5 AI applications in their respective industry. For every application the group need to provide a real example. One of these applications should present a potential limitation or ethical concern related to AI use in that industry.
- Each group presents their findings and examples at the end of the activity.

Teacher role:

- Circulate amongst groups to provide guidance and answer questions.
- Ensure groups stay focused on identifying concrete AI applications and their impact.
- Prompt groups to think critically about both benefits and limitations.
- Remind students of time remaining and encourage them to prepare their presentations.

Note for Teachers:

Encourage students to compare AI applications across different industries rather than focusing only on the technology itself. Guide discussions towards both the benefits and limitations of AI, highlighting how the same technology may have different social, ethical and economic impacts depending on the context. Encourage students to justify their opinions using examples and evidence from the lesson.

Follow-up questions (examples):

"How is the AI application you described similar to or different from what we saw in another industry?"
"What might happen if this AI system made a mistake?"

8. Reflection & Assessment

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Reflection and Discussion

Teacher actions:

The teacher leads a whole-class discussion after the case study analysis to consolidate learning and encourage critical reflection. No additional written task is required in this section.
Purpose: Synthesise understanding across all industries and reflect on AI's broader impact.

Guided discussion questions:
- Which industry's use of AI surprised you the most? Why?
- What are some common benefits of AI across all the industries we explored today?
- What might happen if an AI system in healthcare or finance makes a wrong decision? Who should be responsible?

- Can you think of a situation where using AI might not be fair to everyone? (For example, in hiring, lending, or diagnosing illness)
- Can you think of other areas of life where AI could be helpful that we haven't discussed today?
- After learning about AI in education through Alex's story, how do you feel about using AI tools in your own learning?

This discussion is oral and teacher-led, aiming to help students articulate what they have learned and connect concepts across different industries.

8.2 Practical Exercise / Student Output

Purpose: Define clearly what students are expected to produce by the end of the lesson.

Student outputs:

A brief group presentation summarising students' finding.
Written responses to the two reflective questions from the case study (approximately 200-300 words each).

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson. It combines:

- A short quiz
- Teacher observation (Checklist)
- Student self-assessment (Exit ticket)

Short quiz:

See Annex 1
When: At the end of the lesson or at the beginning of the next session.
How: Individually (paper or digital).

Teacher instructions:

- Distribute the quiz to students (included at the end of this lesson plan).
- Allow approximately 10 minutes to complete it.
- Review answers briefly with the class or collect for quick checking. Answers are not graded; they serve as formative feedback.
- The quiz can be done on paper, in a shared document, or using online tools (e.g., Kahoot, Google Forms). Answers are not graded; they are used as quick formative feedback to guide the next steps of teaching.

Teacher Observation Checklist:

See Annex 2
Purpose: Support teacher observation during the group activity and presentations.
How it is used:

- The checklist is completed during group work and presentations, based on group observation.
- Teachers observe group collaboration and understanding, not individual performance in detail.
- Filled in progressively during the lesson.

- Used by the teacher for reflection and future planning; not shared formally with students.

Student Self-Assessment (Exit Ticket)
See Annex 3
At the end of the lesson, students complete a short self-assessment individually (on paper or digitally).
When: At the very end of the lesson, after the reflection discussion.
How: Individually, on paper or digitally.

Teacher instructions:
- Distribute a 1 exit ticket per student.
- Allow approximately 5 minutes to complete it.
- Review answers or collect for quick checking.

8.4 Informal Success Criteria

By the end of the lesson, a student is considered to have met the learning objectives if they can:
- Name and briefly describe at least three AI applications in their assigned industry, supported by a real-world example for each.
- Explain in simple terms how at least one of those applications' benefits users, businesses, or society (e.g., through efficiency, safety, or personalisation).
- Identify at least one potential limitation or ethical concern related to AI use in the industry they explored.
- Connect their industry's AI applications to at least one application discussed by another group, demonstrating cross-industry awareness.

These criteria are not formally assessed or graded. They are used by the teacher to gauge the overall level of understanding during group work and presentations, and to identify students or groups who may need additional support.

9. Adaptations for Diverse Learners

Diverse learners refer to students with different learning speeds, confidence levels, and prior digital experience.

Support strategies:

- Provide simplified summaries or visual infographics of each industry's AI applications for students who struggle with reading comprehension.
- Allow students to work in pairs or small groups during case study analysis if individual work is challenging.
- Offer sentence starters or writing frames for reflective questions (e.g., "AI helps students by...", "One limitation of AI in education is...").
- Provide additional teacher guidance and check-ins for groups that need more support during the industry exploration activity.
- Use visual aids, videos, or demonstrations to supplement written materials.

Time constraints:

If time is limited, reduce the number of industries explored (focus on 3 industries instead of 5).
Complete the case study analysis as homework and discuss responses in the next lesson.

10. Resources / Attachments

Core Learning Materials

- Module 3 "AI Applications across Industries" - provided as reading material for students
- Case study: The Future of Learning - Alex's Journey with AdaptLearn (included in Module 3 materials)
- Teacher presentation slides covering key AI applications across the five industries (to be prepared by the teacher based on Module 3 content)
- Supplementary videos or infographics about AI in different industries – Optional
- Access to platforms mentioned in the module (Khan Academy, Duolingo, etc.) for demonstration purposes – Optional

Printable Documents (as Annexes):

- Annex 1: Short Quiz
- Annex 2: Teacher Observation Checklist
- Annex 3: Student Self-Assessment (Exit ticket)

Annex 1

Short quiz
The quiz can be done on paper, in a shared document, or using online tools (e.g., Kahoot, Google Forms). Answers are not graded; they are used as quick formative feedback to guide the next steps of teaching.

1. Which of the following best describes how AI is used in adaptive learning platforms such as Duolingo or Khan Academy?
a) AI tracks individual progress and adjusts content to match each student's level and pace.
b) AI assigns the same exercises to all students based on age group.
c) AI replaces teachers by delivering all lessons automatically.
d) AI only evaluates students at the end of a course.

2. In healthcare, which of the following is an example of AI supporting medical professionals?
a) AI prescribes medication without any human involvement.
b) AI replaces surgeons in all types of operations.
c) AI analyses medical images to detect early signs of disease and support diagnosis.
d) AI only manages hospital administrative tasks.

3. How does AI contribute to fraud detection in the finance industry?
a) It randomly blocks transactions to prevent suspicious activity.
b) It analyses patterns in transaction data to identify unusual behaviour in real time.
c) It requires customers to verify every transaction manually.
d) It stores all financial records in a secure database.

4. Which of the following best explains how AI personalises content on social media and entertainment platforms?
a) It shows the same trending content to all users worldwide.
b) It limits the amount of content available to avoid information overload.
c) It selects content based only on a user's age and location.
d) It analyses viewing or interaction history to recommend content each user is likely to enjoy.

5. Which of the following is a potential ethical concern about AI use across industries?

a) AI systems may produce biased outcomes if trained on unrepresentative data.
b) AI can only be used in high-income countries.
c) AI always makes decisions more slowly than humans.
d) AI reduces the need for computers and digital devices.

Correct Answers:

1 – a), 2 – c), 3 – b), 4 – d), 5 – a)

Annex 2

Teacher Observation Checklist

Use this checklist to observe student groups during the lesson. Tick (✓) when the criterion is met. This is a group-based observation tool; individual student performance is not recorded in detail.

Group Name(s)	Contributes to group discussion	Identifies 3+ AI applications with real examples	Explains real-world impact (benefits)	Identifies a limitation or ethical concern	Presents findings clearly to the class	Notes
Group 1 (Education):	☐	☐	☐	☐	☐	
Group 2 (Healthcare):	☐	☐	☐	☐	☐	
Group 3 (Finance):	☐	☐	☐	☐	☐	
Group 4 (Social Media & Entertainment):	☐	☐	☐	☐	☐	
Group 5 (Automotive):	☐	☐	☐	☐	☐	

General observations:

Annex 3

Student Exit Ticket

Name (optional): ______________________________ Date: _______

1. Name one AI application you found most interesting today and explain in one sentence what it does.
Application: ____
What it does: ____

2. In your own words, describe one benefit and one limitation of AI in any industry discussed today.
Benefit: ____
Limitation: ____

3. Rate yourself from 1 to 5: "I can explain how AI is used in at least two different industries."
1 □ — 2 □ — 3 □ — 4 □ — 5 □

4. Rate yourself from 1 to 5: "I can identify at least one ethical concern about AI in any of the industries we discussed."
1 □ — 2 □ — 3 □ — 4 □ — 5 □

5. Write one question you still have about AI in any of the industries explored today
____

General observations:

Lesson Plan 4: Fundamentals of Computer Vision & Cognitive Computing

1. General Information

Course and Module	Course 1: A Digital Primer: Artificial Intelligence Essentials Module 4: Cognitive Computing and AI Technologies
AI Tool / Technology Used	Pixel Value Extractor, ELIZA Chatbot, Teachable Machine
Target Age Group	15–18 years (Secondary education).
Subject Area	ICT, Technology, STEM
Estimated Duration	90 minutes (2 sessions of 45 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Distinguish between rule-based pattern matching systems (e.g. ELIZA) and adaptive AI systems that learn from data, providing concrete examples of each.
- Name, describe, and correctly sequence the five core stages of a standard image processing pipeline.
- Analyse how digital images are encoded numerically using RGB pixel values, and explain how resolution affects image quality.
- Evaluate how intelligent systems (e.g. robots) perceive and interpret their environment using CNNs, Reinforcement Learning, and Sensor Fusion.
- Reflect critically on the ethical implications of deploying AI in sensitive contexts such as therapy or healthcare.

This lesson introduces students to the foundations of cognitive computing and computer vision, combining conceptual understanding with hands-on exploration of how AI systems process images and language.

3. Competences Developed

- Digital competence: Using AI-related tools to extract, interpret and evaluate numerical and linguistic data.
- Problem-solving competence: Understanding algorithmic processes behind image recognition and robotic perception.
- Critical thinking competence: Analysing limitations, biases and ethical boundaries of AI-based communication systems.
- Collaboration competence: Working collaboratively in pairs or small groups to analyse AI outputs and share findings.
- Ethical reflection (added): Evaluating the societal impact of deploying AI in contexts requiring human empathy.

4. Prerequisites and Learning Context

4.1 Prior Knowledge Required

Students should have a basic understanding of what Artificial Intelligence is and be familiar with simple digital tools.
No prior programming knowledge is required – all activities are tool-based and guided step by step.

4.2 Learning Context

This lesson builds on previous introductory AI concepts and prepares students for deeper exploration of cognitive computing.
It combines conceptual discussion with practical experimentation using visual and linguistic AI tools.
Suitable for classroom environments with internet access and basic digital devices.

5. Classroom Environment

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:

- Test all online tools (Pixel Value Extractor, ELIZA, Teachable Machine) and ensure they are accessible from school devices.
- Prepare selected example images (varied resolution, lighting, and backgrounds) and save them for quick access during the lesson.
- Review The Mechanical Turk and ELIZA examples to be able to answer student questions fluently.
- (Optional) Prepare a printed one-page handout summarising the five image processing pipeline steps for students who benefit from visual support.

5.2 Classroom Setup

Students work individually or in pairs using laptops or tablets.
A projector or interactive whiteboard is strongly recommended for demonstrations.

5.3 Technical Requirements

- Hardware: Computers or tablets with keyboards; projector or interactive whiteboard.

- Software: Modern web browser (Chrome, Edge, or Firefox). No local software installation required.

- Internet: Stable connection required throughout the lesson.

- Contingency: If internet access is unstable, prepare offline screenshots of each tool for use with the whole class.

6. Materials Needed

- PCs or tablets with stable internet access.
- Online Pixel Value Extractor tool (browser-based, no login required).
- ELIZA chatbot (e.g. NJIT or Masswerk instances, freely accessible).
- Sample digital images with varied resolutions, lighting conditions, and backgrounds (pre-selected by teacher).
- Projector or interactive whiteboard for teacher demonstrations.
- (Optional) Printed handout: Image Processing Pipeline reference sheet (5 stages).
- (Optional) Printed worksheet: Digital Vision Report template for student output.

7. Step-by-Step Lesson Procedure

This lesson uses historical examples and practical image-based exploration to introduce key concepts of cognitive computing and AI technologies through guided analysis and discussion.

7.1 Warm-Up / Introduction (15 minutes).

Purpose
To introduce cognitive computing and challenge students' assumptions about machine intelligence through historical and early AI examples.

Teacher Actions
- Present the 18th-century chess automaton The Mechanical Turk; ask: 'Was this truly intelligent, or a clever simulation?' Allow 2 minutes for student responses.
- Briefly explain Cognitive Computing: how modern AI systems attempt to mimic aspects of human perception, reasoning, and learning – and why that matters.
- Introduce ELIZA as one of the first chatbots; demonstrate a short interaction live, then explain how simple pattern-matching rules create the illusion of understanding.

Student Actions

- Discuss in pairs (2 min): Do The Mechanical Turk and ELIZA represent real intelligence? What is the difference between imitation and genuine understanding?
- Share brief opinions with the class; teacher notes key terms on the board (e.g. cognition, simulation, rules, learning).

7.2 Main Activity (55 minutes) – AI in Action

Phase 1: The Numerical Image – RGB Analysis (15–20 minutes)
Purpose: Help students understand that digital images are numerical data and that AI processes visual information as numbers.

Teacher actions:

- Demonstrate how a digital image is composed of pixels, each defined by three values: Red, Green, Blue (0–255).
- Show how to use the Pixel Value Extractor: select different image regions and read the RGB output.
- Explain: 'When you look at a red apple, you see colour. The computer sees [255, 30, 20]. This is how AI starts to "see" the world.'

Student actions:

- Use the Pixel Value Extractor to analyse a teacher-selected image.
- Identify RGB values from at least three different regions of the image.
- Compare colour differences and discuss: How does numerical variation change visual appearance?
- Record findings in the Digital Vision Report (see Section 8.2).

Phase 2: The Image Processing Pipeline (20 minutes)

Purpose: Help students understand the logical sequence through which AI systems transform raw visual data into meaningful output.

Teacher actions:

- Introduce the five-step image processing pipeline with clear visuals or a diagram on the board:
- Step 1 – Acquisition: Raw image data captured via sensors.
- Step 2 – Pre-processing: Resizing, contrast adjustment, noise reduction.
- Step 3 – Segmentation: Separating foreground objects from background.
- Step 4 – Classification: Using CNNs to detect patterns and assign labels.
- Step 5 – Output: Results visualised or used for automated decision-making.
- Briefly explain each step with a simple, relatable example (e.g. a face recognition system on a phone).

Student actions:

- Work in small groups to reconstruct the correct order of the five stages using printed or digital cards.
- Each group explains one stage in their own words to the class.
- Groups discuss: How could an error in Step 2 (pre-processing) affect the final classification output?

Phase 3: AI in Robotics – Sensor Fusion & Reinforcement Learning (15 minutes)
Purpose: Extend students' understanding of cognitive computing by exploring how AI systems combine perception and learning in robotics.

Teacher actions:

- Present how robots use multiple sensors (camera, lidar, GPS) via Sensor Fusion to build spatial models of their environment (e.g. SLAM – Simultaneous Localisation and Mapping).
- Introduce Reinforcement Learning: a learning paradigm based on trial, error, and reward feedback.
- Provide one concrete real-world example (e.g. autonomous vehicles or warehouse robots) and show a short video clip if available.

Student actions:

- Analyse the example: Which types of sensors are being used and why?
- Discuss: How does the robot improve its behaviour over time through Reinforcement Learning?
- Compare with Phase 1: How is robotic perception different from simple image analysis?

7.3 Reflection (5–10 minutes)

Teacher actions:

- Invite students to share the most surprising thing they learned about how AI processes images.
- Write key terms on the board: perception, data, learning, sensors, classification, bias, threshold.
- Pose closing question: 'How might errors in training data affect an AI system's decisions in the real world?'

Student actions:

- Share one key insight from the lesson.
- Explain one concrete difference between rule-based systems and learning-based systems.
- Reflect: How could errors in data affect AI decisions? Give one example.

Note for Teachers:

Encourage students to think beyond how Computer Vision works by discussing where and why it is used in everyday life. Use real-world examples such as facial recognition, autonomous vehicles and medical imaging to stimulate critical reflection about both the benefits and the ethical challenges of AI-powered vision systems. Emphasise that technological accuracy does not always guarantee fairness or responsible use.

8. Assessment & Reflection

Note for teachers:

The assessment methods below are flexible and complementary. Select and adapt the most suitable approaches according to student needs, time constraints, and learning objectives. Not all tools must be applied in every implementation.

8.1 Discussion – ELIZA and AI Ethics

Students reflect on their interaction with ELIZA and evaluate whether the chatbot appeared coherent, relevant, or empathetic.

Guided discussion question:

- 'Can algorithmic systems replace human empathy in sensitive contexts such as therapy or counselling? What are the ethical risks and societal implications?'
- Encourage students to consider: Who is responsible if an AI gives harmful advice?
- Discuss real cases where chatbots have been used in mental health apps and what ethical frameworks should guide their deployment.

8.2 Practical Exercise – Digital Vision Report

Students write a 'Digital Vision Report' (150–200 words) that must include:

- Extracted RGB values from a chosen image segment, with interpretation of what those values represent visually.
- A short summary of the ELIZA chatbot interaction, evaluating whether the bot seemed intelligent.
- An analysis of at least two factors (e.g. lighting, perspective, background complexity) that may cause AI systems to misclassify objects.

A brief personal reflection: In what context should AI perception systems NOT be used autonomously, and why?

8.3 Assessment Methods

Assessment is formative and takes place during and immediately after the lesson. It combines a quick quiz, teacher observation, and student self-assessment.

Quick Quiz (5 Questions – Conceptual Understanding)

- What is the main difference between a pattern-matching system (like ELIZA) and an adaptive AI system that learns from data?
- Which information is stored in a single RGB pixel, and why is this numerical representation important for AI systems?
- Name and briefly describe two stages of the image processing pipeline. Give a simple example for each.
- Why can changes in lighting or background cause AI systems to misclassify objects? Name two real-world consequences of such errors.
- Why is ELIZA not considered a truly intelligent cognitive system, despite appearing conversational?

Teacher Observation Checklist

- Student explains how an image becomes numerical data using RGB values.
- Student correctly interprets RGB values from the Pixel Value Extractor.
- Student distinguishes between human perception and machine perception with examples.
- Student critically evaluates ELIZA's responses and identifies its limitations.
- Student contributes constructively to group discussion with relevant examples.
- Student identifies at least one ethical concern related to AI perception systems.

Student Self-Assessment

- I can explain how AI systems 'see' the world using numbers.
- I understand why AI systems can make mistakes and can give a real example.
- I can distinguish between simulated intelligence (rule-based) and real learning systems.
- I feel confident discussing ethical limits of AI technologies in class.
- I can describe the five stages of the image processing pipeline in my own words.

9. Adaptations for Diverse Learners

- Visual support: Use a mosaic or grid analogy to explain pixels and resolution; provide a printed pipeline diagram students can annotate.
- Guided instruction: Provide step-by-step screen guides with screenshots for all digital tools to support students with limited digital experience.
- Language support: Provide a key vocabulary list (10–12 terms) with simple definitions at the start of the lesson.
- Extended challenge: Independent research task on AI perception systems in space exploration (e.g. Mars rovers or autonomous satellites); students present findings in 3 minutes at next session.
- Reduced complexity: Students who struggle with the full Digital Vision Report may submit key bullet points with at least one RGB observation and one ethical reflection.

10.Resources/Attachments

The following resources support the implementation of this lesson:

- Digital Resources
- Pixel Value Extractor tool.
- ELIZA chatbot.
- Teachable Machine.
- Teacher presentation slides on Cognitive Computing and Computer Vision.
- Short video or visual resource explaining how computers process images.
- Digital images with different resolutions, lighting conditions and backgrounds.

Physical / Printable Materials

- Image Processing Pipeline reference sheet.
- Digital Vision Report template.
- Reflection questions.
- Exit Ticket.

Teacher Resources

Teachers are encouraged to adapt the resources according to the available digital tools, internet access and students' prior knowledge. If online tools are not accessible, screenshots or printed examples may be used to support classroom discussion.

Lesson Plan 5: Don’t Trust the Machine: When AI Is Wrong (and You Believe It)

1. General Information

Course and Module:	Course 1: A Digital Primer: Artificial Intelligence Essentials Module 5: Ethical Considerations and the Future of AI
AI Tool / Technology Used:	Search engines, generative AI tools (e.g. ChatGPT), social media algorithms (conceptual use)
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Artificial Intelligence / Ethics & Citizenship
Estimated Duration:	60 minutes

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Recognise that AI systems and search engines can produce incorrect or misleading information.
- Identify risks related to blind trust in AI-generated content.
- Understand how bias, privacy issues, and automation affect everyday decisions.
- Reflect on their own responsibility when using AI tools.
- Apply critical thinking when evaluating AI outputs.

This lesson focuses on challenging students' blind trust in AI by exposing real-life errors, manipulation, and limitations of AI systems, encouraging responsible and critical use of artificial intelligence.

3. Competences Developed

- Ethical reasoning: Reflecting on the ethical challenges associated with Artificial Intelligence, including bias, misinformation, privacy and responsible use of digital technologies.
- Critical thinking: Evaluating AI-generated information critically rather than accepting it as automatically accurate or trustworthy.
- Digital literacy: Developing responsible habits when using AI tools, search engines and digital platforms to access and verify information.
- Social awareness: Recognising how Artificial Intelligence influences society, personal decision-making and online interactions.
- Decision-making skills: Applying informed judgement when interpreting AI outputs and deciding how to respond to potentially inaccurate or biased information.

