LESSON 79

AI and Machine Learning Basics

80 minutes (2 periods) | Ethics, Society, and Emerging Technologies | Outcome 1.14 | Outcome 1.13 | Outcome 2.7 | Outcome 2.9 | Outcome 3.7 | Outcome 3.6 | Outcome 1.6

WHAT THIS LESSON DOES

In this lesson, you'll uncover the essentials of Artificial Intelligence and Machine Learning. Follow step-by-step guidance to define AI, explore Machine Learning categories, set up a Python environment, and build models like linear regression for practical predictions.

CURRICULUM OUTCOMES

Outcome	What the specification asks for
1.14	explain when and what machine learning and AI algorithms might be used in certain contexts
1.13	identify important computing developments that have taken place in the last 100 years and consider emerging trends that could shape future computing technologies
2.7	implement algorithms using a programming language to solve a range of problems
2.9	assemble existing algorithms or create new ones that use functions (including recursive), procedures, and modules
3.7	use algorithms to analyse and interpret data in a way that informs decision-making
3.6	represent data to effectively communicate in a graphical form
1.6	explain the operation of a variety of algorithms

WHAT STUDENTS SHOULD BE ABLE TO DO

- Understand the fundamental concepts of Artificial Intelligence (AI) and its various types.
- Grasp the principles of Machine Learning (ML) as a subset of AI, including its main categories and learning mechanisms.
- Develop practical skills in setting up a Python environment for ML projects using relevant libraries.
- Apply supervised learning techniques to build and interpret models like linear regression and decision tree classifiers.
- Enhance ML models by incorporating additional features and visualisations for deeper insights.

BEFORE THE LESSON

- Confirm VS Code and a working Python version are installed on every machine, and that the Python extension is present so 'Python: Create Environment' appears in the Command Palette.
- Test pip install scikit-learn, numpy and matplotlib from a terminal behind the school filter before class. If installs are blocked, arrange a wheels cache or offline install in advance.
- Have the full code for ml_demo.py and character_strength.py saved somewhere the class can copy from, so time goes on reasoning rather than transcription errors.

AI and Machine Learning Basics

HOW THE LESSON RUNS

- Introduction
- What is Artificial Intelligence (AI)?
- What is Machine Learning (ML)?
- Set Up Your Project
- Machine Learning - Linear Regression
- Extension
- Machine Learning - Classifier
- Extension
- Wrap Up

TEACHING MOVES

- 2. What is Artificial Intelligence (AI)? Anchor narrow versus general AI in examples students already use: a phone voice assistant and a streaming recommender are narrow. Stress general AI does not exist yet, to correct science-fiction expectations before ML.
- 3. What is Machine Learning (ML)? Draw the line clearly: ML learns patterns from data rather than following hand-written rules. Name all three types, but flag that both today's models are supervised, so students expect labelled input and output pairs.
- 4. Set Up Your Project. Do this live on the projector. The virtual environment and the pip install are where lessons die. Watch that students select the .venv interpreter, not the system Python, before running any code.
- 5. Machine Learning - Linear Regression. Before running, ask students to predict roughly what price a larger house should get. Then show the line of best fit is just the model minimising error, not magic. Connect coefficient to slope.
- 6. Extension. For fast finishers only. Point out that adding bedrooms means the model now fits a plane, not a line. Ask why more features might, or might not, improve a prediction.
- 7. Machine Learning - Classifier. Contrast this with the regression model: here the output is a category, not a number. Explain train_test_split holds back data so the accuracy score tests unseen examples, not memorised ones.
- 8. Extension. Have students read the visualised tree aloud as a set of if-then questions on the features. This makes the decision tree interpretable and connects it back to rule-based thinking.
- 9. Wrap Up. Ask students to state, in one sentence each, the difference between AI and ML and between the two models built today. Use their answers to catch anyone still conflating them.

WHAT TO WATCH FOR

- Students think a high accuracy score means the model is genuinely good, without noticing the dataset is tiny and simulated. Point at the sample size in the code. Ask what would happen with only a handful of test rows, and note that real ML needs far more, more varied data.
- Students believe ML is programmed with explicit rules like any other code they have written. Contrast the regression code with an if-else version they could imagine writing by hand. The model derives the relationship from data; nobody typed the slope.
- Students run code against the system Python and hit missing-library errors despite having created a virtual environment. Show them the interpreter selector in the VS Code status bar. The .venv must be selected, and packages must be installed inside that terminal, for scikit-learn to import.

DIFFERENTIATION

- Emerging: If the environment is fighting a student, get them running the code in the working demo interpreter first so they see a result, then return to fixing their own .venv. Pair them with someone whose install succeeded and have them narrate what each line does rather than typing from scratch.
- Developing: Ask them to change one input value and predict the output before running, checking their reasoning against the result. Have them explain aloud why the classifier uses train_test_split before you accept they have finished.
- Proficient: Push into the extension tasks, then ask them to try a feature that should not help and observe whether accuracy changes. Ask them to sketch how a labelled dataset of their own choosing, evidence they could reuse in an Applied Learning Task, could train a similar supervised model.