4. Prerequisites and Learning Context:

4.1 Prior knowledge required:

Students should be familiar with:

- Using Google or other search engines
- Using social media platforms
- Hearing about or using AI tools such as ChatGPT

No technical knowledge of AI systems or programming is required.

4.2 Learning Context

- This lesson is designed as part of Module 5 and focuses on ethical awareness and future implications of AI. It is strongly connected to students' daily digital habits and intentionally avoids abstract ethical theory.
- The lesson places students in situations where AI decisions directly affect their own lives, helping them understand why ethical responsibility matters.

5. Classroom Environment

5.1 Pre-lesson Preparation

The teacher should prepare:

- A slide or board with statements for voting
- Scenario cards (printed or digital)
- A timer for activities

5.2 Classroom Setup

Students work:

- Individually (voting moments)
- In small groups (3–4 students)
- Classroom should allow discussion and movement.

5.3 Technical requirements (for in-class activities)

Hardware:
Projector or interactive board

Internet:
Optional (lesson can run offline)

6. Materials Needed

6.1 Physical Materials

- Slides or printed statements
- Scenario cards
- Exit ticket sheets (or digital form)

7. Step-by-Step Lesson Procedure

This lesson uses everyday situations familiar to students and guided ethical scenarios to challenge blind trust in AI and encourage responsible, critical decision-making.

7.1. Warm-Up / "AI or Truth?" (10 minutes)

Teacher action

The teacher shows 5 statements and asks students to vote TRUE / FALSE quickly.
Example statements:

- "If a website is the first result on Google, it must be reliable."
- "AI detectors can always detect AI-generated text."
- "If an answer from AI sounds confident, it is probably correct."
- "Using AI for homework is not cheating if the final answer is correct."
- "Social media algorithms show neutral content."

After voting, the teacher reveals:
All 5 statements are false or misleading (intentionally). I'll explain why, with a very short explanation.

Answers:

1."If a website is the first result on Google, it must be reliable."
❌ FALSE
Why:
Google ranks content based on relevance, SEO, popularity, and paid positioning, not on truth or accuracy.
The first result is not necessarily the most correct one — only the best positioned.
Key classroom phrase:
"Google ranks pages, not truth."

2. "AI detectors can always detect AI-generated text."

❌ FALSE

Why:

AI detectors:

often fail,

produce false positives,

do not work well with edited or mixed (human + AI) texts.

Key classroom phrase:

"AI detectors guess — they don't know."

3."If an AI answer sounds confident, it is probably correct."

❌ FALSE

Why:

AI does not know whether something is true.

It only predicts words that sound plausible together.

An answer can sound very confident and still be completely wrong.

Key classroom phrase:

"Confidence is not accuracy."

4 - "Using AI for homework is not cheating if the final answer is correct."

❌ FALSE / MISLEADING

Why:

Most educational rules consider AI use misuse if it is not declared or cited.

Learning is not just about submitting the correct answer.

Key classroom phrase:

"Correct answer ≠ honest work."

5 - "Social media algorithms show neutral content."
❌ FALSE
Why:
Algorithms promote content that:
gets more clicks,
triggers emotions,
keeps users engaged for longer.
They are not neutral — they optimise engagement.
Key classroom phrase:
"If it's free, you are the product."
"If you trusted at least one of these… you already trusted AI too much."

7.2. Main Activity (30 minutes) – When AI Decides for You

Part 1 – "AI Made the Decision" (15 minutes)

Purpose
The aim of this activity is to help students understand that AI systems can make impactful decisions based on probabilities, and that high accuracy does not guarantee fairness, safety, or ethical acceptability for every individual.
Students analyse real-life scenarios where AI systems are used to make decisions that directly affect people, encouraging critical reflection on responsibility, trust, and human oversight.

Group Organisation

- Students work in small groups of 3–4.
- Groups should be as mixed as possible to encourage different perspectives and discussion. Each group represents people affected by an AI-based decision.
- Pair work is not recommended for this activity, as small-group discussion supports deeper reflection and disagreement.

Teacher Input: Explaining "The AI is 85% accurate"

Before distributing the scenarios, the teacher explains the following concept in simple language:
"When we say that an AI system is 85% accurate, it means that out of 100 decisions, about 85 are correct and 15 are wrong."
The teacher then asks students:
"If you are one of the 15 people affected by a wrong decision, would that feel acceptable to you?"
The teacher briefly clarifies that:
Accuracy does not mean fairness.
Accuracy does not mean the system is correct for everyone.
A high percentage can still hide systematic bias or harmful outcomes.
Key idea introduced:
High accuracy does not mean low harm.

Part 2 – "Would You Trust AI With This?" (10 minutes)

Purpose
The aim of this activity is to help students recognise that trust in AI is context-dependent and that not all decisions should be delegated to automated systems.
By analysing everyday situations, students reflect on where AI support is acceptable and where human judgement remains essential, without framing the discussion as a moral lecture.

Activity Description
This activity is conducted individually first, followed by a short guided discussion.
Students are presented with a series of situations in which Artificial Intelligence systems are used to make or influence decisions that could affect their daily lives.
For each situation, students must decide how much they trust AI to make that decision.

Instructions to Students

The teacher explains:
"For each situation, decide how much you would trust an AI system to make this decision."
Students choose one option only for each case:
I trust AI
I'm not sure
I don't trust AI
Students should make their choices individually and silently at first.

Teacher tip:

To help students better visualise this concept, the teacher may draw 100 small circles on the board and mark 15 of them in a different colour to represent incorrect decisions. This simple visual representation helps students understand that even a high accuracy rate can still affect many individuals.

Scenario Distribution

Each group receives one Scenario Card.
Each card contains:

- a description of the AI-based decision,
- a statement about the system's accuracy (85%),
- a hidden limitation or condition that affects the decision-making process.

Students first read the scenario and the accuracy claim.
After 2 minutes, the teacher reveals the hidden condition printed on the card.

Group Task: Structured Analysis

Groups must discuss and agree on answers to the following questions, in the given order:

- Personal Impact

If this AI decision affected you personally, would you accept it?
(Yes / No / Not sure – one answer per group)

- Risk Identification

What could go wrong in this situation?
(e.g. unfair exclusion, misunderstanding, discrimination, lack of context)

- Responsibility

If the AI makes a wrong decision, who should be responsible?
(the company, the institution using the AI, programmers, humans supervising the system)

- Human Control

Should a human be able to change or override this decision? Why or why not?

The teacher circulates among groups, asking guiding questions but not providing answers, encouraging students to justify their opinions.

Sharing and Reflection

Each group shares one sentence only, completing the following frame:
"We would / would not accept this AI decision because..."
The teacher writes key words from student responses on the board (e.g. fair, risky, biased, fast, efficient, unfair).
The teacher closes the activity with the reflection:
"AI decisions may feel objective, but AI systems do not understand consequences. Humans do."

SCENARIO CARDS
(Ready to print or copy into slides)

Scenario Card 1 – Scholarship Decision
The Decision
An AI system decides which students receive a scholarship based on academic records and online activity.
The Accuracy Claim
The AI system is 85% accurate.
Hidden Condition (revealed later)
The AI was trained mainly using data from students over 18 and from previous years.

Scenario Card 2 – Social Media Account Flag

The Decision

An AI system automatically flags and temporarily blocks social media accounts for “suspicious behaviour”.

The Accuracy Claim

The AI system is 85% accurate.

Hidden Condition (revealed later)

The AI does not understand sarcasm, jokes, or context in posts and comments.

Scenario Card 3 – Online Exam Monitoring

The Decision

An AI system monitors students during online exams and detects possible cheating.

The Accuracy Claim

The AI system is 85% accurate.

Hidden Condition (revealed later)

The AI struggles with poor internet connections and unusual camera angles.

Scenario Card 4 – Student Risk Prediction

The Decision

An AI system predicts which students are “at risk of failing” and recommends extra monitoring.

The Accuracy Claim

The AI system is 85% accurate.

Hidden Condition (revealed later)

The AI was trained mostly on data from students with similar social and economic backgrounds.

Scenario Card 5 – Content Recommendation
The Decision
An AI system decides which news and videos students see most often online.
The Accuracy Claim
The AI system is 85% accurate.
Hidden Condition (revealed later)
The AI prioritises content that generates strong emotional reactions and longer screen time.

Part 2 – "Would You Trust AI With This?" (10 minutes)

Purpose
The aim of this activity is to help students recognise that trust in AI is context-dependent and that not all decisions should be delegated to automated systems.
By analysing everyday situations, students reflect on where AI support is acceptable and where human judgement remains essential, without framing the discussion as a moral lecture.

Activity Description
This activity is conducted individually first, followed by a short guided discussion.
Students are presented with a series of situations in which Artificial Intelligence systems are used to make or influence decisions that could affect their daily lives.
For each situation, students must decide how much they trust AI to make that decision.

Instructions to Students

The teacher explains:
"For each situation, decide how much you would trust an AI system to make this decision."
Students choose one option only for each case:

- I trust AI
- I'm not sure
- I don't trust AI

Students should make their choices individually and silently at first.

Situations Presented

The teacher displays or reads the following situations one by one:

- An AI system chooses which news you see online.
- An AI system decides whether your social media account is suspicious.
- An AI system corrects and grades your exams.
- An AI system recommends what career or study path you should follow.
- An AI system decides who should be stopped by the police for further checks.

Individual Decision Phase (4 minutes)

- Students record their choices on paper or mentally.
- No discussion is allowed at this stage.
- The goal is to capture instinctive trust reactions.

IGroup Reflection and Discussion (6 minutes)

Students then discuss their answers in their small groups.
The teacher guides the discussion using the following questions:

- Why did you trust AI in some cases but not in others?
- Which decisions felt “too important” to give to a machine?
- Did accuracy matter more than consequences?
- Would you feel differently if the decision affected someone else instead of you?

The teacher encourages students to justify their choices and to listen to different perspectives.

Teacher’s Role

During the discussion, the teacher should:

- Avoid confirming or rejecting student opinions.
- Encourage explanation rather than agreement.
- Highlight differences between convenience, efficiency, and responsibility.

Key idea to reinforce verbally:
“Trust in AI is not all or nothing. It depends on the situation.”

Transition to Part 3

The teacher concludes Part 2 by saying:
“If we trust AI in some situations but not in others, then the real question is not ‘Is AI good or bad?’ but ‘Who decides, and who is responsible?’”
This sentence naturally leads into Part 3 – Reality Check.

Part 3: "Reality Check: Who Is Responsible?" (10 minutes)

Purpose

The aim of this activity is to help students understand that Artificial Intelligence systems do not make moral or responsible decisions on their own, and that responsibility always lies with humans.

This part connects students' previous reflections to broader ethical concerns such as accountability, bias, privacy, and the future role of AI in society.

Activity Description

This is a teacher-guided whole-class reflection designed to consolidate learning from Parts 1 and 2.

The activity encourages students to move from personal reactions to societal responsibility, without introducing abstract ethical theory.

Teacher Prompt

The teacher writes the following sentence on the board or displays it on the screen:

"AI does not care about you. It only follows patterns."

The teacher allows 5–10 seconds of silence before continuing.

Guided Reflection Questions

The teacher asks the following questions, allowing short student responses after each one:

- Who designs AI systems?
- Who decides what data is used to train them?
- Who chooses where AI is used?
- Who is affected when AI makes a wrong decision?

Students are encouraged to answer freely.

There are no "correct" answers at this stage.

Linking to Ethical Concerns

The teacher briefly connects student responses to key ethical issues:

- Bias – when AI systems reflect unfair patterns in data
- Privacy – when personal data is collected or used without consent
- Job displacement – when automation replaces human work
- Human rights and responsibility – when decisions affect people's lives

This explanation should remain short and clear, avoiding long theoretical input.

Key Message

The teacher states clearly:

"AI can support decisions, but it should not replace human responsibility."

Optional follow-up question:

"In which situations should humans always have the final decision?"

Closing Reflection

To close the activity, the teacher says:

"Technology is powerful, but it is not neutral in its effects.

The way we choose to use AI today will shape the society we live in tomorrow."

Students are then asked to complete the Exit Ticket or final reflection task.

Note for Teachers:

This lesson encourages students to question AI-generated information rather than accepting it automatically. Facilitate open discussion about misinformation, bias and digital responsibility, reminding students that AI is a powerful tool but not an infallible source of truth. Encourage learners to verify information using reliable sources and to justify their opinions with evidence rather than assumptions.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:
The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Discussion

Teacher actions:
The teacher leads a whole-class discussion immediately after completing Parts 1–3 of the lesson activities.
No additional technical or practical task is required in this section.

Purpose: Consolidate learning by helping students articulate how their perception of AI has changed, focusing on trust, responsibility, and real-life consequences rather than technical details.

Guided discussion questions:

The teacher guides the discussion using some or all of the following questions:

- In which situations today did you trust AI automatically, without questioning it?
- What made you stop trusting AI in some scenarios?
- Why can a system be "85% accurate" and still be unfair to some people?
- Who should be responsible when an AI system makes a wrong decision?
- In which situations should humans always have the final decision?

This discussion is oral and teacher-led, aiming to help students connect the activities with their everyday digital behaviour and future responsibilities as citizens.

8.2 Practical Exercise / Student Output

Purpose: Define clearly what students are expected to produce by the end of the lesson, focusing on reflection and personal awareness rather than technical performance.

Student outputs

Active participation in group and whole-class discussions.

A short individual exit ticket completed at the end of the lesson.

Exit Ticket (2 minutes)

When:

At the very end of the lesson, after the reflection discussion.

How:

Individually, on paper or digitally.

What students do:

Students answer three short prompts:

- Write one situation where trusting AI could be risky.
- Write one decision you believe AI should not make alone.
- Write one question you now have about AI and the future.

Teacher use
The exit tickets are collected as quick formative feedback.
They are not graded and are used to:
- check understanding,
- identify misconceptions,
- inform future lessons on AI ethics and responsibility.

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.
It combines:
- a short quiz,
- teacher observation,
- student self-assessment.

Quick Quiz (5 questions)

When:
At the end of the lesson or at the beginning of the next session.
How:
Individually (paper or digital).
Answers:
Provided below in the teacher notes / answer key.

Teacher instructions:
- Distribute the quiz to students.
- Allow approximately 5 minutes to complete it.
- Review answers briefly with the class or collect for quick checking.

Quiz questions

- What does it mean when we say an AI system is "85% accurate"?
- Why can an AI system be accurate and still be unfair?
- Name one ethical concern related to AI mentioned in today's lesson.
- Who should be responsible if an AI system makes a harmful decision?
- Write one simple action you can take to avoid blindly trusting AI or online information.

Possible Answers (Teacher Notes)

What does it mean when we say an AI system is "85% accurate"?
It means that out of 100 decisions, about 85 are correct and about 15 are wrong.

Why can an AI system be accurate and still be unfair?
Because errors may affect some people more than others, and bias in data can lead to systematically worse outcomes for certain groups.

Name one ethical concern related to AI mentioned in today's lesson.
Examples: bias/discrimination, privacy/surveillance, job displacement, misinformation/deepfakes, academic honesty.

Who should be responsible if an AI system makes a harmful decision?
Humans and institutions: the organisation using the AI and the people who design, deploy, and supervise it (AI itself cannot be responsible).

Write one simple action you can take to avoid blindly trusting AI or online information.
Examples: check another source, look for evidence, verify with reliable websites, ask "who benefits?", question confident answers, do not copy-paste without understanding/citing.

Teacher Observation Checklist

Purpose:
Support teacher observation during Parts 1–3 of the lesson.
How it is used:
The checklist is completed during group activities and whole-class discussion.
When:
Filled in progressively during the lesson.
After the lesson:
Used by the teacher for reflection and planning.
Results do not need to be formally shared with students, but general feedback may be discussed.
Checklist
[ ] Participates actively in group discussion
[ ] Identifies at least one risk of blind trust in AI
[ ] Explains why accuracy does not guarantee fairness
[ ] Recognises the need for human responsibility
[ ] Listens to and respects different opinions

Student Self-Assessment

Purpose
Encourage students to reflect on their own learning and awareness after the lesson.

When
Completed individually at the end of the lesson (together with the exit ticket).

How
On paper or digitally.

What students do
Students complete the following:
• Rate 1–5: I can explain why trusting AI blindly can be risky.
• Rate 1–5: I can identify situations where humans should have the final decision.
• One sentence: One thing I will think about differently when using AI.

After completion

Students keep the self-assessment for personal reflection.
The teacher may briefly address common doubts or misconceptions in the following lesson.

9. Adaptations

Additional Support

- Provide simplified definitions of key terms such as misinformation, bias, hallucination, privacy and AI-generated content.
- Use concrete, age-appropriate examples of AI mistakes before moving to more abstract ethical discussions.
- Provide guided worksheets with sentence starters to help students structure their answers.
- Allow students to work in pairs when analysing AI outputs or fact-checking information.

For Students with Lower Digital Confidence

- Demonstrate step by step how to compare an AI-generated answer with reliable online sources.
- Provide a short checklist for evaluating information: source, date, author, evidence and possible bias.
- Use teacher-selected examples instead of asking students to search independently.
- Pair students with more digitally confident classmates during verification activities.

Fast Finishers

- Find an additional example of AI misinformation or biased output and explain why it could be problematic.
- Compare two AI-generated answers to the same question and evaluate which one is more reliable.
- Create a short "responsible AI use" checklist for classmates.
- Write a short paragraph explaining how AI tools should be used responsibly in schoolwork.

10.Resources/Attachments

Digital Resources

- Generative AI tool (e.g., ChatGPT or equivalent).
- Internet search engine for fact-checking activities.
- Teacher presentation on AI bias, misinformation and hallucinations.
- Short videos or news examples illustrating AI errors.
- Interactive whiteboard or projector.

Physical / Printable Materials

- Fact-checking worksheet.
- AI Output Analysis Sheet.
- Case Study Cards.
- Reflection Questions.
- Exit Ticket.

Course 2:
Delving Deeper in AI with Python and Scratch

Empowering Secondary School STEM Education with AI
Training and Resources for Students and Educators
2024-1-DE03-KA220-SCH-000247346

Lesson Plan 6:
Aylin and Alara: AI with Python & Neural Networks

1. General Information

Course and Module	Course 2: Delving Deeper in AI with Python and Scratch Module 1: Aylin and Alara – Your AI Adventure Begins
AI Tool / Technology Used	Python (Google Colab)
Target Age Group	15–18 years (Secondary education)
Subject Area	ICT / Computer Science / AI Literacy
Estimated Duration	90 minutes (or 2 sessions of 45 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Explain what Artificial Intelligence (AI) is and how it relates to neural networks, decision trees, and reinforcement learning, using everyday examples (e.g. recommendation systems).
- Write and run basic Python code in Google Colab to simulate AI decision-making using weighted inputs.
- Create a simple AI decision model using Python, based on weighted inputs that simulate a neural network output.
- Critically evaluate how different weight assignments affect AI recommendations, and identify at least one potential bias or ethical concern in their own model.

This lesson uses the fictional story of Aylin and her AI smartwatch Alara to make abstract AI concepts tangible and personally relevant for secondary school learners.

3. Competences Developed

- Digital competence: Using Google Colab and Python for hands-on coding.
- Problem-solving: Debugging code, completing coding challenges, adjusting weights to reach a target decision.
- Critical thinking: Understanding how AI 'thinks' through rules and weighted data; questioning whose values are encoded.
- Collaboration: Pair work in coding tasks using Driver/Navigator roles.
- Creativity: Modifying code to explore different outcomes and design personalised decision systems.
- Ethical reflection (added): Recognising how weight choices reflect values, and who could be disadvantaged by biased weights.

4. Prerequisites and Learning Context

4.1 Prior Knowledge Required

- Basic computer skills: opening links, typing, running pre-written code cells in a browser-based environment.
- No prior knowledge of AI or programming is required – all concepts are introduced within the session.
- Familiarity with basic logic (e.g. 'if–then' rules) is helpful but not required.

4.2 Learning Context

- This lesson is the introductory session for Course 2 Module 1 but works as a standalone AI literacy lesson.
- Foundational concepts introduced here (rule-based systems, weighted decisions, learning from data) serve as building blocks for subsequent lessons.
- AI concepts are linked to everyday digital experiences (recommendation systems, social media algorithms) to ensure immediate relevance.

5.Classroom Environment

5.1 Pre-lesson Preparation

- Open and test the Google Colab notebook thoroughly in advance; verify that all four stations load correctly
- Prepare projector/screen with the Aylin and Alara image from Module 1.
- Verify that all student devices can access the internet and open Google Colab.
- Familiarise yourself with all four coding stations and their expected outputs.
- (Optional) Print Station Summary Cards and Exit Tickets if physical materials are preferred.

5.2 Classroom Setup

- Students work in pairs sharing one device – this fosters discussion and collaboration.
- Use Driver/Navigator roles; rotate every 5–7 minutes so both students gain coding experience.
- Organise the classroom so the teacher can circulate easily between groups.

5.3 Technical Requirements

- **Hardware**: Computers or tablets with keyboards; projector or interactive whiteboard.
- **Software**: Modern web browser (Chrome, Edge, Firefox) to access Google Colab. No installation required.
- **Internet**: Stable connection required throughout the lesson to run Python code online.

Note: Google Colab is a browser-based notebook environment that allows students to run Python code without any software installation. Students open the notebook, follow the stations, and execute or edit code directly in the browser.

6. Materials Needed

6.1 Digital Resources

- One device (PC/laptop/tablet) per student pair, with a web browser and internet access.
- Google Colab notebook (requires a Google account; pre-shared link via QR code or classroom platform).
- Printed or digital Logic Flowchart template (for warm-up).
- Printed or digital Python Cheat Sheet (basic syntax: variables, lists, if-else).
- Story excerpts from the Aylin and Alara narrative (Module 1).

6.2 Physical / Printable Materials

- Station Summary Cards (optional, one set per pair – see Section 10).
- Exit Ticket (optional, one per student – see Section 10).
- Poster materials (paper, markers) for the Personal AI Decision Poster output activity.

7. Step-by-Step Lesson Procedure

This lesson uses the Aylin and Alara story context and three coding stations in a Google Colab notebook to introduce key AI ideas through guided, hands-on exploration.

7.1 Warm-Up – "The Human Algorithm" (10 minutes)

A. Setup (2 min)

- Display the Aylin and Alara image from Module 1 on the projector.
- Choose 2–3 volunteers for quick responses during brainstorming.

Have Google Colab open and ready to demonstrate.

B. Opening Question & Brainstorming (3 min)

"How many of you have used YouTube, TikTok, or Spotify today? Have you ever wondered how these apps know what you like?"

- Facilitate quick brainstorming: students share examples (video suggestions, personalised playlists, friend recommendations).
- Summarise: 'These are all examples of AI recommendations – but how does the AI actually make these decisions?'

C. Character Introduction & Bridge to Coding (3 min)

Point to the Aylin and Alara image: 'This is Aylin and her AI smartwatch Alara. Just like you get recommendations, Aylin gets advice from Alara every day – but how does Alara actually think?'
'Today, you'll find out by programming a small AI yourself!'
Open Google Colab and run a brief demo: name = input("What's your name? ") / print("Hello " + name + "! I'm Alara.")
'See? Not magic – just a program that takes input, processes it, and gives output. This is how AI systems start.'

D. State the Learning Goal (1 min)

'Today you'll write code that makes decisions – just like a small AI brain. By the end, you'll understand how an AI learns to make choices and why the values you give it matter.'

7.2 Main Activity (30 minutes) – AI in Action

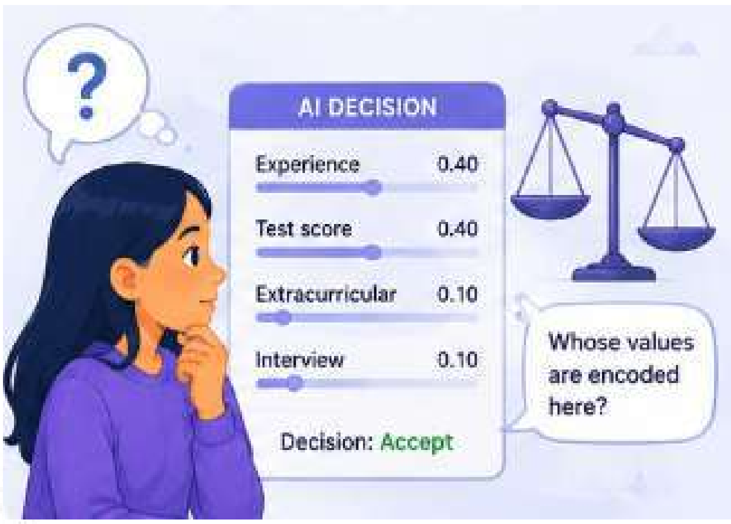
Part 1: Teacher Input – Key Concepts (5 minutes)

Write on board/projector:
INPUT → [WEIGHTS → SCORE → THRESHOLD] → DECISION
Simple explanations:

- Input: 'The information you give the computer (e.g. camera quality, price).'
- Weight: 'How important each factor is (0.5 = very important, 0.1 = less important).'
- Score: 'The calculated result after applying weights.'
- Threshold: 'The line that separates decisions (above 7 = buy, below 5 = don’t buy).'
- Decision: 'The final recommendation based on the score.'

Part 2: Explain Workflow (3 minutes)

- Assign pairs: one Driver (types), one Navigator (reads instructions and checks maths).
- Complete stations 1–3 in order; only change marked sections (# TODO or values to fill in).
- If an error occurs: re-run the cell, compare with a neighbouring pair, then ask the teacher.
- Timing: Station 1 = 10 min (teacher-led), Station 2 = 10 min (pair work), Station 3 = 7 min (creative application), + 5 min buffer.



Station 1: Teacher Demonstration – School Route Decision (10 minutes)

Teacher demonstrates live on projector:

```
# Decision: School route
rain = input("Is it raining? (yes/no): ")
bus = input("Is the bus coming? (yes/no): ")
time = input("Are you in a hurry? (yes/no): ")

score = 0
if rain == 'no': score += 0.5 # Rain matters most
if bus == 'yes': score += 0.3
if time == 'no': score += 0.2

if score > 0.5:
 print('Walk!')
else:
 print('Take the bus!')
```

- Ask: 'What factors matter when you're at the bus stop in the morning? Rain? Bus coming? Time pressure?'
- Assign weights live with the class: 'We give rain 0.5 because getting wet is the worst outcome for most people.'
- Test two scenarios with the class and discuss the results together.
- Key message: 'The biggest weight goes to the most important factor – just like when you say: the main thing is not getting wet!'

Station 2: Student Practice – Phone Buying Advisor (10 minutes)

```
# Alara's Phone Buying Advisor
camera = float(input('Camera quality (0-10): '))
price = float(input('Price (0-10, 10=cheap): '))
battery = float(input('Battery life (0-10): '))

w_camera, w_price, w_battery = 0.5, 0.3, 0.2
score = camera*w_camera + price*w_price + battery*w_battery

if score > 7: print('BUY IT!')
elif score > 5: print('Maybe...')
else: print('Probably not')
```

- Students enter their own phone ratings; observe how the weights drive the recommendation.
- Teacher circulates and asks: 'Why does camera get 0.5 weight? Are weights adding up to 1.0?'
- 'Try rating a phone with Camera=9, Price=3, Battery=7. Then Camera=5, Price=9, Battery=5. Compare recommendations.'

Station 3: Creative Application – My Personal Decision Maker (7 minutes)

- Students choose their own decision situation and 3 factors with weights that reflect their own values.
- Weights must sum to 1.0; students write a brief 'because...' justification for each weight.
- Teacher prompt: 'If something is super important to you, give it 0.6 or more. The weights show what YOU value most.'
- Fast finishers: add a 4th factor, test edge cases (all ratings = 0 or 10), or reverse weights to see how decisions change.

7.3 Closure & Reflection (5 minutes)

Write on board:

MY VALUES → WEIGHTS → AI DECISION

- 'What surprised you about how AI makes decisions?'
- 'How is giving weights to factors similar to how you make decisions in real life?'
- 'What happens if we give wrong weights to an AI? Example: camera weight = 0.9, price weight = 0.1 for someone on a tight budget.'
- 'Why is it important that weights add up to 1.0?'

Closing message: 'Remember: AI isn't magic. It's just maths + your values. The weights you choose reflect what YOU think is important. That's why different people need different AI recommendations – and why we must think carefully about who designs AI systems.'

Note for Teachers:

Students may find the concept of neural networks abstract at first. Emphasise that the goal is not to understand advanced mathematics but to recognise how simple neurons, inputs and weighted connections contribute to AI decision-making. Encourage students to experiment with different values, discuss unexpected results and explain their reasoning to their classmates.

8. Assessment & Reflection

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Discussion

- Technical understanding: 'In your own words, what does a weight do in an AI neural network?'
- Real-world connection: 'Where in your daily life have you seen AI make decisions like the ones we built today?'
- Bias awareness: 'Your phone buying AI gave camera quality 50% weight. What if someone values price more? How does changing weights change who the AI favours?'
- Limitations: 'Our AI only looked at 3 factors. What important factors might it be missing? How could that lead to unfair decisions?'
- Future thinking: 'Now that you know how simple AI decisions work – what is ONE area where you think AI should NOT make decisions for humans?'

8.2 Student Output – Personal AI Decision Poster

Each pair creates a Personal AI Decision Poster (digital or on paper) that includes:

- A real-life decision scenario (e.g. 'Should I study or go out tonight?').
- 3 factors with assigned weights (must sum to 1.0), each with a short 'because...' explanation.
- A sample input/output from their AI code.
- A brief reflection: Does the AI's recommendation match what they would actually decide? If not, why not?

2–3 pairs briefly present their poster to the class, explaining their weight choices and any surprises.

8.3 Quick Quiz (5 Questions)

- What does a 'weight' do in an AI neural network? a) Makes the program run faster b) Shows how important a factor is c) Counts how many times something happens d) Changes the colour of the output [Answer: b]
- In our phone buying AI, camera quality had a weight of 0.5. What does this mean? a) The camera is 50% of the total decision b) The camera score is divided by 0.5 c) The camera gets 5 extra points d) The camera doesn't matter much [Answer: a]
- Why should weights in an AI decision maker add up to approximately 1.0? a) Because 1.0 is a lucky number b) Because computers only understand 1 c) Because it represents 100% of all decision factors d) Because Python requires it [Answer: c]
- Which of these is an example of AI making a weighted decision in real life? a) A calculator adding numbers b) Spotify recommending songs based on your listening history c) A printer printing a document d) A clock showing the time [Answer: b]
- If you change a weight from 0.3 to 0.6, what happens? a) The program stops working b) That factor becomes twice as important in the decision c) The score gets smaller d) Nothing changes [Answer: b]

8.4 Teacher Observation Checklist

- Student can explain why a specific weight changes the final recommendation.
- Student can identify the if-elif-else structure in the code.
- Student participates in the discussion about whether Alara's 'thinking' reflects fixed rules or human-assigned values.
- Student identifies at least one ethical concern related to biased weights.
- Student successfully runs and modifies code in Google Colab.

8.5 Student Self-Assessment (scale 1–5)

- I can open and run code in Google Colab. ○ ○ ○ ○ ○
- I understand how to change weights in the AI code. ○ ○ ○ ○ ○
- I can explain what a 'weight' does in AI. ○ ○ ○ ○ ○
- I see how AI makes decisions in my daily life. ○ ○ ○ ○ ○
- I can think of at least one ethical concern about AI decision-making. ○ ○ ○ ○ ○
- The most interesting thing I learned today was: ________________

9. Adaptations.

Differentiation Strategie

- For fast learners: Add a 4th factor, create nested decisions, design a scoring system for a real personal problem, or explore how biased weights affect minority groups.
- For learners who need more support: Provide pre-filled weight values, use simpler examples (just 2 factors), or work in teacher-led small groups with scaffolded code.
- For visual learners: Create bar charts showing weight distribution; use coloured highlighting in code (# RED = most important, # BLUE = medium, # GREEN = least).

- For attention challenges: Timer with 5-minute checkpoints; movement breaks between stations; shorter tasks focused on Stations 1 and 2 only.
- Extensions: Design recommendation system for music or movies; explore bias – what if weights encode stereotypes instead of personal values?

Technical Contingency Plans

- If Colab doesn't work: Use offline Python with Thonny; provide printed code templates for students to trace through manually.
- If no internet: Use prepared code files on USB drive with an offline Python installation.
- If not enough devices: Groups of 3 with clear role rotation (Driver, Navigator, Calculator/Observer).
- If students finish early: 'Design Challenge' – improve the phone buying advisor with better factors or more nuanced decision thresholds.

10. Resources / Attachments

Digital Resources

- Google Colab notebook (Lesson 6).
- Python neural network starter code.
- Teacher presentation on simple neural networks.
- Interactive whiteboard or projector.
- Optional short video explaining artificial neurons

Physical / Printable Materials

- Neural Network Diagram worksheet.
- Python Cheat Sheet.
- Coding Challenge Sheet.
- Reflection Questions.
- Exit Ticket.

Lesson Plan 7:
Aylin and Alara: AI with Data Structures and Logic in Python

1. General Information

Course and Module	Course 2: Delving Deeper in AI with Python and Scratch Module 1: Aylin and Alara – Your AI Adventure Begins
AI Tool / Technology Used	Python (Google Colab)
Target Age Group	15–18 years (Secondary education)
Subject Area	ICT / Computer Science / AI Literacy
Estimated Duration	90 minutes (or 2 sessions of 45 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Explain the concept of “Data Input” and how variables serve as named containers for information in an AI-driven programme, using a concrete example from the lesson.
- Write and execute Python code using lists to store, access, and organise multiple data points correctly.
- Create a simple rule-based decision engine using if-else and elif logic to simulate how an algorithm makes choices based on specific criteria.
- Evaluate the “Garbage In, Garbage Out” (GIGO) principle by observing how incorrect or biased data entries lead to faulty programmatic decisions, and reflect on who is responsible.
- Identify at least one ethical implication of rule-based AI systems: who is responsible when Alara makes a wrong decision?

This lesson focuses on how data structures and logical rules influence AI behaviour, highlighting the impact of data quality and rule design on real-world outcomes.

3. Competences Developed

- Digital competence: Writing Python code and using Google Colab to execute algorithms.
- Problem-solving: Troubleshooting syntax errors and logically structuring if-else statements to achieve a desired outcome.
- Critical thinking: Evaluating how data input affects algorithmic decisions; recognising the limitations of rule-based systems.
- Computational thinking (added): Breaking down complex decisions into smaller, logical steps that a computer can process – a transferable skill beyond programming.

- Ethical reflection (added): Understanding that AI "intelligence" is a direct reflection of human-coded logic and data, and considering the responsibility of the creator.

4. Prerequisites and Learning Context

4.1 Prior Knowledge Required

- Basic computer skills: opening links, typing, running pre-written code cells in a browser-based environment (Google Colab).
- No prior knowledge of AI or programming is required; all concepts are introduced within the session.
- Basic logic (if–then rules) is helpful but briefly introduced during the warm-up.

4.2 Learning Context

- This lesson is the introductory session for Course 2 Module 1 and works as a standalone AI literacy lesson.
- Foundational concepts introduced here (variables, lists, rule-based logic, GIGO) act as building blocks for subsequent lessons in the module.
- AI concepts are linked to everyday digital experiences to ensure personal relevance and engagement.

5. Classroom Environment

5.1 Pre-lesson Preparation

- Test the Google Colab notebook in advance and verify that all four stations load correctly.
- Prepare sticky notes (or small cards) with data points written clearly (numbers, fruit names, or colours).

- Plan the hidden sorting rule for the warm-up: start simple (numerical order or alphabetical).
- Have a projector/screen ready and (optional) print Station Summary Cards and Exit Tickets.

5.2 Classroom Setup

- Students work in pairs sharing one device, using Driver/Navigator roles rotated every 5–7 minutes.
- Arrange the classroom so the teacher can circulate easily and reach all pairs.

5.3 Technical Requirements

- **Hardware**: Computers or tablets with keyboards; projector or interactive whiteboard.
- **Software**: Modern web browser (Chrome, Edge, Firefox) to access Google Colab. No installation required.
- **Internet:** Stable connection required throughout the lesson.

Note: Google Colab is a browser-based notebook environment. Students open the notebook link (shared via QR code or classroom platform), follow the stations, and execute or edit code directly in the browser.

6. Materials Needed

6.1 Digital Resources

- One device (PC/laptop/tablet) per student pair, with a web browser and internet access.
- Google Colab notebook with 4 pre-built stations (requires a Google account; link shared in advance).
- Printed or digital Logic Flowchart template (for warm-up).
- Printed or digital Python Cheat Sheet (basic syntax: variables, lists, if-else, elif).
- Course 2 Master Document (digital or printed excerpts for the Aylin & Alara story context).

6.2 Physical / Printable Materials

- Sticky notes (5–8 per group) with data points written clearly (numbers, words, or colours).
- Marker pens and whiteboard space for drawing parallels during the warm-up.
- Station Summary Cards (optional, one set per pair – see Section 10).
- Exit Ticket (optional, one per student – see Section 10).

7. Step-by-Step Lesson Procedure

This lesson uses the Aylin and Alara story context and four coding stations in a Google Colab notebook to introduce key AI ideas through guided, hands-on exploration.

7.1 Warm-Up – "The Human Algorithm" (10 minutes)

A. Setup (2 min)
Gather attention at front; choose one volunteer to act as the "AI.
Distribute sticky notes with data points to 5–8 other students randomly.

B. Activity (3 min)

- Tell the class: 'Our volunteer is going to act like an AI. I'll give them a secret rule. Note-holders will come up one by one. The AI will decide where to place them in line.'
- Whisper the rule to the "AI" student (e.g. 'Sort smallest to largest' or 'Alphabetical A–Z').
- Note-holders come up in random order; the "AI" arranges them, thinking aloud.
- Pause after 3–4 notes: ask the class to guess the rule.

C. Reveal & Connect to AI (4 min)

- Reveal the rule once all notes are sorted. Draw parallels on the whiteboard:
- Sticky notes = Data points in a computer.
- The AI student's brain = A programme following instructions.
- The sorting rule = Algorithm (a set of steps).
- Introduce the concept of a List: 'In Python, we put data in something called a List – just like our sticky notes in a row.'

```
numbers = [10, 3, 25, 7, 15]
sorted(numbers) # Python sorts this automatically!
```

Bridge to AI: 'AI isn't magic – it's instructions + data. Today we'll store data in variables and lists, then write rules (if-else) for the computer to make decisions.'
Briefly open Google Colab, create a new notebook, and demo creating and sorting a list.

7.2 Main Activity (30 minutes) – AI in Action

Part 1: Teacher Input – Key Terms (5 minutes)

Write on board/projector:
INPUT → [DECISION TREE] → OUTPUT (with WEIGHTS & THRESHOLD)

- Input/Output: 'What the computer receives / what it gives back.'
- Decision: 'A yes/no choice – like a fork in the road.'
- Weight: 'How important is this information? (e.g. grades matter more than homework).'
- Threshold: 'The point where the decision changes (e.g. above 50% = pass).'
- Decision Tree: 'Multiple if–then questions in sequence, each branching to the next.'

Part 2: Explain Workflow (3 minutes)

- Assign pairs: Driver (types) and Navigator (reads instructions and plans). Rotate roles every 5–7 minutes.
- Complete stations 1–4 in order; only modify sections marked with # TODO.
- If an error occurs: re-run the cell, compare with a neighbouring pair, then ask the teacher.
- Timing: Station 1 = 5 min, Station 2 = 7 min, Station 3 = 8 min, Station 4 = 7 min, + 5 min buffer.

1. Station 1: Variables – Change the Output (5 minutes)

Students run the code, change a variable value, and observe what changes in the output.
Teacher circulates and asks: 'What did you change? What changed in the output?'
Watch for common errors: missing quotation marks around strings.
Guiding question: 'Why do we need variables in programming instead of just writing values directly?'

Station 2: Toy Neuron – Weighted Decision (7–10 minutes)

```
# Station 2: Simple AI Decision – Should I watch a movie?
homework_done = 5 # 1–10: how finished
movie_exciting = 8 # 1–10: how exciting

weight_homework = 0.7
weight_movie = 0.3

score = (homework_done * weight_homework) + (movie_exciting * weight_movie)
threshold = 6.0

if score > threshold:
 print('Decision: YES, watch the movie!')
else:
 print('Decision: NO, do homework first.')
```

- Explain: 'The computer calculates a number and compares it to the threshold.'
- Ask: 'What happens if you set threshold to 10? Why? What if homework_done = 0?'
- Real-world connection: 'This is how real AI decides things – like whether an email is spam.'
- Key misconception to address: 'In this simple model, weights always add up to 1.0 so that the total score stays on a consistent scale.'

4. Station 3: Alara's Battery Check – Simple if-else (8 minutes)

Teacher demonstrates first (5 min), then students work independently.

```
# Alara's Battery Check
battery = 15

if battery < 20:
 print('Alara: Low battery! Please charge me.')
else:
 print('Alara: I am ready to help you, Aylin!')
```

- Emphasise: the colon (:) at the end of if and else lines, and mandatory indentation.
- Difference between = (assignment) and == (comparison) – a very common beginner mistake.
- Encourage experimentation: 'Change battery to 50 – what happens?'
- Critical check: ensure all students successfully run this station before moving on.

Station 4: Alara's Battery Status – elif Chain (7 minutes)

```
battery = 50

if battery > 80:
 print('Alara: Full power! Let's go, Aylin!')
elif battery > 20: # NOTE: order matters!
 print('Alara: I am active and ready.')
else:
 print('Alara: Battery low! Save energy now.')
```

- Explain: 'The computer reads top to bottom. Once it finds a True condition, it executes that block and skips the rest.'
- Anticipated problem: 'Why does it never say Full power? – Because 90 is also > 20! Order must be: first > 80, then > 20.'
- Probing question: 'What would happen if we switched the order of conditions? Test it!'
- Extension task for fast finishers: Add a > 90 condition ('Super charged!') and a 50–80 condition ('All systems go!').

7.3 Closure & Reflection (5 minutes)

Write on board:
INPUT → PROCESSING → OUTPUT (Variables, if/else, Weights)

- 'What surprised you most today?'
- 'Where do you see similar decision trees in real life?' (apps, games, navigation, medical diagnosis)
- 'What is the difference between human and computer decision-making?'
- Closing message: 'Remember: AI isn't magic. It's instructions (like if/else) working on organised data (like variables and lists). Today you programmed a tiny AI brain – and you are responsible for its decisions.'

Note for teachers:

Encourage students to focus on the reasoning behind the code rather than memorising Python syntax. The aim of the lesson is to understand how AI uses variables, lists and logical conditions to process information and make decisions. Allow students time to experiment, make mistakes and discuss different approaches, reinforcing that debugging is a natural and valuable part of programming.

8. Assessment & Reflection

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Discussion

- The Responsibility of the Creator: 'If Alara makes a wrong decision because our rules were not precise, who is responsible – Alara or the person who wrote the if-else chain?'
- From Fixed Rules to Learning: 'Alara follows exact rules we gave her. How is this different from a system that learns from patterns (e.g. face recognition)?'
- Real-World Data Bias (GIGO): 'What happens if we forget to define the else safety net? What if the battery data is corrupted and sends a letter instead of a number?'
- Ethical implication: 'If a medical AI uses biased data (e.g. trained only on data from one demographic), what decisions might it get wrong? Who is harmed?'

8.2 Student Output – Smartwatch Logic Engine

- Students produce a 'Smartwatch Logic Engine' – a Python script (.py file or shared Google Colab link) that:
- Runs without syntax errors (correct use of colons and indentation).
- Uses elif statements in the correct order (e.g. checking > 80 before > 20).
- Produces output messages that reflect the Alara & Aylin story from Module 1.
- Includes at least one comment explaining a design decision (e.g. why > 80 is checked first).

8.3 Quick Quiz (5 Questions)

- What is the main purpose of a VARIABLE in programming? a) To make the programme colourful b) To store information with a name so we can use it later c) To connect to the internet d) To delete old data [Answer: b]
- In our lesson, we created a rule for Alara to check her battery. What is this decision-making process called? a) Random guessing b) If-else logic c) Internet searching d) Mathematical calculation [Answer: b]
- When we create rules for AI, why is the ORDER of these rules important? a) The computer only checks rules until it finds one that matches b) Computers read rules from right to left c) Python prefers alphabetical order d) It makes the programme faster [Answer: a]
- What does 'Garbage In, Garbage Out' mean for AI systems? a) Old computers should be recycled b) Bad quality data leads to bad decisions c) AI needs regular maintenance d) Programmes should be simple [Answer: b]
- To store a collection of items like your favourite songs, which data structure is best? a) A single variable b) A list c) An if-statement d) A print command [Answer: b]

8.4 Teacher Observation Checklist

- Student correctly indents print commands in if-else blocks.
- Student can explain what happens to the logic if the battery_level variable changes.
- Student can describe the difference between = and == with an example.
- Student participates in the discussion about AI responsibility and the GIGO principle.
- Student identifies at least one real-world consequence of biased data in an AI system.

8.5 Student Self-Assessment

- I can explain how a computer uses an if statement to make a decision. (Yes / Partially / Not yet)
- I feel confident using elif to add more than two options to a programme. (Yes / Partially / Not yet)
- I understand that Alara's "intelligence" depends on the rules and data I provide. (Yes / Partially / Not yet)
- I can describe what "Garbage In, Garbage Out" means in my own words. (Yes / Partially / Not yet)

9. Adaptations

For Diverse Learners

- Simplified instructions: Provide a "Logic Map" with arrows showing how the if-else gate works before students start typing.
- Scaffolded code (for learners new to coding): Provide a Starter Code file where battery_level and the basic if: / else: structure are already written; students only fill in Alara's messages.

- Syntax Cheat Sheet: One-page reference showing the colon (:) position and required indentation (4 spaces) to prevent the most common errors.
- Extra time: Students who need more time focus only on Stations 1 and 3; Station 4 (elif chain) is optional.
- Advanced extension – Mood Sensor: Fast learners add a second variable is_charging = True/False. Challenge: Modify the code so Alara says "I am charging, please wait" if is_charging is True, regardless of battery level. This introduces Boolean operators and nested logic.

Technical Contingency Plans

- If Colab doesn't work: Use Thonny or Python IDLE offline; provide printed code templates.
- If no internet: Prepared code files on USB drive with an offline Python installation.
- If not enough devices: Groups of 3 with role rotation (Driver, Navigator, Observer/Calculator).
- If students finish early: "Code Detective" – provide a code snippet with 3 intentional bugs to find and fix.

10. Resources / Attachments

Digital Resources

- Google Colab notebook (Lesson 7).
- Python Cheat Sheet.
- Logic Flowchart template.
- Teacher presentation on variables, lists and logical conditions.
- Interactive whiteboard or projector.

Physical / Printable Materials

- Logic Flowchart worksheet.
- Python syntax reference sheet.
- Coding Challenge Cards.
- Reflection Questions.
- Exit Ticket.

Lesson Plan 8:
Aylin and Alara: How AI 'Thinks' (Rules, Simple Neurons & Decision Trees)

1. General Information

Course and Module:	Course 2: Delving Deeper in AI with Python and Scratch Module 1 - Aylin and Alara - Your AI Adventure Begins
AI Tool / Technology Used:	Python (Google Colab) + simple rule-based logic + toy neuron (weighted sum + threshold)
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Computer Science / AI literacy
Estimated Duration:	60 minutes (optional extension: 90-120 min)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Distinguish rule-based systems from learning-based AI (basic ML idea).
- Explain the 'neuron' metaphor: inputs + weights + threshold -> decision.
- Complete and run 4 short Python stations (1-4) and interpret outputs.
- Identify limitations and potential bias in simple classification/recommendation rules.

This lesson focuses on building foundational AI literacy by helping students understand how simple AI systems "think" through rules, weights and thresholds.

3. Competences Developed

- Digital competence: Using Artificial Intelligence tools and Python programming to understand how AI systems process data, make predictions and solve real-world problems through practical experimentation.
- Problem-solving: Designing, testing and improving AI-based solutions by analysing results, identifying errors and refining algorithms to achieve more accurate outcomes.
- Critical thinking: Evaluating how AI models make decisions, recognising their limitations, questioning the reliability of outputs and reflecting on the impact of data quality on model performance.
- Collaboration: Working effectively in pairs or small groups to develop coding activities, share ideas, test AI models and discuss different approaches to solving problems.
- Creativity: Exploring different coding solutions, modifying AI models and experimenting with alternative strategies to design innovative and effective Artificial Intelligence applications.

4. Prerequisites and Learning Context

4.1 Prior knowledge required:

- Students should have basic computer skills such as opening links, typing simple text, and running pre-written code cells in a browser-based environment (Google Colab).
- No prior knowledge of artificial intelligence or programming is required.
- The lesson may optionally connect to previous ICT topics such as basic logic, variables, or "if–then" rules, but these concepts are briefly introduced during the lesson itself.

4.2 Learning Context

- This lesson is designed as an introductory session within Module 1 and can also be implemented as a standalone AI literacy lesson.
- It introduces foundational concepts such as rule-based systems, simple decision-making, and the idea of learning from data, which will be further developed in subsequent lessons of the module.
- The lesson connects AI concepts to students' everyday digital experiences, such as recommendation systems used in music, video, and social media platforms, helping students understand how AI influences daily decision-making.

5. Classroom Environment

5.1 Pre-lesson Preparation

- Before the lesson, the teacher should:
- Ensure access to the Google Colab notebook and test it in advance.
- Prepare a projector or screen for a short teacher demonstration.
- Check that all devices can access the internet and open the notebook correctly.

- Familiarise themselves with the four stations and the expected outputs.
- (Optional) Print station summary cards and exit tickets if physical materials are used.

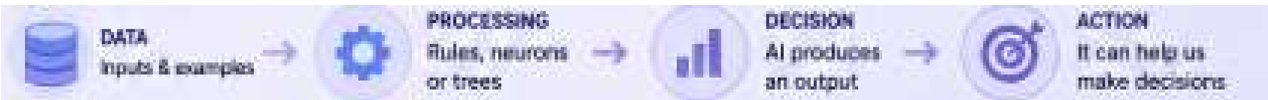
5.2 Classroom Setup

- Students work in pairs using one shared device to encourage collaboration and discussion.
- Pair-programming roles (driver and navigator) are recommended and should be rotated every 5–7 minutes.
- The classroom should be organised so that the teacher can easily move between groups to monitor progress and provide support.

5.3 Technical requirements (for in-class activities)

- **Hardware**: Computers or tablets with keyboards; projector or interactive whiteboard for demonstrations.
- **Software**: Modern web browser (e.g. Chrome, Edge, Firefox); access to Google Colab (no software installation required).
- **Internet** connection: A stable internet connection is required throughout the lesson.

Note: Google Colab is a browser-based notebook environment that allows students to run Python code online without installing any software. Students will open the Google Colab notebook, follow the stations, and execute or edit the code directly in the browser.



6. Materials Needed

6.1 Digital Resources:

- Projector/screen for a short teacher demonstration.
- Video: 3Blue1Brown - 'But what is a neural network?' (subtitles available in all languages).

6.2 Physical Materials:

- Module 1 notebook in Google Colab (Course 2 educational materials).
- (Optional) Printed station summary cards and a 2-minute exit ticket.
- (included as part of this lesson plan – printable or digital)

7. Step-by-Step Lesson Procedure

This lesson uses a short story context ("Aylin and Alara") and four coding stations in a Google Colab notebook to introduce key AI ideas through guided exploration.

7.1. Warm-Up / Introduction (5–10 minutes)

Teacher actions

- **Set the context (1–2 min):**

The teacher briefly introduces the story characters and explains that the lesson will explore how computers make decisions.

- **Warm-up question (2 min):**

"How does your phone guess what you want to watch or listen to?"
The teacher encourages a few short student responses.

- **Mini input (2–3 min):**

The teacher explains the difference between:
Rule-based systems (if X, then Y), and
Learning-based AI (systems that improve using data).

- **Technical orientation (2–3 min):**

The teacher explains that students will use Google Colab, which allows Python code to run online without installing any software.
A short demonstration shows how to:
run a code cell,
edit a value,
reset and re-run the notebook if needed.

Class organisation

Students work in pairs (recommended) or groups of three if devices are limited.
Pair-programming roles are used:
Driver: types and runs the code
Navigator: reads instructions and checks logic
Roles are swapped every 5–7 minutes to ensure active participation.

7.2. Main Activity (20–30 minutes) – AI in Action

Part 1 – Introduction/ Teacher input: key terms (3–5 minutes)

Before students begin the practical work, the teacher introduces key terms in simple language:

- Input / Output: what the computer receives and what it returns
- Decision: a yes/no choice
- Weight: the importance given to an input
- Threshold: the point at which a decision changes
- Decision tree: choosing different outcomes using conditions

Part 2 – Teacher instructions: how students will work (2–3 minutes)
The teacher explains the workflow:
Each pair works through Stations 1–4 in order.
Students read the instructions in each station, run the code, and modify only the indicated values or lines.
If an error occurs, students are encouraged to:
re-run the cell,
compare their work with another pair,
ask the teacher for support.

Part 3: Guided Practice:

Step 1 – Setup (3–5 minutes)

Purpose: Ensure all students can access and use the Google Colab notebook.
Teacher actions:
Ask students to open the Google Colab notebook in pairs.
Confirm that all pairs can:
run a code cell,
identify which parts of the code are editable.
Solve basic technical issues before moving on.

Class organisation:
Students work in pairs (one shared device).
Pair-programming roles are assigned (driver / navigator).

Step 2 - Station 1 (5 min): Inputs/outputs and variables

Purpose: Help students understand how changing a variable affects the program output.
Activity: Students work with a pre-prepared code cell and complete one missing value.

What students do (in the notebook)

- Run the provided code cell.
- Replace one missing variable value.
- Observe how the output changes.

Code provided in the notebook (This code is already prepared by the teacher / project team.)

Python

```
# Station 1: Variables change the output

name = "_____" # TODO: write your name (or a nickname)
age = 14 # TODO (optional): change the age

print("Hello", name)
print("Next year you will be", age + 1)
```

What students must do

Replace "_____" with a name (e.g. "Aylin").
Click **Run**.
(Optional) Change the value of age and run the cell again.

Expected output (example)

If name = "Aylin" and age = 14:
Hello Aylin
Next year you will be 15

Teacher role

Explain clearly:
"You are not creating code from zero. The notebook is prepared; you only fill in small parts."
Walk around the classroom and check that all pairs can run the cell.

Ask the guiding question:
"What did you change, and what changed in the output?"

If students get an error (simple guidance)
Check that quotation marks are correct ("Aylin").
Re-run the cell.

Fast finisher (optional 1–2 minute extension)
Python
```
# Extension: add a new variable
favourite_subject = "Science"
print("Favourite subject:", favourite_subject)
```

Step 3 - Station 2 (10–12 min): Toy neuron (weighted sum + threshold)

Purpose
Help students understand how a simple AI decision can be made by combining inputs, weights, and a threshold.

Teacher pacing note:

- For younger students, the toy neuron concept should be introduced gradually. Teachers may first explain the meaning of inputs using everyday examples (e.g. homework done, slept well), then introduce weights as the importance of each input, and finally explain the threshold as the point where the decision changes.

- Running the example once together with the whole class before students modify the values can help ensure that all students understand the logic of the model.

Activity
Students explore a pre-prepared "toy neuron" model by changing numeric values and observing how the decision changes.

What students do (in the notebook)
Run the provided code cell.
Change one or more numeric values (inputs or threshold).
Observe how the final decision ("Yes" or "No") changes.

Code provided in the notebook
(This code is already prepared by the teacher / project team. Students do not write it from scratch.)

Python

```python
# Station 2: Toy neuron (simple decision)

# Inputs (example values)
homework_done = 1 # 1 = yes, 0 = no
slept_well = 1 # 1 = yes, 0 = no

# Weights (importance of each input)
w_homework = 0.6
w_sleep = 0.4

# Threshold
threshold = 0.7

# Weighted sum
score = homework_done * w_homework + slept_well * w_sleep
```

```
# Decision
if score >= threshold:
 decision = "YES"
else:
 decision = "NO"

print("Score:", score)
print("Decision:", decision)
```

What students must do

Run the cell once with the default values.
Change **only one value at a time,** for example:
set slept_well = 0, or
increase the threshold to 0.9.
Run the cell again and observe what changes.

Expected behaviour (example)

With threshold = 0.7 → decision may be **YES**.
With threshold = 0.9 → decision may change to **NO**, even with the same inputs.
(Exact numbers are less important than observing the change.)

Teacher role

Explain in simple terms:

- inputs = information,
- weights = importance,
- threshold = how strict the decision is.

Encourage students to test **different thresholds.**
Reinforce the idea that this is **not real AI**, but a simplified model to understand how decisions work.

Check-in question

"If the threshold is higher, does the model say 'YES' more often or less often? Why?"

If students get an error (simple guidance)

Check that values are numbers (0 or 1, not text).

Make sure the cell was re-run after changing values.

Fast finisher (optional 2-minute extension)

Python

```
# Extension: change the importance (weights)
w_homework = 0.3
w_sleep = 0.7
```

Step 4 - Station 3 (8–10 min): Decision tree (if/else choices)

Purpose

Help students understand how computers make decisions by following conditional rules (if / else statements) organised as a simple decision tree.

Activity

Students complete a pre-prepared decision tree by filling in one missing condition and observing how different inputs lead to different outcomes.

What students do (in the notebook)

Run a ready-made code cell.

Complete one missing condition in the decision tree.

Test different input values and observe how the output changes.

Code provided in the notebook

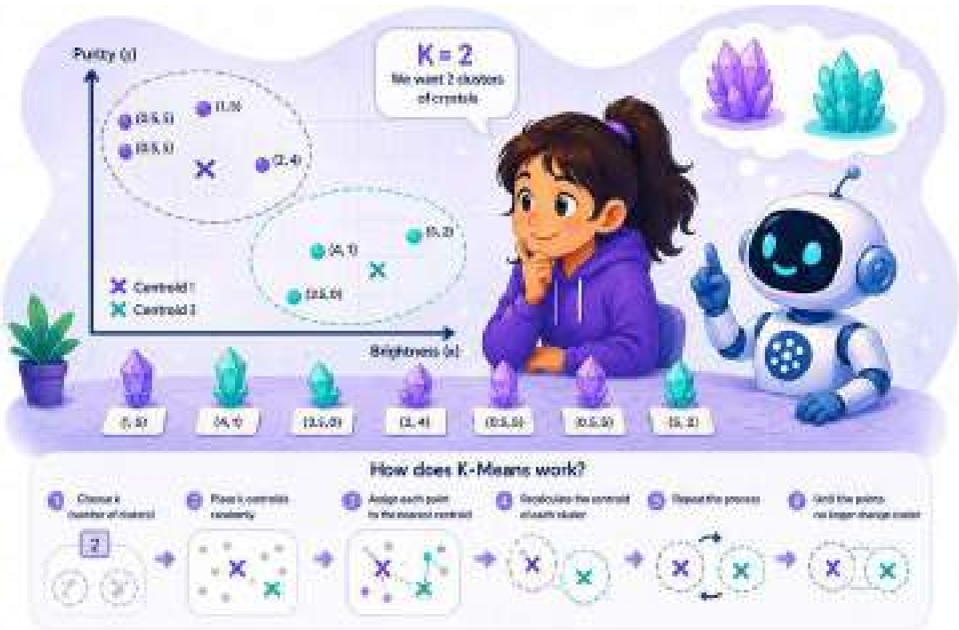
(This code is already prepared by the teacher / project team. Students only complete the indicated part.)

Python

```
# Station 3: Decision tree – choosing how to go to school

weather = "rainy" # options: "sunny", "rainy"
distance = "far" # options: "near", "far"


# Decision tree
if weather == "sunny":
 transport = "walk"
else:
 # TODO: complete the condition
 if distance == "_____":
 transport = "bus"
 else:
 transport = "bike"
```



print("Recommended transport:", transport)

What students must do

Replace "____" with the correct value ("far").

Run the cell.

Change the values of weather and/or distance and run the cell again.

Expected behaviour (example)

If weather = "sunny" → walk

If weather = "rainy" and distance = "far" → bus

If weather = "rainy" and distance = "near" → bike

Teacher role

Remind students that the computer follows rules step by step.

Support pairs who struggle with conditions or indentation.

Encourage students to test all possible combinations of inputs.

Check-in question

"Which condition is checked first, and why does that matter?"

If students get an error (simple guidance)

Check spelling of text values ("sunny", "rainy", "near", "far").

Check indentation (spaces at the beginning of lines).

Re-run the cell.

Fast finisher (optional 2-minute extension)

Python

```
# Extension: add a new condition
has_bike = True # True or False

if weather == "sunny" and has_bike:
 transport = "bike"
```

Step 5 - Station 4 (8–10 min): Rule-based classification and limitations

Purpose
Help students understand how rule-based systems classify information and why such rules can fail or introduce bias.

Activity
Students use a simple keyword-based classifier to categorise music genres and analyse incorrect or biased classifications.

What students do (in the notebook)
Run a pre-prepared code cell.
Modify keywords or inputs.
Observe correct and incorrect classifications.

Code provided in the notebook
(This code is already prepared by the teacher / project team. Students only change the indicated values.)

Python
```
# Station 4: Rule-based music classification

song_description = "fast beats and electronic sounds" # TODO: change this text
```

```
if "guitar" in song_description or "drums" in song_description:
 genre = "Rock"
elif "electronic" in song_description or "beats" in song_description:
 genre = "Electronic"
elif "piano" in song_description or "violin" in song_description:
 genre = "Classical"
else:
 genre = "Unknown"

print("Song description:", song_description)
print("Predicted genre:", genre)
```

What students must do

- Run the cell with the default description.
- Change song_description (e.g. add or remove keywords).
- Run the cell again and observe how the classification changes.

Expected behaviour (example)

"fast beats and electronic sounds" → Electronic
"guitar and electronic sounds" → may be classified incorrectly or unexpectedly
"soft piano music" → Classical

Teacher role

- Emphasise that the computer is only following fixed rules, not “understanding” music.
- Guide students to find examples that are misclassified.
- Introduce the idea of bias and limitations in rule-based systems.

Check-in question

“Which example was classified incorrectly, and why did the rule fail?”

If students get an error (simple guidance)

- Check spelling and use of quotation marks.
- Ensure keywords match exactly (e.g. "electronic" vs "electronics").
- Re-run the cell after changes.

Note for teachers:

Students may initially view AI as something complex or mysterious. Reinforce that today's activities simplify how AI makes decisions through rules, weighted values and decision trees. Encourage students to predict outcomes before running the code, compare different solutions and explain their reasoning. Emphasise that there is rarely a single correct solution and that analysing mistakes is an important part of learning how AI systems work.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Discussion

- Teacher actions:
- The teacher leads a whole-class discussion immediately after completing Stations 1–4.
- No additional coding or practical task is required in this section.

Purpose: Consolidate learning and support critical reflection after the practical activities.

Guided discussion questions:

- What part of today was 'AI' and what part was just rules?
- What data would you need to make recommendations better than fixed keywords?
- Where can bias or errors appear in simple rules?
- How is a weighted decision (Station 2) more flexible than a hard rule?

This discussion is oral and teacher-led, aiming to help students articulate what they experienced during the coding stations.

8.2 Practical Exercise / Student Output

Purpose: Define clearly what students are expected to produce by the end of the lesson.

Student outputs:

- A completed Google Colab notebook with Stations 1–4 executed (basic or advanced level, depending on time and student ability).
- A short exit ticket completed individually.

Exit Ticket (2 minutes)

When: At the very end of the lesson, after the reflection discussion.
How: Individually, on paper or digitally.
What students do: Answer three short prompts.
Exit ticket prompts:

- Write one example of a rule-based system.
- Write one example of a learning-based system.
- Describe one limitation you noticed in today's activities.

Teacher use:
Collected as quick formative feedback.
Not graded; used to check understanding and inform future lessons.

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.
It combines:

- a short quiz,
- teacher observation,
- student self-assessment.

Quick Quiz (5 questions)

When: At the end of the lesson or at the beginning of the next session.
How: Individually (paper or digital).
Answers: Provided in the teacher notes / answer key.
Teacher instructions:

- Distribute the quiz to students.
- Allow approximately 5 minutes to complete it.
- Review answers briefly with the class or collect for quick checking.

- What is Alara in the story?
- Which station introduces a simple 'neuron' decision with a threshold?
- Which structure do we use in Station 3 to choose between options?
- What is rule-based classification?
- If you increase the threshold in Station 2, will the model say 'yes' more often or less often?

Possible Answers:
1. What is Alara in the story?
A simple AI character that represents how computers make decisions using rules and basic logic.

2. Which station introduces a simple “neuron” decision with a threshold?
Station 2 – Toy neuron (weighted sum + threshold).

3. Which structure do we use in Station 3 to choose between options?
An if / else conditional structure (decision tree).

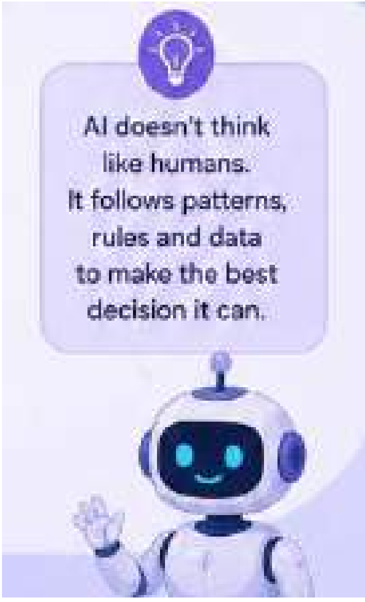
4. What is rule-based classification?
A system that classifies information using fixed rules or keywords instead of learning from data.

5. If you increase the threshold in Station 2, will the model say “yes” more often or less often?
Less often, because the decision becomes more strict.

Teacher Observation Checklist

Purpose: Support teacher observation during the practical activity.
How it is used:
The checklist is completed during Stations 1–4, based on group observation.
Teachers observe pair work, not individual performance in detail.

When:
Filled in progressively during the lesson.
After the lesson:
The checklist is used by the teacher for reflection and planning.
Results do not need to be shared formally with students, but key points may be discussed informally.



Checklist
[ ] Opens and runs the notebook without help.
[ ] Completes Station 1 correctly.
[ ] Explains the effect of the threshold in Station 2.
[ ] Uses if/else logic correctly in Station 3.
[ ] Works collaboratively (role rotation).
[ ] Identifies one limitation or potential bias.

Student Self-Assessment

Purpose: Encourage student reflection on their own learning.
When:
Completed individually at the end of the lesson (together with the exit ticket).

How:
On paper or digitally.
What students do:
Rate their understanding (1–5).
Write one short reflective sentence.

After completion:
Students keep the self-assessment for personal reflection.
The teacher may briefly discuss common difficulties in the next lesson.

- Rate 1-5: I can explain the difference between rules and learning.
- Rate 1-5: I can explain what the threshold does in Station 2.
- One sentence: What is one limitation of today's approach?

9.Adaptations

For Diverse Learners

Diverse learners refers to students with different learning speeds, confidence levels, and prior digital experience.

Support strategies:
Focus on Stations 1–2 only.
Provide a pre-opened notebook link.
Offer closer teacher guidance and peer support.

Time constraints:
Complete Stations 3–4 as homework or in a second session.

Fast finishers:
Complete advanced options in Stations 2 or 3.
Propose one improvement to reduce misclassification or bias.

10. Resources/Attachments

Digital Resources

- Google Colab notebook (Lesson 8).
- Teacher presentation on rules, simple neurons and decision trees.

- Interactive whiteboard or projector.
- Optional introductory video on AI decision-making.

Physical / Printable Materials

- Logic Flowchart template.
- Decision Tree worksheet.
- Station Summary Cards.
- Python Cheat Sheet.
- Reflection Questions.
- Exit Ticket.

Annexes

To be printed:

Station Summary Cards

These cards support student autonomy during the lesson and help teachers manage time and classroom flow. Each card can be printed (one per pair/group) or shown digitally.

Station 1 – Inputs, Outputs & Variables

Goal

Understand how changing a variable changes the program output.

What you do

Run the code cell.

Change the missing variable value.

Run the cell again.

Try this

Write a name.

Change the age value.

Think about

What changed in the output when you changed the variable?

Station 2 – Toy Neuron (Weights & Threshold)

Goal
See how a computer makes a simple decision using numbers.
What you do
Run the code.
Change one value (input or threshold).
Run the code again.
Try this
Increase the threshold.
Change one input from 1 to 0.
Think about
Does a higher threshold make the decision easier or harder?

Station 3 – Decision Tree (If / Else)

Goal
Understand how computers follow rules step by step.
What you do
Complete the missing condition.
Run the code.
Change the input values.
Try this
Test all possible combinations.

Think about
Which rule is checked first? Why does this matter?

Station 4 – Rule-Based Classification & Limitations

Goal

Understand how rule-based systems classify information and why they can fail.

What you do

Run the classifier.

Change the text description.

Observe the result.

Try this

Create an example that is classified incorrectly.

Think about

Why did the rule fail? Is this fair?

Teacher note:

These cards are used during the guided practice (Section 7.2, Part 3).

They are optional and can be printed or shared digitally.

Exit Ticket – Module 1: Aylin and Alara – Your AI Adventure Begins

Duration: 2 minutes

To be completed: Individually, at the end of the lesson

Student Exit Ticket

Name (optional): ______________________

Date: ______________________

1 Rule-based system

Write one example of a rule-based system (if X, then Y).

…………………………………………………………………………

2 Learning-based system

Write one example of a system that learns from data.

…………………………………………………………………………

3 Limitation
Write one limitation you noticed in today's activities.
....

Teacher notes (not visible to students)

When to use: At the very end of the lesson, after the reflection discussion (Section 8).
How: Students complete the ticket individually on paper or digitally.
Purpose: Quick formative assessment to check understanding and support future lessons.
Assessment: Not graded.

Lesson Plan 9:
Design an AI Rule System: How Machines Make Decisions

1. General Information

Course and Module:	Course 2: Delving Deeper in AI with Python and Scratch Module 2: AI Escape Room Treasure Hunt with Calculator
AI Tool / Technology Used:	Calculator, rule-based logic, numerical reasoning, thresholds, algorithmic thinking
Target Age Group:	15–18 years (Secondary education)
Subject Area:	STEM / Mathematics / Digital Skills / Artificial Intelligence Literacy
Estimated Duration:	60 minutes (optional extension: 90 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Explain how rule-based AI systems make decisions using numbers and thresholds
- Design a simple AI decision system based on numerical rules
- Test and verify decisions using a calculator
- Identify limitations and potential bias in rule-based systems
- Work collaboratively to analyse and justify AI-driven decisions

This lesson develops AI literacy by shifting students from AI users to AI system designers.

3. Competences Developed

- Digital competence: Understanding how rule-based Artificial Intelligence systems process information and make automated decisions using numerical data and logical conditions.
- Mathematical competence: Applying numerical reasoning, calculations and threshold-based decision-making to design and evaluate simple AI systems.
- Problem-solving: Designing, testing and refining rule-based decision systems while identifying possible improvements and limitations.
- Logical and critical thinking: Analysing how different inputs influence AI decisions, recognising potential bias and evaluating the fairness of algorithmic rules.
- Collaboration and communication: Working cooperatively in teams to develop, explain and justify AI decision systems using clear reasoning and evidence.

4. Prerequisites and Learning Context:

4.1 Prior knowledge required:

Students should:

- Be comfortable with basic mathematical operations
- Know how to use a calculator
- Be able to follow structured instructions and work in groups
- No prior knowledge of artificial intelligence or programming is required.

4.2 Learning Context

This lesson is designed to be implemented in a regular classroom setting using printed and teacher-guided materials.

The learning activities are based on:

- Printed Scenario Cards (Student Version)
- Teacher Support Notes
- Board or projector
- Exit tickets (optional)
- No internet connection or digital platforms are required.

5. Classroom Environment

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:
 Print the Scenario Cards and Exit Tickets
 Prepare Teacher Support Notes
 Ensure calculators are available
 Prepare the board/projector for explanations

5.2 Classroom Setup

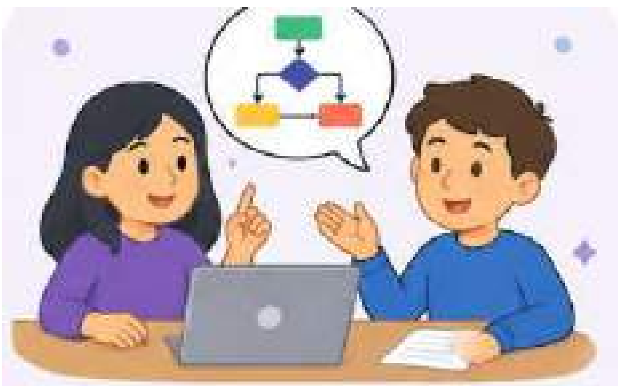
Students work in groups of 3–4.
Each group has:
 One scenario card
 One calculator
 Paper for calculations

5.3 Technical Requirements (for in-class activities)

Hardware: calculators, board or projector
Software: none
Internet: not required

6. Materials Needed

- Printed Scenario Cards (Student Version)
- Teacher Support Notes
- Calculators
- Exit Tickets (optional)



7. Step-by-Step Lesson Procedure

This lesson introduces rule-based artificial intelligence through the design and testing of simple AI decision systems. By working with numerical inputs, weights, and thresholds, students explore how machines process information and make decisions. Through practical scenarios and calculator-based testing, students identify the limitations and potential bias of rule-based systems, understanding that AI imitates logical decision-making but does not replace human judgement.

7.1. Warm-Up / Introduction (5–10 minutes)

Purpose

The purpose of this warm-up is to introduce students to the idea of AI as a rule-based decision system, and to prepare them for designing their own AI rules later in the lesson.

Teacher Introduction (3–4 minutes)

Teacher actions

The teacher introduces Artificial Intelligence as systems that:

- calculate values
- compare numbers
- follow predefined rules

The teacher explains in simple terms that many AI systems do not "think" or "understand", but instead apply rules to data in order to reach a decision.

The teacher introduces the key terms that will be used during the lesson:

Input – information given to the system

Weight – how important each input is

Threshold – the limit that determines the final decision

These terms are written on the board or shown on the projector.

Guiding Questions and Short Discussion (3–5 minutes)

The teacher asks the following guiding questions to the whole class:
Can a number change a decision?
What happens if a rule is badly designed?
Student actions
Students answer orally, with short contributions.
No correct or incorrect answers are given at this stage.

Teacher role
Encourage multiple viewpoints.
Paraphrase student answers if needed.
Avoid giving explanations yet; focus on activating prior thinking.

Transition to the Main Activity
The teacher closes the warm-up by saying:
"In this lesson, you will see how changing numbers, weights, or thresholds can completely change a decision.
Today, you will not only use AI rules — you will design them."
This sentence leads directly into the main activity.

7.2. Main Activity: Designing a Rule-Based AI System (35–40 minutes)

Purpose
This main activity guides students step by step through the design, testing, and analysis of a simple rule-based AI system, using numerical inputs and a calculator. Students work collaboratively to understand how decisions emerge from rules, weights, and thresholds.
This activity extends Module 2 by shifting students from solving predefined rules to designing and evaluating their own rule-based systems.

Part 1 –Introduction to the Task (5 minutes)

Teacher actions

The teacher explains the task clearly:
"In the previous module, you solved AI rules designed by others.
In this activity, you will design your own AI decision system."
The teacher explains that:
The system will make one clear decision (YES/NO or Option A/B)
The decision must be based only on numbers and rules
There is no single correct solution

The teacher briefly presents the structure of a rule-based AI system on the board:

- Inputs – The numbers the system uses to make a decision.
- Weights – The importance given to each input.
- Threshold – The limit that determines when a decision changes.
- Decision – The final result produced by the system.

No formulas are introduced at this stage.

Student actions

Students listen and ask clarification questions if needed.

Part 2 – Group Organisation and Scenario Selection (3–5 minutes)

Teacher actions

The teacher:

- Divides students into groups of 3–4
- Distributes one Printed Scenario Card (Student Version) to each group
- Explains that each group will work on one scenario only

The teacher reminds students:
"You must agree on decisions as a group. All choices must be justified."

What the Teacher Does and Says
Teacher Explanation (Before Testing Phase)
The teacher clearly explains:
"You are not looking for real data.
You will create your own test profiles."
"These profiles are imaginary combinations of numbers that help you check if your AI system works correctly."
The teacher adds:
"Think of different types of users.
Create one with low values, one with medium values, and one with high values."
The teacher writes on the board:
Low values (0–3)
Medium values (4–6)
High values (7–10)
The teacher emphasises:
"You are testing your rules, not trying to find the correct answer."

Student actions

Students read their scenario card together
They discuss what kind of decision their AI system must make

What Students Receive

Each group receives:

- One Printed Scenario Card (Student Version)
- One calculator

Students do not receive:

- Predefined profiles
- Correct answers
- Real data

Profiles are hypothetical and created by students to test their system.

Teacher support notes

Before presenting the full example with three inputs and different weights, the teacher may start with a simplified version using only two inputs and equal weights.

Example:

Score = (Input A × 0.5) + (Input B × 0.5)

Threshold = 5

This simplified introduction can help students understand how weighted decisions work before analysing the more complex example presented below.

Teacher Model Example (For Reference Only)

This example should not be shown at the beginning of the activity. It can be used if students need additional support or during final discussion.

Teacher Model Example

Scenario Card 3 – Music Recommendation Engine

Step 1 – Designed AI System (Example)

Selected Inputs:

Similarity to previous songs

Listening time

Artist popularity

Assigned Weights:

Similarity = 0.5

Listening time = 0.3

Artist popularity = 0.2

(Weights sum = 1.0 → optional but recommended)

Threshold: 6

Decision Rule

Score =

(Similarity × 0.5) +

(Listening time × 0.3) +

(Artist popularity × 0.2)

If Score ≥ 6 → Recommend YES

If Score < 6 → Recommend NO

Step 2 – Test Profiles (Teacher Example)

Profile 1 – Low Values

Similarity = 2

Listening time = 3

Artist popularity = 2

Score =

(2×0.5) + (3×0.3) + (2×0.2)

= 1 + 0.9 + 0.4

= 2.3

Decision: **NO**

Profile 2 – Medium Values

Similarity = 5

Listening time = 6

Artist popularity = 4

Score =

(5×0.5) + (6×0.3) + (4×0.2)

= 2.5 + 1.8 + 0.8

= 5.1

Decision: **NO (below threshold)**

Profile 3 – High Similarity, Medium Popularity

Similarity = 8

Listening time = 7

Artist popularity = 5

Score =

(8×0.5) + (7×0.3) + (5×0.2)

= 4 + 2.1 + 1

= 7.1

Decision: **YES**

Step 3 – Analysis (Model Reflection)

Possible discussion points for teacher:

Similarity has the biggest impact (weight 0.5)

A song can have good listening time but still not be recommended

The threshold (6) makes the system moderately strict

Artist popularity has less influence

Possible Limitation

If a song is new and not popular yet, the system may not recommend it even if the user would like it.

This introduces the idea of:

Bias toward popular content

Limited understanding of personal taste

Part 3: AI System Challenge (Peer Testing)

Purpose

To help students understand that rule-based AI systems can be tested, challenged, and improved.

This stage shifts students from designers to critical evaluators, reinforcing the logic explored in Module 2.

Teacher Actions

The teacher explains:

"In Module 2, you solved rule systems created by others.

Now, another group will test yours."

Each group exchanges their AI system (inputs, weights, threshold) with another group.

The teacher instructs:

"Your goal is to stress-test their AI.

Try to find a profile that makes their system behave in an unexpected way."

Time limit: 5–7 minutes.

The teacher circulates and asks:

"Why did you choose this extreme value?"

"What are you trying to prove?"

Student Actions

Students:

- Analyse another group's AI rule system.
- Create one challenging test profile.
- Calculate the result.
- Present their findings to the original group.

Original group:

- Explains why their system produced that result.
- Decides whether to adjust weights or threshold..

Student Actions

Students:

- Analyse another group's AI rule system.
- Create one challenging test profile.
- Calculate the result.
- Present their findings to the original group.

Original group:

- Explains why their system produced that result.
- Decides whether to adjust weights or threshold.

Teacher Notes

Look for:

- Groups that copy values randomly (encourage intentional testing).
- Groups that immediately change everything (encourage reasoning first).
- Interesting failures that can be shared in plenary.

Do NOT:

- Judge which system is "better".
- Give the correct weight distribution.

Key Learning Link

This stage mirrors the Escape Room logic:
Students previously tested systems created by the teacher.
Now they test systems created by peers.
They experience how rule-based systems behave under pressure.

Part 4 –AI vs Reality Round

Purpose

To highlight the limitations of purely numerical decision systems and connect AI logic with real-life judgement.

Teacher Actions

The teacher explains:

"Your AI works with numbers.
But does it understand context?"

The teacher presents 2 quick "real-life twists" related to the scenario.

Example (Music):

"The song is new and has no popularity data yet."
"The user recently changed their music taste."
Students have 30 seconds to answer:
What would your AI decide?
Do you agree with that decision?
The teacher invites quick reactions (hands raised, short answers).

Student Actions

Students:

- Compare AI output with their own intuition.
- Identify where the system oversimplifies reality.
- Suggest one possible improvement.

Teacher Notes

- Encourage fast responses.
- Do not turn this into a long ethical debate.
- The goal is awareness, not moral judgement.

If discussion becomes abstract, redirect to:
"What number would need to change for the decision to be different?"

Possible Student Reaction Patterns

Students may say:
1 “Yes, I agree — the numbers are clear.”
2 “No, because real life is more complex.”
3 “It depends — we would need more data.”
4 “Our system cannot detect new trends.”
5 “The AI doesn’t understand emotions.”

Teacher Guidance Notes

If students say “I don’t know” or remain silent, the teacher can ask:
“If the popularity is zero because the song is new, should that mean it is bad?”
“Can numbers measure personal taste?”
“What is missing from your system?”
If students all agree with the AI decision, ask:
“Would you trust this system 100% in real life?”
“Who would be responsible if the decision is wrong?”

Important Teacher Reminder

The goal is NOT to criticise AI.

The goal is to help students see:

- AI is consistent.
- AI depends on input design.
- AI cannot interpret context unless programmed to.

Key Learning Link

This stage reinforces a core concept of Module 2:

AI systems follow rules consistently, but consistency does not equal understanding.

Note for Teachers:

Encourage students to justify every rule, weight and threshold they create rather than focusing only on obtaining the "correct" result. Remind them that AI systems reflect the decisions made by their designers, and that even simple rule-based systems can produce unfair or biased outcomes if the rules are poorly designed. Promote discussion, experimentation and iterative improvement throughout the activity.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Discussion

The teacher gathers the class after the group work.
Instead of asking all groups to present their system, the teacher invites brief reflections from different groups, focusing on what they discovered during testing.
The teacher may ask:

- "Did your AI system behave exactly as you expected?"
- "Was there a moment when the result surprised you?"
- "If you could change one thing in your system, what would it be?"

The teacher selects 2–3 groups to share a short comment.
The discussion remains short and focused.
There is no correction of answers. The emphasis is on reflection rather than evaluation.

Student actions

- Groups briefly discuss their experience before speaking.
- One student per selected group shares one insight.
- Other students listen and compare with their own experience.
- Students recognise that different rule designs led to different outcomes.

Purpose

This discussion helps students move from "doing" to "understanding".
It allows them to:

- Reflect on how numerical rules influenced decisions.
- Recognise the impact of weights and thresholds.
- Understand that rule-based systems can behave differently depending on design choices.

Teacher Note

- If students give very short answers, the teacher may gently prompt them to explain what changed when they modified a value.
- The aim is not to debate whether AI is good or bad, but to help students realise that AI systems reflect the rules created by humans.

8.2 Practical Exercise / Student Output

Purpose
To define clearly what students are expected to produce by the end of the lesson and to provide evidence of understanding of rule-based AI systems.

Student Outputs

By the end of the lesson, students should have:

A completed Scenario Card with:

- 3 selected inputs
- Assigned weights
- A defined threshold
- A written decision rule
- Three tested profiles (low, medium, high values) with calculated results
- One identified limitation of their AI system

These materials serve as evidence that students understand how rule-based systems operate and how numerical factors influence decisions.

Exit Ticket (2–3 minutes)

When

At the very end of the lesson, after the reflection discussion.

How

Individually, on paper.

What students do

Students answer three short prompts:

- What is one input your AI system used?
- What does the threshold control?
- Why can a rule-based AI system produce unexpected results?

Teacher Use

Collected as quick formative feedback.

Not graded.

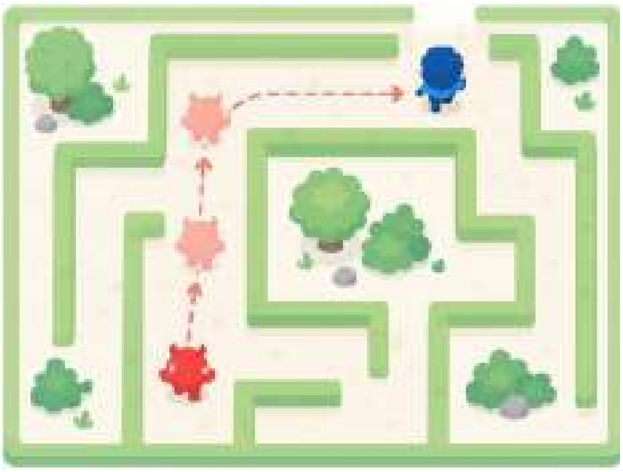
Used to check whether students understand:

- The role of inputs and weights
- The impact of thresholds
- The limitations of rule-based AI

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.
It combines:

- a short quiz,
- teacher observation,
- student self-assessment.



Quick Quiz (5 questions)

When: At the end of the lesson or at the beginning of the next session.
How: Individually (paper or digital).
Answers: Provided in the teacher notes / answer key.

Teacher instructions:

- Distribute the quiz to students.
- Allow approximately 5 minutes to complete it.
- Review answers briefly with the class or collect for quick checking.
- What is a rule-based AI system?
- What role do weights play in a decision system?
- What does the threshold control in an AI system?
- Why can changing one input value affect the final decision?
- Give one limitation of rule-based AI systems.

Possible Answers:

- A system that makes decisions using predefined numerical rules.
- Weights determine how important each input is in the final calculation.
- The threshold determines when the system changes from one decision to another (e.g. YES/NO).
- Because the final result depends on the weighted sum of inputs.
- It may oversimplify reality, ignore context, or produce unfair results if poorly designed.

Teacher Observation Checklist

Purpose: Support teacher observation during the practical activity.
How it is used:
The checklist is completed during the main group activity (designing and testing the AI rule system)..
Teachers observe pair work, not individual performance in detail.
When:
Filled in progressively during the lesson.
After the lesson:
The checklist is used by the teacher for reflection and planning.
Results do not need to be shared formally with students, but key points may be discussed informally.

Checklist
[ ] Selects relevant inputs logically.
[ ] Assigns weights with reasoning.
[ ] Defines a clear threshold.
[ ] Tests at least three different profiles.
[ ] Explains how a numerical change affects the decision.
[ ] Identifies at least one limitation.
[ ] Collaborates effectively within the group.

Student Self-Assessment

Purpose: Encourage student reflection on their own learning.
When:
Completed individually at the end of the lesson (together with the exit ticket).
How:
On paper or digitally.
What students do:
Rate their understanding (1–5).
Write one short reflective sentence.
After completion:
Students keep the self-assessment for personal reflection.
The teacher may briefly discuss common difficulties in the next lesson.

- Rate 1–5: I can explain how a rule-based AI system makes decisions.
- Rate 1–5: I understand how weights and thresholds influence results.
- One sentence: What is one limitation of rule-based AI systems?

9. Adaptations

For Diverse Learners

Diverse learners refers to students with different learning speeds, confidence levels, and prior digital experience.

Support Strategies

- For students who need additional support:
- Allow them to use only 2 inputs instead of 3.
- Provide suggested example weights (e.g. 0.5 / 0.3 / 0.2).
- Guide them to test simple profiles first (clear low vs high values).
- Offer closer teacher guidance during the testing phase.

- Pair them with a peer who can support calculations.
- The focus should remain on understanding how the system works rather than on complex calculations.

Time Constraints

If time is limited:

- Reduce testing to 2 profiles instead of 3.
- Skip the peer challenge phase (Part 3).
- Leave the AI vs Reality Round for the next session.
- The core objective (design + test + reflect) should still be completed.

Fast Finishers

Students who finish early may:

- Add a fourth input to their system.
- Compare two different thresholds and analyse the difference.
- Redesign their system to make it “fairer” or more balanced.
- Create a new scenario and challenge another group.
- These extensions deepen understanding without changing the structure of the lesson.

10.Resources/Attachments

Digital Resources

- Course 2 Master Document (Lesson 9).
- AI Rule System Calculator (Google Sheets, Excel or digital calculator).
- Teacher presentation on rule-based Artificial Intelligence systems.
- Interactive whiteboard or projector for demonstrations.
- Optional online simulator or spreadsheet for testing different weights and thresholds.

Physical / Printable Materials

- AI Rule Design Worksheet.
- Scenario Cards (e.g., Scenario Card 1 – Viral Video Booster and the remaining scenario cards).
- Decision Rule Template.
- Weight and Threshold Recording Sheet.
- Reflection Questions.
- Exit Ticket.

Annexes

To be printed:

Scenario Card 1 – Viral Video Booster

Your AI must decide:

Boost the video → YES or NO

You are designing an AI system for a social media platform that decides whether a video should be promoted to more users.

Available inputs (0–10):

Watch time

Number of likes

Number of shares

Comments activity

Step 1 – Design Your System
Choose 3 inputs
Assign weights
Define a threshold
Write your decision rule (e.g. If score ≥ threshold → YES)

Step 2 – Create Test Profiles
Create 3 imaginary video performance profiles:
Low engagement (0–3)
Medium engagement (4–6)
High engagement (7–10)
Only include the inputs you selected.

Example Format
Profile 1
Input 1 = ___
Input 2 = ___
Input 3 = ___

After each profile:

- Calculate the result
- Compare with your threshold
- Decide: Should the video be boosted?

Step 3 –

Analyse Your System

- Did popularity dominate the decision?
- Could this system promote low-quality content?
- What would you adjust?

Scenario Card 3 – Music Recommendation Engine

Your AI must decide:

Recommend this song → YES or NO

You are designing a music streaming recommendation system.

Available inputs (0–10):

Similarity to previous songs

Listening time

Artist popularity

Friend recommendations

Step 1 – Design Your System

Choose 3 inputs

Assign a weight to each input

Define a threshold

Write your decision rule (e.g. If score ≥ threshold → YES

Step 2 – Create Test Profiles

You must create 3 imaginary listener profiles.

These are not real people.

They are numerical combinations used to test your AI system.

Create:

One profile with mostly low values (0–3)

One profile with medium values (4–6)

One profile with high values (7–10)

Example format:

Profile 1

Similarity = __

Listening time = __

Artist popularity = __

After creating each profile:
Calculate the result
Compare it with your threshold
Decide: **Does your AI recommend the song**?

Scenario Card 4 – AI Fitness Challenge Coach
Your AI must decide:
Increase training intensity → YES or NO
You are designing an AI fitness app.

Available inputs (0–10):
Energy level
Sleep quality
Previous performance
Heart rate recovery

Step 1 – Design Your System
Choose 3 inputs
Assign weights
Define a threshold
Write your decision rule (e.g. If score ≥ threshold → YES)

Step 2 – Create Test Profiles
Create 3 imaginary daily user conditions:
Low readiness (0–3)
Medium readiness (4–6)
High readiness (7–10)
Only include selected inputs.

Example Format

Profile 1

Input 1 = ___

Input 2 = ___

Input 3 = ___

After each profile:

Calculate the result

Compare with threshold

Decide: Should intensity increase?

Step 3 – Analyse Your System

Could this system push someone too hard?

Does it consider how someone feels?

What are its limitations?

Lesson Plan 10: Scratch Meets Artificial Intelligence

1. General Information

Course and Module:	Course 2: Delving Deeper in AI with Python and Scratch Module 3: Scratch Meets Artificial Intelligence
AI Tool / Technology Used:	Scratch 3.0 (scratch.mit.edu) + Machine Learning for Kids (ML4K) + Text-to-Speech Extension
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Computer Science / AI Literacy
Estimated Duration:	90 minutes (optional extension: 120 minutes)

2. Learning Objectives (Outcomes)

By the end of this lesson, students will be able to:

- Navigate the Scratch interface and identify its main elements: Stage, Code Area, Sprite Area, and Block Palette.
- Create a basic Scratch project by adding sprites, backdrops, and connecting blocks from different categories.
- Explain what a Scratch AI extension is and why AI extensions require an internet connection.
- Train a simple text classification model using Machine Learning (ML) for Kids (ML4K) and connect it to a Scratch project.
- Add and configure the Text-to-Speech extension to build a project in which a sprite speaks and responds to user input.
- Reflect on how trained ML models differ from traditional block-based programming and identify real-world parallels.

This lesson focuses on helping students understand how rule-based AI systems make decisions by designing, testing and evaluating their own algorithms. Through practical scenarios, students explore how inputs, rules, weights and thresholds shape AI behaviour while reflecting on fairness, bias and responsible AI design.

3. Competences Developed

- Digital competence: using Scratch, training a cloud-based ML model (ML4K), and integrating AI extensions into creative projects.
- Problem-solving: debugging block scripts, improving model accuracy by adding training examples, and adjusting Text-to-Speech settings.
- Critical thinking: reflecting on the difference between rule-based programming and machine learning, and identifying limitations of trained models.

- Collaboration: working in pairs using Driver/Navigator roles to build and test Scratch projects.
- Creativity: designing an original interactive project that combines ML classification with spoken responses.

4. Prerequisites and Learning Context

4.1 Prior Knowledge Required

Students are expected to have:

- Basic digital skills: using a browser, creating an account on an online platform, and navigating a web-based tool.
- A foundational understanding of what AI is, gained from Course 1: A Digital Primer: Artificial Intelligence Essentials (Modules 1–3) or an equivalent introduction.
- Familiarity with machine learning at a conceptual level — i.e., that models learn from examples rather than fixed rules.
- No prior programming experience is required. Scratch is designed for first-time coders, and all ML4K activities are guided step by step.

4.2 Learning Context

- This lesson is part of Course 2 — Delving Deeper in AI with Python and Scratch, Module 3: Scratch Meets Artificial Intelligence.
- The lesson connects theoretical AI concepts from Course 1, particularly machine learning, data training, and AI applications to hands-on creative practice. Students move from understanding what AI does to actively building a small AI-powered project.
- The session can be delivered as a single 90-minute lesson or split into shorter sessions: 1 session (or more if needed) covering Scratch basics and project setup (Units 3.1–3.3), and 1 session (or more if needed) focusing on AI extensions (Units 3.4–3.6).

5. Classroom Environment

5.1 Pre-Lesson Preparation

Before the lesson, the teacher should:

- Test access to Scratch (https://scratch.mit.edu) and Machine Learning for Kids (https://machinelearningforkids.co.uk) on school computers.
- Verify that the Text-to-Speech extension loads correctly in Scratch (requires an active internet connection).
- Create a simple demo Scratch project with the Text-to-Speech extension enabled, to show students during the introduction.
- Familiarise themselves with the ML4K sentiment classifier workflow (Steps 1–9 in Unit 3.5) so they can support students effectively.
- Prepare a short slide deck or visual overview of the Scratch interface elements for the warm-up.
- (Optional) Pre-create a teacher ML4K account and a sample project so students can see a trained model in action before building their own.
- Ensure all student devices have a working microphone or webcam if the Video Sensing extension will be explored as an extension activity.

5.2 Classroom Setup

- Students work in pairs, sharing one device, to promote collaboration and discussion.
- Pair-programming roles are used throughout the lesson:
- Driver: operates the keyboard and mouse, types code, and interacts with the tools.
- Navigator: reads instructions, suggests what to do next, and checks that the logic makes sense.
- Roles should be swapped at least once during the main activity (recommended after completing Station 2) to ensure active participation from both students.
- The classroom should allow the teacher to move freely between pairs. A projector or interactive whiteboard is required for demonstrations and the warm-up discussion.

5.3 Technical Requirements (for In-Class Activities)

Hardware:
- Computers or laptops with a working keyboard and mouse (one per pair).
- Projector or interactive whiteboard for teacher demonstrations.
- Webcam (optional; required only if Video Sensing extension activities are used).

Software:
- Modern web browser (Chrome, Edge, or Firefox recommended).
- Scratch 3.0: https://scratch.mit.edu
- Machine Learning for Kids: https://machinelearningforkids.co.uk

Internet connection:
- Required throughout the lesson. AI extensions process data on cloud servers and cannot function offline.

Backup plan:
- If internet access is unreliable, fall back to the block-based Scratch activities only (Units 3.1–3.3), which do not require internet once the page has loaded.
- If ML4K cannot be accessed, the teacher demonstrates the trained model on the projector and students focus on the Text-to-Speech activity only.

6. Materials Needed

6.1 Digital Resources

- Module 3 "Scratch Meets Artificial Intelligence" provided as reading and reference material for students.
- Scratch platform: https://scratch.mit.edu
- Machine Learning for Kids: https://machinelearningforkids.co.uk
- Teacher slide deck covering the Scratch interface, AI extensions overview, and the ML4K workflow (prepared by the teacher based on Module 3 content).
- Scratch Machine Learning Studio for project inspiration: https://scratch.mit.edu/studios/3995548/ (Optional)

6.2 Physical Materials

- Printed or digital Station Instruction Cards (one set per pair) - prepared by the teacher based on Units 3.5 and 3.6.
- Printed glossary of Scratch block categories with colour-coding and example uses (Optional)
- Quiz for formative assessment (see Annex 1)
- Exit Ticket / Self-Assessment form (see Annex 3)

7. Step-by-Step Lesson Procedure

The timing below is indicative for a 90-minute lesson. Teachers may shorten or extend individual parts depending on student needs and available time.

7.1 Warm-Up / Introduction (10 minutes)

Teacher Actions
Opening question (5 min): "Have you ever talked to a computer or phone and had it understood you? Or seen a machine recognise what you draw or say? Where do you think that comes from?"

Collect 2–3 examples from students (e.g., Siri, Google Translate, spam filters, face unlock on phones) and briefly connect them to the idea that these tools are trained on examples rather than programmed with fixed rules.

Key terms introduced in simple language (3-5 min):

Scratch: a visual block-based programming language where code blocks snap together like puzzle pieces — no typing of syntax required.
Extension: an add-on module that gives Scratch new capabilities, including AI-powered ones.
Machine Learning: a type of AI where the computer learns to recognise patterns from examples you provide, rather than following fixed rules you write.

Text-to-Speech: an AI technology that converts written text into spoken audio.
Lesson goal (2 min):
"Today you will build your own AI-powered Scratch project. You will train a small machine learning model that understands feelings in text, and add a voice so your sprite can speak. By the end, your Scratch sprite will listen, understand, and respond out loud."
Mini demo- optional (2 min):
The teacher opens a pre-built Scratch project and demonstrates the Text-to-Speech extension in action to motivate students and set expectations.

Class Organisation
Whole-class discussion, then transition to pair work at computers. Driver/Navigator roles assigned before students open computers.

7.2 Main Activity — Scratch and AI in Action (65 minutes)

Suggested station timing:

- Station 1 — 15 minutes
- Station 2 — 30 minutes
- Station 3 — 20 minutes (optional extension)

If time is limited, Stations 1 and 2 form the core of the lesson. Station 3 can be used for fast finishers or set as homework.

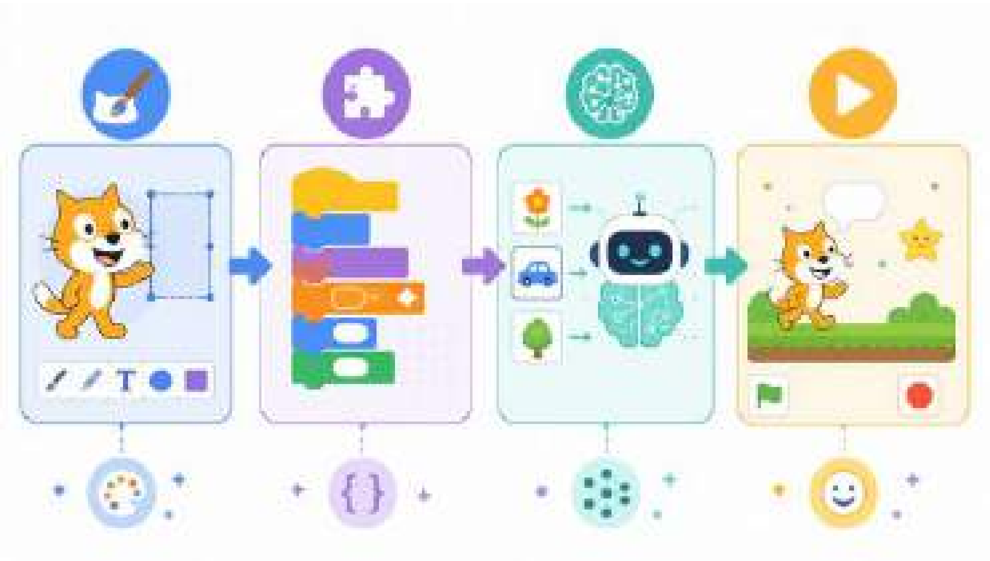
First-time account creation and ML model training may require additional time (5-10 minutes) depending on internet speed and student familiarity with online platforms. The teacher should be prepared to extend the lesson or continue in a follow-up session if needed.

Part 1 — Teacher Introduction: Key Concepts (5 minutes)

Before students open their computers, the teacher provides a structured overview using the slide deck:
The Scratch interface: Stage (what the audience sees), Code Area (where scripts are built), Sprite Area (characters and objects), Block Palette (colour-coded categories).
How AI extensions work: extensions add new block categories; AI extensions require internet because processing happens on cloud servers.
The ML4K workflow in three steps: Train (provide examples) → Learn (the model finds patterns) → Make (use the model in Scratch).

Teacher Guidance:

- Explain that students will work through three stations in pairs. Each station builds on the previous one, ending with a complete AI-powered talking Scratch project.
- Emphasize that mistakes during model training are expected and valuable learning opportunities.
- Clarify success criteria: students do not need a perfect model but a working workflow.



Part 2 — Teacher Instructions: How Students Will Work

The teacher explains the workflow (3-5 min):

- Each pair works through Stations 1–3 in order, using the Station Instruction Cards.
- If an error occurs, students should: re-read the instruction, check for missing blocks, compare with another pair, then ask the teacher.
- Driver/Navigator roles are active throughout. Roles swap after Station 2 is complete.
- Teacher monitor collaboration balance and intervene if one student dominates the activity.

Station 1 — Getting Started with Scratch (15 minutes)

Objective: Navigate the Scratch interface and create a basic working script.

Student tasks:

- Open Scratch at https://scratch.mit.edu and create an account (or log in).
- Identify the Stage, Code Area, Sprite Area, and Block Palette. Name one block from at least three different colour-coded categories.
- Choose a sprite and build the basic movement and sound script from Unit 3.3 (when green flag clicked → move 50 steps → play sound → say a message for 2 seconds). Test it.
- Change the backdrop and customise the message the sprite says.

Quick reflection question: **"**Which block category did you use to start the program when the green flag was clicked? Why is this called an Event?"

Teacher role:

- Circulate and check that all pairs have a working project before moving to Station 2. Verify every pair successfully runs a script triggered by the green flag.
- For pairs who finish early: ask them to add a loop so the sprite keeps moving until the red stop button is clicked.

Station 2 — Train an AI Model and Connect it to Scratch (30 minutes)

Objective: Train a text sentiment classifier in ML4K and integrate it into a Scratch project that gives different spoken responses based on whether the user types a positive or negative message.
This station follows the step-by-step workflow from Unit 3.5 of Module 3.

Student tasks:
Working in Driver/Navigator pairs, students have to:

- Create a new ML4K project and define at least two sentiment labels (e.g., Positive and Negative).
- Add a range of training examples for each label, including short phrases and full sentences.
- Train the model and test predictions using new example phrases.
- Improve model performance by adding additional or more varied examples if classifications are inaccurate.
- Connect the trained ML model to Scratch.
- Program conditional responses using Scratch blocks so that the sprite reacts differently depending on the classification result.

Students should test their project several times using unexpected or ambiguous phrases.

Quick reflection questions:
"Did your model always classify messages correctly? What happened when the phrasing was unusual?"
"What is the difference between the if-then blocks you wrote in Scratch and the AI model doing the classifying?"

Teacher role:

- The teacher circulates continuously to support technical progress and conceptual understanding.
- Remind pairs to swap Driver/Navigator roles after the model is trained.
- For pairs with low model accuracy: prompt them to add more varied training examples.
- If students experience difficulties, prompt troubleshooting rather than provide direct solutions.

Station 3 — Add a Voice: Text-to-Speech Extension (20 minutes)

Objective: Students enhance their AI Scratch project by replacing text speech bubbles with spoken responses using the Text-to-Speech extension. This activity transforms the AI interaction into a multimodal experience and completes the final stage of the practical task.
This station follows Unit 3.6 of Module 3:

Student tasks:

- Working in Driver/Navigator pairs, students have to:
- Open the Scratch project created during Station 2.
- Add the Text-to-Speech extension.
- Replace existing "say" blocks with spoken output using the appropriate Text-to-Speech blocks.
- Configure different spoken responses for Positive and Negative classifications.
- Test the interaction several times using different user inputs.
- Adjust volume, voice, or phrasing to improve clarity and user experience.

Students should confirm that the sprite produces an audible response for at least two different classification outcomes.

Optional extension for fast finishers:

- Add a ‘forever’ loop so the sprite keeps asking for input rather than stopping after one response.
- Add a third label to the ML4K model (e.g., ‘Neutral’) and create a third response branch in Scratch.
- Use the Translate extension to make the sprite respond in a language different from the one the user typed in.

Teacher role:

- Circulate and ensure all pairs hear at least one spoken response from their sprite before the activity ends.
- Prompt students to think about why voice makes an AI interaction feel more natural.
- If technical issues occur, encourage students to verify block connections rather than rewriting code immediately.

7.3 Reflection / Discussion (10 minutes)

- Teacher-led whole-class discussion.
- Purpose: Consolidate learning, encourage critical reflection, and connect the practical experience to broader AI concepts.

Guided discussion questions:

- What surprised you most about how the ML model worked? Did it ever make mistakes you did not expect?"
- "What is the difference between the if-then blocks you wrote in Scratch and the part where the AI decided whether the message was positive or negative?"
- "What would happen if your training examples were all from one person's writing style? Would the model work equally well for everyone?"
- "Where could a tool like this — that understands feelings in text and responds with a voice — be used in real life?"
- "If a company used AI to automatically read employees' emails to detect if they were happy or unhappy at work, would that be acceptable? Why or why not?"
- "Your model made mistakes when classifying some messages. What could go wrong in real life if a sentiment analysis system used in customer service or social media moderation misunderstands someone's emotions?"

Note for Teachers:

Students may focus primarily on making their Scratch project work correctly. Encourage them to go beyond programming by discussing why the AI behaves as it does and how different rules influence its decisions. Emphasise that even simple Scratch projects illustrate fundamental AI concepts such as decision-making, algorithms and logical reasoning. Promote creativity, experimentation and collaborative problem-solving throughout the activity.

8. Reflection and Assessment

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Practical Exercise / Student Output

Expected student outputs:

- A working Scratch project (saved to their account) in which: a sprite asks the user a question; an ML4K text classifier determines whether the response is Positive or Negative; the sprite gives a different spoken response using the Text-to-Speech extension depending on the classification.
- A short, written answer (2–3 sentences per pair) describing: what training examples they used, one situation where the model classified text incorrectly, and what they would do to improve it.

8.2 Assessment

Assessment is formative and takes place during and immediately after the lesson. It combines:

- A short quiz
- Teacher observation (Checklist)
- Student self-assessment (Exit ticket)

Short Quiz

See Annex 1
When: At the end of the lesson or at the beginning of the next session.
How: Individually, on paper or digitally
Teacher instructions:

- Distribute the quiz to students (included at the end of this lesson plan).
- Allow approximately 10 minutes to complete it.
- Review answers briefly with the class or collect for quick checking. Answers are not graded; they serve as formative feedback.
- The quiz can be done on paper, in a shared document, or using online tools (e.g., Kahoot, Google Forms). Answers are not graded; they are used as quick formative feedback to guide the next steps of teaching.

Teacher Observation Checklist:

See Annex 2
Purpose: Support teacher observation during the group activity and presentations.
How it is used:

- The checklist is completed during group work and presentations, based on group observation.
- Focuses on collaboration, technical progress, and ability to explain what the pair is doing.
- Filled in progressively during the lesson.
- Used by the teacher for reflection and future planning; not shared formally with students.

Student Self-Assessment (Exit Ticket)

See Annex 3
At the end of the lesson, students complete a short self-assessment individually (on paper or digitally).
When: At the very end of the lesson, after the reflection discussion.
How: Individually, on paper or digitally.

- **Teacher instructions:**
- Distribute a 1 exit ticket per student.
- Allow approximately 5 minutes to complete it.
- Review answers or collect for quick checking.

8.3 Informal Success Criteria

By the end of the lesson, the student is considered to have met the learning objectives if they can:

- Identify at least four Scratch block categories by name and colour and explain what each is used for.
- Explain in simple terms the three-step ML4K workflow: training examples → train model → use in Scratch.
- Demonstrate a working Scratch project where the sprite's response changes based on the ML model's classification of user input.
- Explain one reason why their model may have made a mistake and describe one way to improve it.
- Identify at least one real-world parallel for the AI capability they used (e.g., spam filters, sentiment analysis in customer service).

These criteria are not formally assessed or graded. They are used by the teacher to evaluate students' understanding during pair work and to identify students who may need additional support.

9. Adaptations for Diverse Learners

Diverse learners refer to students with different prior coding experience, learning speeds, and digital confidence.

Support strategies:

- Provide a printed glossary of Scratch block categories with colour-coding and one example use for each.
- Offer a partially pre-built Scratch project (sprite and variable already created) so students who struggle with setup can focus on the ML logic.
- Allow students who find Station 2 challenging to skip the ML4K integration and focus only on the Text-to-Speech extension, building a simpler interactive talking sprite.
- Provide step-by-step screenshot guides for the ML4K workflow (available in the Module 3 educational materials).

For students with lower digital confidence:

- Pair them with a more experienced classmate and assign them the Navigator role first.
- Provide a pre-created ML4K project with labels already set up; students only need to add training examples and train the model.

Fast finishers:

- Add a 'Neutral' third label to the ML model and create a third response branch in Scratch.
- Add a 'forever' loop and an option to type 'quit' to end the conversation.
- Use the Translate extension to make the sprite respond in a different language from the one the user typed in.
- Explore the Video Sensing extension and brainstorm a project idea combining motion detection with spoken responses.

Time constraints:

- If only 60 minutes are available, focus on Stations 1 and 2. Station 3 can be assigned as a homework.
- If Station 2 takes longer than expected, students can demonstrate the ML model working within ML4K (Step 6) without completing the full Scratch integration.

10. Resources / Attachments

Core Learning Materials

- Module 3 "Scratch Meets Artificial Intelligence" - reading and reference material for students
- Scratch platform: https://scratch.mit.edu
- Machine Learning for Kids: https://machinelearningforkids.co.uk
- Teacher slide deck covering the Scratch interface, AI extensions overview, and the ML4K workflow
- Scratch Machine Learning Studio: https://scratch.mit.edu/studios/3995548/ - Optional

Printable Documents (as Annexes):

- Annex 1: Short Quiz
- Annex 2: Teacher Observation Checklist
- Annex 3: Student Self-Assessment (Exit ticket)

Annex 1

Short quiz

The quiz can be done on paper, in a shared document, or using online tools (e.g., Kahoot, Google Forms). Answers are not graded; they are used as quick formative feedback to guide the next steps of teaching.

1. Which statement best describes Scratch?

a) A textual programming language used by professional software engineers.
b) A visual block-based programming language that makes coding accessible to beginners.
c) An artificial intelligence platform for building machine learning models.
d) A hardware robot kit for teaching programming to children.

2. Why do AI extensions in Scratch require an active internet connection?

a) So that the teacher can monitor student progress remotely.
b) Because Scratch needs to download new blocks every time an extension is used.
c) Because internet speeds affect how fast the sprite moves on the stage.
d) Because the AI processing takes place on cloud servers, not on the student's computer.

3. What must you do in Machine Learning for Kids before you can use an AI model in your Scratch project?

a) Nothing — the platform comes with pre-trained models for all purposes.
b) Write Python code to define the classification algorithm.
c) Provide training examples for each category, train the model, then connect it to Scratch.
d) Purchase a subscription to access the Scratch integration.

4. Your ML4K model keeps classifying ‘I am a bit tired today’ as Negative. What is the best fix?

a) Delete the model and recreate it from scratch.

b) Add more varied examples of mild or neutral phrases to the relevant label and retrain.

c) The Text-to-Speech extension is interfering with the classifier.

d) The model can only understand feelings written in capital letters.

Annex 2

Teacher Observation Checklist

Use this checklist to observe student groups during the lesson. Tick (✓) when the criterion is met. This is a group-based observation tool; individual student performance is not recorded in detail.

Pair Names	Sets up Scratch and accesses ML4K without help	Trains ML model with sufficient examples	Connects ML model to Scratch and builds a working script	Adds Text-to-Speech and customises voice / language	Can explain what the AI does in their project	Notes
Pair 1:	☐	☐	☐	☐	☐	
Pair 2:	☐	☐	☐	☐	☐	
Pair 3:	☐	☐	☐	☐	☐	
Pair 4:	☐	☐	☐	☐	☐	
Pair 5:	☐	☐	☐	☐	☐	
Pair 6:	☐	☐	☐	☐	☐	
Pair 7:	☐	☐	☐	☐	☐	
Pair 8:	☐	☐	☐	☐	☐	

General observations:

Annex 3

Student Exit Ticket

Name (optional): ______________________________ Date: _______

1. Name one Scratch block category and describe what it is used for.
Category: ____
What it is used for: ____

2. Describe in 1–2 sentences how Machine Learning for Kids and Scratch work together in the project you built today.
____
____
____

3. Rate yourself from 1 to 5: “I can explain how an ML model is trained and why training data quality matters.”
1 □ — 2 □ — 3 □ — 4 □ — 5 □

4. Rate yourself from 1 to 5: “I can build a Scratch project that uses an AI extension to respond to user input.”
1 □ — 2 □ — 3 □ — 4 □ — 5 □

5. Write one question you still have about Scratch, machine learning, or the Text-to-Speech extension.
My question: ____
____

Lesson Plan 11: Python Game Programming – Catch Me If You Can: Simple Game AI in Action

1. General Information

Course and Module:	Course 2: Delving Deeper in AI with Python and Scratch Module 4: Python Game Programming with AI (Pygame / Arcade Libraries)
AI Tool / Technology Used:	Python + Pygame or Arcade library + Rule-based AI logic (conditions, distance-based behaviour, simple feedback)
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT / Computer Science / STEM
Estimated Duration:	90 minutes (optional extension: 120 minutes)

2. Learning Objectives (Outcomes)

- Explain in simple language the role of the game loop and screen update in a Python game.
- Implement movement of a player controlled by arrow keys using Pygame or Arcade.
- Describe and test a basic game AI logic where an enemy follows the player using distance.
- Adjust parameters (speed, reaction distance) and interpret their effect on the "intelligent" behaviour of the enemy.
- Reflect on how this logic is similar to or different from "real AI" in modern games.

This lesson introduces students to simple rule-based AI in video games through hands-on programming with Python and a game library, helping them understand how "intelligent" behaviour can emerge from basic logical rules.

3. Competences Developed

- Digital competence – basic Python programming, use of a game library (Pygame/Arcade)
- Problem-solving – debugging simple errors, tuning parameters to reach a desired behaviour
- Critical thinking – analysing whether the observed behaviour is "intelligent" or just simple rules
- Collaboration – pair programming and joint testing of the game.
- Creativity – experimenting with variations of the game (colours, speeds, game over conditions)

4. Prerequisites and Learning Context

4.1 Prior knowledge required

Students are expected to have basic digital skills, such as opening files, typing text, and running pre-written programs on a computer.
They should be familiar with very basic programming concepts, including:

- the idea of a variable,
- simple conditional logic (if / else),
- running and modifying pre-prepared Python code.

No prior experience with game development or artificial intelligence is required.
All activities are based on starter code, and students are not expected to write full programs from scratch.

4.2 Learning Context

- This lesson is part of Module 4 – Python Game Programming with AI, within the course Delving Deeper in AI with Python.
- This lesson directly uses the "Catch Me If You Can" game code developed in Course 2 Module 4. Students apply concepts introduced in Course 1 (such as decision-making, rule-based systems, and the sense-decide-act cycle) in a practical game context.

- The lesson can follow Module 1 (Aylin & Alara - decision trees, rule-based systems), Module 2 (AI Escape Room - probability and simulation) and Module 3 (Scratch Meets Artificial Intelligence) helping students see the connection between abstract AI principles and concrete game behaviours.
- The lesson focuses on applying previously introduced AI concepts—such as decision-making and rule-based logic—in a practical and motivating context: video game development.
- Students explore how simple AI behaviours used in games (e.g. chasing, reacting, adapting) are created through programming rules, helping them understand that “intelligent” game characters are usually driven by predefined logic rather than true learning systems.
- This lesson prepares students for later modules that introduce more advanced AI behaviours, such as randomness, scoring systems, and learning strategies.

5. Classroom Environment

5.1 Pre-lesson Preparation

Before the lesson, the teacher should:

- Ensure that Python and the selected game library (Pygame or Arcade) are installed and working.
- Test the starter game code in advance.
- Prepare a projector or interactive whiteboard for a short demonstration.
- Check that all student devices can run the game correctly.
- Familiarise themselves with the AI behaviour implemented in the starter code.
- (Optional) Prepare an example of a completed or partially completed game to show expected results.

Note: The complete source code is provided separately as part of the Course 2 educational resources and is not included in this lesson plan.

5.2 Classroom Setup

Students work mainly in pairs, sharing one device, to promote collaboration and discussion.

Pair-programming roles are recommended:

- Driver: controls the keyboard and runs the code.
- Navigator: reads instructions, checks logic, and suggests improvements.

Roles should be swapped every 5–7 minutes to ensure active participation from both students.

The classroom should be organised so that the teacher can move easily between groups to provide support and guidance.

5.3 Technical requirements (for in-class activities)

Hardware

- Computers or laptops with keyboards.
- Projector or interactive whiteboard for demonstrations.

Software

- Python installed.
- Pygame or Arcade library.
- Code editor (e.g. VS Code, Thonny).
- Modern web browser (for documentation or optional online resources).

Internet connection

- Required only for initial setup, updates, or optional reference materials.
- Once installed, the lesson can be conducted offline.

Troubleshooting for Pygame/Arcade installation

- If students cannot run the game because of installation issues, the teacher can:
- Provide a pre-configured computer where the game is already running for demonstration.
- Share a zipped folder with a tested virtual environment or portable Python installation (if school policy allows).
- Use screenshot sequences or a short video of the game in action so that students can still analyse and discuss the AI behaviour even if they cannot run the code themselves.

6. Materials Needed

6.1 Digital Resources

- Base Python game file “Catch Me If You Can” (Module 4).
- Short slide deck on game loop, entities (player, enemy, target) and simple game AI.
- Access to Pygame/Arcade documentation, if possible (links provided in course materials).

6.2 Physical Materials (optional)

- Printed support sheet with a simple diagram “player – enemy – target” and space for notes.
- Printed self-assessment / exit ticket.

7. Step-by-Step Lesson Procedure

The timing below is indicative for a 90-minute lesson. Teachers may shorten or extend individual parts depending on student needs and available time.

7.1 Warm-Up / Introduction (10 minutes)

Teacher actions

- Display the question: "In which games have you seen enemies that follow you or react to your position? Is this 'artificial intelligence'?"
- Collect 2–3 student examples (popular games) and link them to the idea that there are often very simple rules running many times per second.
- Briefly present the lesson goal: to create a mini-game where an enemy follows the player using a simple AI-like logic.

Class organization: Whole-class discussion followed by transition to pair work at computers.

Student actions:

- Students share examples of games with reactive enemies.
- Students participate in short discussion about whether this behaviour is "real AI".

Key terms (introduced in simple language)

- Game loop – the cycle that updates the game many times per second.
- Entity – an object in the game (player, enemy, target).
- Distance – how far the enemy is from the player, used to decide when and how to react.

7.2 Main Activity – AI in Action (60 minutes total)

Within the 60-minute main activity (AI in Action), the suggested timing per station is:

- Station 1 – 15–20 minutes
- Station 2 – 20 minutes
- Station 3 – 15–20 minutes (optional extension)

If time is limited, Stations 1 and 2 form the core of the lesson; Station 3 can be used only if there is enough time or as homework.

Part 1 – Introduction: Teacher input on key terms

Before students begin the practical work, the teacher introduces key terms in simple language:

- Game loop – the cycle that updates the game many times per second.
- Entity – an object in the game (player, enemy, target).
- Distance – how far the enemy is from the player, used to decide when and how to react.
- Rule-based AI – the enemy follows predefined rules (if-then logic) rather than learning from experience.

The teacher explains that students will work with pre-prepared starter code and will modify only specific parameters to observe how AI behaviour changes.

Part 2 – Teacher instructions: How students will work

The teacher explains the workflow:

- Each pair works through Stations 1-3 in order.
- Students locate the indicated code section, make small changes, run the game, and observe the result.

If an error occurs, students should:

- Re-run the code,
- Check for typing errors or missing characters,
- Compare their work with another pair,
- Ask the teacher for support.

Pair-programming roles are assigned: Driver (types and runs code) and Navigator (reads instructions and checks logic). Roles should be swapped every 5-7 minutes.

Part 3 – Guided Practice: Stations 1-2
Objective: Understand how the enemy computes the direction towards the player and moves.
Student tasks:

- Find the part of the code where dx, dy and dist (e.g. using math.hypot) are calculated.
- See how the enemy position is updated using dx/dist and dy/dist.
- Change ENEMY_SPEED (e.g. from 2.5 to 4) and test the game.

Quick reflection questions:

- "When the enemy speed increases, does the game feel more 'intelligent' or just more difficult?"
- "If dist is 0, why does the code avoid dividing by zero?"

Part 4 – Station 3: Reaction distance (optional)
Objective: Introduce a reaction distance for the enemy (react_distance).
Student tasks

- Add a variable react_distance (for example 150).
- Modify the code so that the enemy only moves when dist < react_distance.
- Test different values (100, 300) and discuss the impact on behaviour.

Check-in question: "With a larger react_distance, does the enemy react earlier or later? Does this make the game feel 'smarter'?"
Teacher role: Walk around, help with syntax errors, remind students to change and test one thing at a time, and encourage short explanations of what they changed and what they saw.
Teacher support (code view)
To help teachers follow the coding steps, Annex 4 provides small screenshots of the relevant code sections before and after each key modification (player speed in Station 1, enemy chasing logic in Station 2, and reaction distance in Station 3).

7.3 Reflection / Discussion (10 minutes)+

Teacher actions: The teacher leads a whole-class discussion immediately after completing Stations 1-3. No additional coding is required in this section.
Purpose: Consolidate learning and support critical reflection after the practical activities.

Guided discussion questions

- "In your game, which parts could we call 'AI', and which parts are just fixed rules?"
- "How does this logic relate to the idea of 'sense – decide – act' we use when talking about AI systems?"
- "What improvements would make the enemy appear more intelligent (predicting movement, avoiding the player, cooperating with other enemies, etc.)?"
- "In real life, where could a similar 'chasing' or tracking behaviour be used (e.g. surveillance, advertising, recommendation systems)?"
- "In which of these cases could this raise ethical concerns (privacy, safety, unfair targeting)? Why?"

Note for teachers

Students often associate game AI with advanced Artificial Intelligence. Emphasise that the behaviours observed in this activity are generated through simple algorithms, conditions and programmed rules rather than autonomous intelligence. Encourage students to experiment with different parameters, analyse how behaviour changes and discuss how similar principles are used in many real-world AI systems.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Practical Exercise / Student Output (10 minutes)

Expected Student Outputs

A working game file in which:

- the player moves with the arrow keys,
- the enemy follows the player using distance-based logic,
- at least one parameter (speed or reaction distance) has been changed and tested.

A short-written note (2–3 sentences) per pair describing:

what they changed,

how this changed the enemy's behaviour.

Informal criteria:

- The game runs without critical errors.
- Basic chasing logic is implemented and visible.
- Students can explain in simple terms what the relevant code section does.

8.2 Discussion (5 minutes)

Purpose: Encourage critical reflection on what students experienced.

Teacher actions: Lead a brief whole-class discussion using the following questions:

- "What part of today's game could we call 'AI', and what part was just following fixed rules?"

- "How does this logic relate to the 'sense – decide – act' cycle we discussed in earlier modules?"
- "What improvements would make the enemy appear more intelligent (e.g., predicting movement, avoiding obstacles, cooperating with other enemies)?"

This discussion is oral and teacher-led, aiming to help students articulate what they learned during the coding stations.

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.
It combines:

- a short quiz,
- teacher observation,
- student self-assessment.

Quick Quiz (5 questions)

Example questions (adapt as needed):
What is the purpose of the game loop in a Pygame game?
What does the screen update command (e.g. pygame.display.flip) do?
What information does the enemy use to decide where to move?

- the player colour
- the distance to the player
- the score
- the game time

What happens when we increase ENEMY_SPEED?
If we increase react_distance, will the enemy react from farther away or closer?

Teacher Observation Checklist

Suggested items (tick per pair):

- Opens and runs the game file without help.
- Can change the player speed and explain the effect.
- Can explain in simple terms how the enemy knows where to go.
- Works collaboratively (role rotation driver / navigator).
- Identifies at least one limitation of the AI logic used.

Student Self-Assessment

Short individual form:

- Rate 1–5: "I can explain how the enemy follows the player in our game."
- Rate 1–5: "I can change a game parameter (speed, reaction distance) and predict its effect."
- Open question: "What do you think is the main limitation of the AI used in this game?"

9. Adaptations for Diverse Learners

Additional support: provide a version of the code with more comments and clearly marked sections for students who need more guidance.

For students with lower digital confidence: provide the file already open, highlight exactly which lines to edit.

Time management: If time is limited, focus on player movement and basic chasing only, leaving the reaction distance as an extension or homework.

Fast finishers:

- Add a Game Over condition when the enemy touches the player.
- Modify the logic so that the enemy runs away from the player instead of chasing.

10. Resources / Attachments

- Base Python file “Catch Me If You Can” – Module 4 game.
- Slide deck on game loop and simple game AI.
- Printable documents:
- Teacher observation checklist.
- Exit ticket / self-assessment.

All code resources for this lesson are part of the Course 2 Educational Materials (Module 4 - Python Game Programming with AI).

The complete Python game file "Catch Me If You Can" is provided separately and is available to teachers and students as part of the project's digital learning resources on the FUTURE-STEM-HUB platform: https://www.future-stem-hub.eu

Annex 1 – Teacher Observation Checklist

Use this checklist to observe student pairs during the lesson. Tick (✓) when the criterion is met.

Pair Names	Opens and runs the game file without help	Can change the player speed and explain the effect	Can explain in simple terms how the enemy knows where to go	Works collaboratively (role rotation driver / navigator)	Identifies at least one limitation of the AI logic used	Notes
Pair 1:	☐	☐	☐	☐	☐	
Pair 2:	☐	☐	☐	☐	☐	
Pair 3:	☐	☐	☐	☐	☐	
Pair 4:	☐	☐	☐	☐	☐	
Pair 5:	☐	☐	☐	☐	☐	
Pair 6:	☐	☐	☐	☐	☐	
Pair 7:	☐	☐	☐	☐	☐	
Pair 8:	☐	☐	☐	☐	☐	

General observations:

Annex 2 – Student Exit Ticket

Name (optional):
Date:

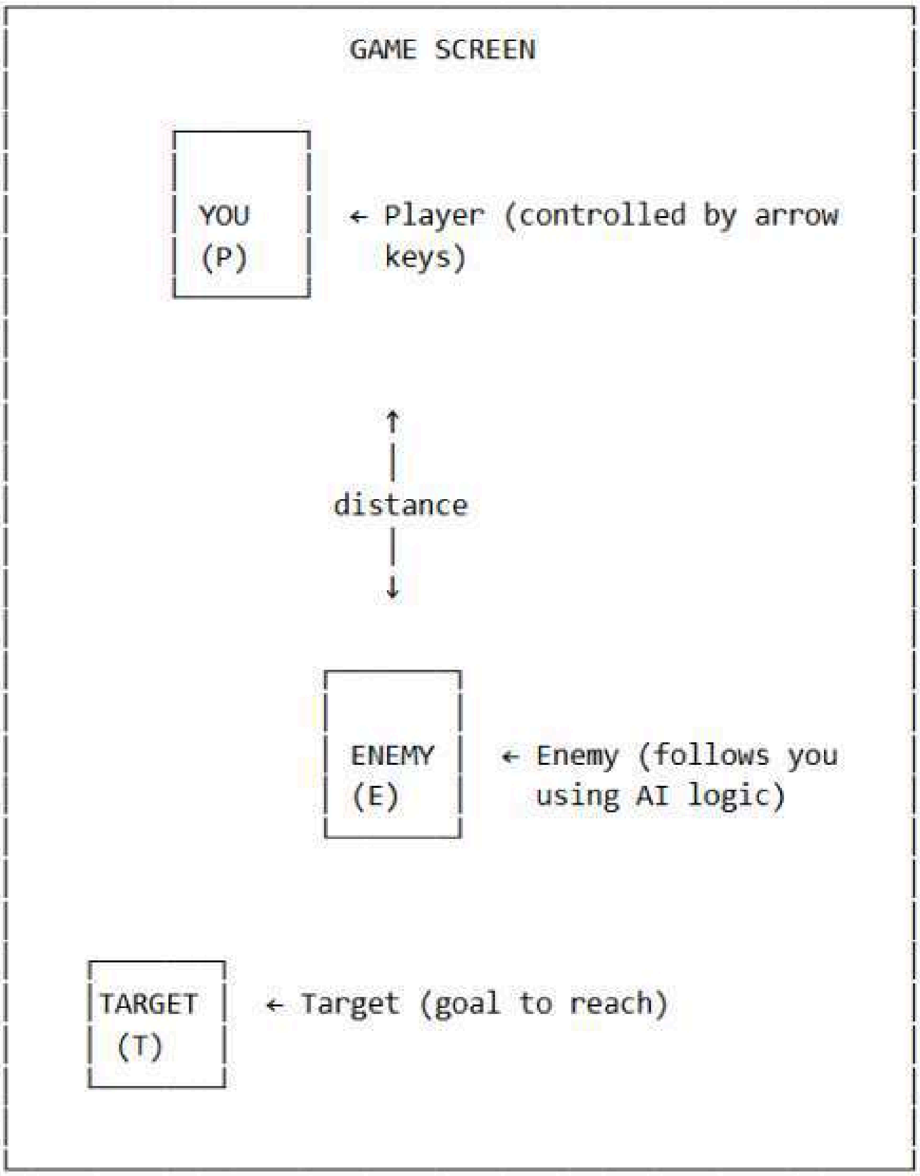
1. What is the purpose of the game loop in a Python game?
2. Describe one parameter you changed in the game today and what happened when you changed it.
3. Rate yourself from 1 to 5:
"I can explain how the enemy follows the player in our game."
1 □ — 2 □ — 3 □ — 4 □ — 5 □
"I can change the game parameter (speed, reaction distance) and predict its effect."
1 □ — 2 □ — 3 □ — 4 □ — 5 □

4. What do you think is the main limitation of the AI used in this game?
5. One thing I learned today:

Thank you!

Annex 3 – Player–Enemy–Target Support Sheet

Name (optional):
Date:



Game Entities Diagram

Key Terms

Game loop: ____

Entity: ____

Distance: ____

Rule-based AI: ____

Notes and Observations

Station 1 – Player Movement:
What happened when you changed the player speed?

Station 2 – Basic AI Logic:
How does the enemy know where to go?

Station 3 – Reaction Distance (optional):
What happened when you changed the react_distance value?

One question I still have:

Annex 4 – Code examples (screenshots) for teachers

This annex shows screenshots of the key code sections used in this lesson:

Screenshot 1 – Player speed definition (PLAYERSPEED) before changes (Station 1)

```python
PLAYERSPEED = 5  # player movement speed

# Inside the game loop:
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    player.x -= PLAYERSPEED
if keys[pygame.K_RIGHT]:
    player.x += PLAYERSPEED
if keys[pygame.K_UP]:
    player.y -= PLAYERSPEED
if keys[pygame.K_DOWN]:
    player.y += PLAYERSPEED
```

Screenshot 2 – Player speed after a sample change (Station 1)

```python
PLAYERSPEED = 8  # increased player movement speed

# Inside the game loop:
keys = pygame.key.get_pressed()
if keys[pygame.K_LEFT]:
    player.x -= PLAYERSPEED
if keys[pygame.K_RIGHT]:
    player.x += PLAYERSPEED
if keys[pygame.K_UP]:
    player.y -= PLAYERSPEED
if keys[pygame.K_DOWN]:
    player.y += PLAYERSPEED
```

Screenshot 3 – Enemy movement logic using dx, dy and dist (Station 2)

```python
dx = player.centerx - enemy.centerx
dy = player.centery - enemy.centery
dist = math.hypot(dx, dy)

if dist != 0:  # avoid division by zero
    enemy.x += int(dx / dist * ENEMYSPEED)
    enemy.y += int(dy / dist * ENEMYSPEED)
```

Screenshot 4 – Enemy movement with added reactdistance condition (Station 3)

```python
reactdistance = 150  # enemy reacts only within this distance

dx = player.centerx - enemy.centerx
dy = player.centery - enemy.centery
dist = math.hypot(dx, dy)

if dist != 0 and dist < reactdistance:
    enemy.x += int(dx / dist * ENEMYSPEED)
    enemy.y += int(dy / dist * ENEMYSPEED)
```

These images are intended to support teachers when introducing the code and checking students' work.

Lesson Plan 12:
How Do Computers "See"? – Introduction to Computer Vision & Object Detection

1. General Information

Course and Module:	Course 2: Delving Deeper in AI with Python and Scratch Module 5: Object Recognition with Roboflow – Introduction to Computer
AI Tool / Technology Used:	Roboflow, Google Colab, YOLOv11, Python
Target Age Group:	15–18 years (Secondary education)
Subject Area:	ICT, Technology, STEM
Estimated Duration:	90 minutes (2 sessions)

2. Learning Objectives (Outcomes)

By the end of this course, students will be able to:

- Definition: Explain the difference between Computer Vision and Image Processing.
- Comprehension: Simulate how computers perceive digital images as "pixels" and "binary number system (0 and 1)".
- Process Analysis: List the 5 basic stages of a computer vision project (Data acquisition, Preprocessing, Model selection, Training, Interpretation).
- Application: Train an object recognition model with a ready-made dataset (Rock-Paper-Scissors) using the Roboflow platform.
- Evaluation: Discuss the importance of data quality and accuracy in training artificial intelligence models.

3.Competences Developed

- Digital competence and AI literacy: Understanding the fundamental principles of computer vision and object detection while using AI tools such as Roboflow and Google Colab to explore how machines interpret visual information.
- Computational thinking and problem-solving: Applying logical reasoning to follow the stages of a computer vision project, analyse image data and understand how AI models are trained and evaluated.
- Data awareness and critical evaluation: Recognising the importance of high-quality datasets, accurate labelling and model evaluation, while reflecting on how data quality influences the performance of Artificial Intelligence systems.
- Collaboration and communication: Working effectively in pairs or small groups to train, test and analyse AI models, sharing observations and discussing results using appropriate technical vocabulary.
- Ethical awareness: Reflecting on the responsible use of computer vision technologies, including issues related to privacy, surveillance, bias and the ethical implications of image recognition systems.
- Creativity and experimentation: Exploring different approaches to model training by testing parameters, evaluating results and proposing improvements to increase the accuracy and reliability of AI models.

4. Prerequisites and Learning Context:

4.1 Prior knowledge required:

- Students are expected to have basic digital literacy skills, including the ability to use a computer, internet browser, and simple online platforms.
- No prior programming experience is required, as guided materials and ready-to-run code cells are provided.
- The lesson is implemented within a STEM learning context that integrates ICT, artificial intelligence, and applied problem solving through real-world scenarios.

4.2 Learning Context

- The lesson takes place in a technology-supported classroom or computer lab environment where students can engage in both conceptual understanding and practical experimentation.
- Learning is structured around inquiry-based and experiential activities, combining physical simulations (pixel drawing, role-play) with digital AI model training tasks using Roboflow and Google Colab.

5. Classroom Environment

5.1 Pre-lesson Preparation

- Computer lab or classroom with internet access
- Projection system or interactive board for demonstrations
- Flexible seating arrangement to allow pair or small-group collaboration
- Space for movement-based activities and role-play simulations
- Quiet and safe environment supporting concentration during technical tasks

5.2 Classroom Setup

- Computers arranged for individual or pair work
- Projector displaying lesson introduction and workflow steps
- Printed worksheets distributed before activities
- Board space allocated for key terminology (dataset, training, inference, accuracy)
- Teacher workstation connected to projection system for live demonstrations

5.3 Technical requirements (for in-class activities)

- **Hardware**: Computers or tablets with keyboards; a projector or interactive whiteboard for demonstrations.
- **Software**: A modern web browser (e.g., Chrome, Edge, Firefox) to access Google Colab; no local software installation is required.
- **Internet**: A stable internet connection is required throughout the lesson to run the Python code online.

6. Materials Needed

- Computers with internet access.
- Google Accounts for students (for Roboflow and Colab login).
- Projector (for teacher presentation).

- Printed Material: Graph paper for the "Pixel Painting" activity.
- Digital Resource: Roboflow "Rock-Paper-Scissors" Dataset

Link: https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw.

7. Lesson Procedure (Step-by-Step Flow)

7.1 Warm-up and Theoretical Foundations (30 Minutes)

1. Introduction and Curiosity Stimulation (5 Minutes):

- The teacher asks the question, "How does a computer recognize a cat in a photograph?"
- The similarity between the human visual system (eye-brain) and the computer vision system (camera-processor) is explained.

7.2 Main Activity: Seeing Like a Computer (10 Minutes):

- Concept: It is reminded that computers work with 1s and 0s (Binary).
- Application: Students are given graph paper.

Each square is given a number between 0 and 1 (0=Black, 1=White, 0.5=Gray). Students are asked to create a letter or shape by coloring the boxes according to the numbers inside the squares.

- Result: It is shown how these numbers turn into an "image" when viewed from a distance. It is emphasized that pixels are numerical values in the computer's memory.

Immediately following the pixel painting activity, a physical game can be added to help understand how the computer "separates" objects.

- Game: Two students are selected from the class. One becomes the "Model," and the other becomes the "Input Data."
- Process: The "Input Data" student holds a hidden object. The "Model" student is only allowed to ask yes/no questions such as "Is it angular?", "Is its color bright?", "Does it have holes?".

Outcome: This helps students understand the "feature extraction" logic of the YOLO algorithm before they even sit at a computer.

Event Text -"Hello Vision Engineers!

Today we have a very special mission. Meet Courier Kerem. Kerem is an autonomous delivery robot that carries packages around our campus. He has the latest motors in his body, but he has a problem: His eyes can see, but his brain can't understand what he sees!
Last week, Kerem tried to step over a stone on the road and fell because he thought it was a flat piece of paper. Another day, he mistook a bag flying in the wind for a rock and stopped dead in his tracks, delaying the delivery.
Kerem's sensors only see numbers and pixels. He needs a 'Vision Brain' to make sense of these pixels. Today, you will be Kerem's AI trainers. You will teach him to distinguish between rock, paper, and scissors (i.e., obstacles and commands). Kerem's ability to deliver packages safely depends entirely on the model you train. Are you ready to teach Kerem about the world?"

Student Worksheet

A. What Does Kerem See?

When Kerem looks at a photo, he doesn't say "Stone" like you do. He sees colored boxes (pixels) arranged in a grid. There is an object captured by Kerem's camera in the grid area below. Color the black areas. Do you think the resulting shape is an obstacle (Stone) or harmless (Paper) for Kerem?
....
....
....
....

B. Data Detective (Roboflow)

To prevent Kerem from making mistakes, we must show him correct examples.
Task: Examine the dataset on Roboflow. Is there an image that is mislabeled (e.g., labeled paper instead of stone)?
Observation: How many images in total are you training Kerem with? ....

C. Training and Testing (Google Colab)

Now update Kerem's brain (YOLOv11) by running the code.
Model Confidence Score: What percentage is written next to an object when your model recognizes it? (e.g., 85%)

Object 1: .... % ....
Object 2: .... % ....

Error Analysis: In what situations does Kerem have difficulty recognizing the object? (Too close, too dark, reverse angle, etc.)
....

Student Worksheet

D. Ethics Committee Decision

While walking around campus, Kerem sees not only packages but also people's faces. Is it correct for Kerem to record the face of every person he sees? Why?
....
....
....

If you only trained Kerem in sunny weather, what can he do when it rains at night?
....
....
....
....

E. EXIT TICKET

The most important thing I taught Kerem today: ....

Why is artificial intelligence's "seeing" different from humans' seeing?...
....

Tips for the Teacher

- Visualization: Bring a small box or toy car into the classroom, stick cardboard eyes on it, and introduce it as "This is Kerem."
- Dramatization: Make one student "Kerem." Blindfold them. Ask the other students to guess the object in their hand by only saying "Pixel characteristics" (Hard, soft, angular).
- Technical Note: Combine the title of YOLOv11, "You Only Look Once," with the story of Kerem having to move very quickly and instantly analyzing dozens of frames every second.

Introduction to the Roboflow Platform (10 min):

• Students go to https://roboflow.com/.
• They create an account by selecting "Sign In" and then "Continue With Google".
• It is explained that Roboflow is an interface that simplifies the process of creating computer vision models.

Dataset Selection and Preparation (15 min):

• A ready-made dataset will be used because labeling data from scratch would be time-consuming.

Step: Go to https://universe.roboflow.com/roboflow-58fyf/rock-paper-scissors-sxsw.

- **Action**: Click the "Download Dataset" button. Select YOLOv11 as the format.
- Students are instructed to select the "Show download code" option and copy the given Python code.

Model Training (20 min):

Platform: Google Colab. Open the ready-made YOLOv11 Colab notebook prepared by Roboflow.

L**ink:** train-yolo11-object-detection-on-custom-dataset.ipynb (Use the link from the lesson document).

- Students paste the copied API code into the relevant cell in Colab and start the training by pressing the "Run" button.

Note: During the training, a brief explanation is given about the speed of the YOLO (You Only Look Once) architecture and its success in object detection (looking at the entire image at once).

Testing Results (Inference) (5 min):

By uploading a sample "Rock," "Paper," or "Scissors" photo to the trained model, the model's prediction percentage (confidence score) is examined. High accuracy rates, such as 99%, are targeted.

Mini Quiz- Ethical Discussion and Homework (5 min):

The following questions are asked to measure students' understanding (oral or via Kahoot).

Discussion 1: The question "If our dataset only contained the hands of white people, could the model recognize the hand of a black person as 'Scissors'?" is discussed, along with "Data Bias" and "Data Scarcity".

....
....
....

Discussion 2: Is it more important to install cameras everywhere for security, or to uphold people's right to anonymity?

....
....
....

Next Step: Students are told they can try creating a new project by taking photos of objects in their own rooms at home.

Note for teachers

Students may be fascinated by the capabilities of Computer Vision but may also assume that these systems are always accurate. Encourage them to analyse both the strengths and limitations of image recognition technologies, discussing how factors such as lighting, image quality, training data and bias can influence AI performance. Promote critical thinking about the responsible use of Computer Vision in everyday life.

8. Reflection & Discussion with students (5–10 minutes)

Note for teachers:

The assessment methods described in this section are flexible and complementary. Teachers are encouraged to select and adapt the most suitable methods according to the needs of their students, time constraints, and learning objectives. Not all assessment tools need to be applied in every implementation of the lesson.

8.1 Practical Exercise / Student Output (10 minutes)

Expected Student Outputs

By the end of the lesson, each student or pair of students should produce:

- A completed Object Detection Observation Sheet showing the images tested and the AI predictions observed.
- A short written explanation of at least one case where the AI correctly identified an object.
- A short written explanation of at least one case where the AI made a mistake or showed uncertainty.
- A brief reflection explaining how image quality, lighting, background or training data may have influenced the results.
- Participation in the final discussion about the responsible use of Computer Vision technologies.
- The written output may be completed on paper, in a shared digital document or as part of a classroom worksheet.

8.2 Discussion (5 minutes)

- What surprised you most about the way computers recognise objects?
- Why do Computer Vision systems sometimes make mistakes even when the object seems obvious to a person?
- How can image quality, lighting or training data influence AI predictions?
- Where do you encounter Computer Vision technologies in your everyday life?
- What ethical issues should developers consider when using Computer Vision for applications such as facial recognition, healthcare or security?

8.3. Assessment

Assessment is formative and takes place during and immediately after the lesson.
It combines:
- a short quiz,
- mini Challenge: Improve the AI,
- student self-assessment.

Example questions (adapt as needed):

What is the main purpose of Computer Vision?
Answer: To enable computers to recognise, analyse and interpret visual information from images or videos.

What type of information does a Computer Vision system analyse?
1. Images and videos
2. Music files
3. Text documents
4. Internet speed

Which factor is most likely to reduce the accuracy of an object detection system?
1. Good lighting
2. High-quality images
3. Poor lighting or blurry images
4. A larger computer screen

What can influence how accurately an AI system recognises an object?
1. The quality of the training data
2. The image quality
3. The viewing angle
4. All of the above

Mini Challenge: Improve the AI

Working individually or in pairs, students analyse one example where the AI failed to recognise an object correctly and propose practical improvements to increase the system's accuracy, fairness or reliability. They present and justify their suggestions using evidence from the lesson.

Self-assessment

Students complete a brief self-assessment at the end of the lesson by reflecting on the following statements:

I can explain what Computer Vision is and how it helps computers recognise objects.
I can identify factors that influence the accuracy of object detection systems.
I understand why Computer Vision systems can make mistakes or show bias.
I can give examples of where Computer Vision is used in everyday life.
I understand why Computer Vision technologies should be used responsibly and ethically.

Students rate each statement using a simple scale:
😀 I understand this well.
😐 I understand some of it but need more practice.
☹️ I need more support to understand this concept.

9. Adaptations

- **For Advanced Students:** Instead of a ready-made dataset, students can be asked to create and label their own "Class Attendance System" datasets by taking photos of their classmates.
- **For Students Requiring Support:** Google Colab codes are provided ready-made and monitored from the teacher's screen; students simply press the "Run" buttons to observe the process and interpret the results.
- **Language Support:** Since the Roboflow and Colab interfaces are in English, the Turkish equivalents of basic terms (Train, Dataset, Inference, Accuracy) are written on the board.

10.Resources / Attachments

- **Time Management:** Model training may vary depending on computer speed/Colab server. If time is insufficient, reduce the number of epochs or proceed directly to the testing phase using pretrained weights.
- **AI Ethics:** Don't focus solely on technical achievement in the lesson; briefly address the ethical responsibilities related to the use of these systems in surveillance or military applications.
- **Link**:https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolo11-object-detection-on-custom-dataset.ipynb

Digital Resources

- Course 2 Master Document (Lesson 12).
- Computer Vision demonstration tool (e.g. Google Teachable Machine, Google Colab notebook or another browser-based object detection platform).
- Teacher presentation on Computer Vision and Object Detection.
- Interactive whiteboard or projector.
- Optional introductory video explaining how computers analyse and recognise images.

Physical / Printable Materials

- Object Detection Observation Sheet.
- Image Analysis Worksheet.
- Prediction vs Reality Recording Table.
- Reflection Questions.
- Exit Ticket.
- Computer Vision Process Diagram (optional).

Summary and Next Steps

The eToolkit has been developed as a practical resource to support secondary school teachers in integrating Artificial Intelligence into their teaching. It combines a clear methodological approach with a set of 12 structured lesson plans that provide step-by-step guidance for classroom implementation. The content covers both introductory and more advanced AI topics, allowing teachers to progressively build their confidence and skills. Overall, the eToolkit is designed to be flexible and easy to use, helping educators translate AI concepts into meaningful learning activities for their students.

Project partners

FUTURE STEM HUB

UNIVERSITÄT DUISBURG ESSEN
Open-Minded

KÜTAHYA
İL MİLLÎ EĞİTİM MÜDÜRLÜĞÜ

tetra solutions

M&M profuture training

ETAP ESCOLA PROFISSIONAL

Project Coordinator:
University of Duisburg-Essen, Germany

Email
mustafa.bilgin@uni-due.de

Website
www.future-stem-hub.eu

Co-funded by the European Union

Funded by the European Union. The opinions and views expressed are, however, the sole responsibility of the author(s) and do not necessarily reflect those of the European Union or the European Education and Culture Executive Agency (EACEA). Neither the European Union nor the EACEA accepts responsibility for them.