



Parcours

Data Manager for agro-environmental projects

UE 3:

Artificial Intelligence Tools and Precision Agriculture Project

March 31 – April 25, 2025

- 3.1 Artificial Intelligence Tools (March 31 – April 11)
- 3.2 Precision Agriculture Project (April 14 – April 25)
 - Project applying techniques from UE 1 and UE 2
 - Groups of 4 students: 2 from the program
 - + 2 students from the Agrotic course (Master 2) who are on an apprenticeship scheme.

- Discover some artificial intelligence tools
 - Understand the benefits of these tools
 - Explore their implementation complexity
 - Identify limitations and risks of their use

By the end of this module, you won't be specialists in Artificial Intelligence!

Assessment: group work (4 students per group)



What is Artificial Intelligence?

AI is not limited to *Generative AI* !

Generative AI / LLM (Large Language Model)

Neural Networks

Machine Learning

Artificial Intelligence



What is Artificial Intelligence?

UE 3

IJCAI'24 Conference topics:

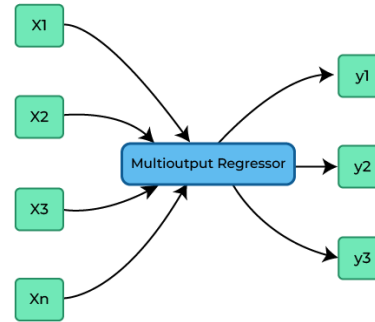
- AI Ethics, Trust, Fairness
- Agent-based and Multi-agent Systems
- Computer Vision
- **Constraint Satisfaction and Optimization**
- Data Mining
- Game Theory and Economic Paradigms
- Humans and AI
- **Knowledge Representation and Reasoning**
- **Machine Learning**
- Multidisciplinary Topics and Applications
- Natural Language Processing
- Planning and Scheduling
- Robotics
- Search
- Uncertainty in AI

- Semantic networks / Ontologies:
representing / organising / reasoning about knowledge
represented with graphs
- Reasoning Under Constraints:
an application to the field of agroecology
- Machine Learning

Supervised Learning: « learning from (labeled) examples »

- Regression

input:
continuous
multidimensional
values



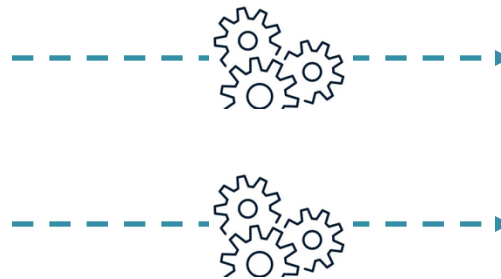
output:
continuous
multidimensional
values

- Classification

input:
continuous
or discrete
multidimensional
values



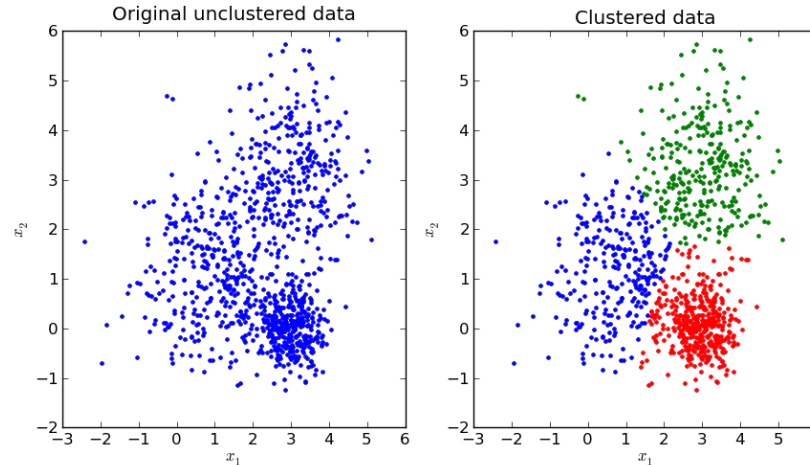
Model to learn



output:
discrete
values
(classes)

Unsupervised Learning: « learning from observation »

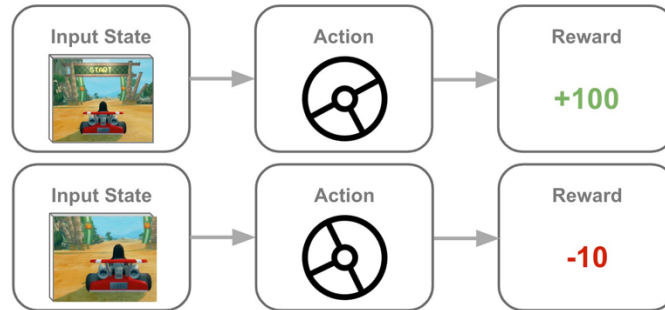
- Clustering



- Dimensionality Reduction (PCA, autoencoders, ...)
- Association Rules (Data Mining, ...)

Reinforcement Learning: « learning from interactions »

- Maximizing a reward function on actions modifying the state of the system



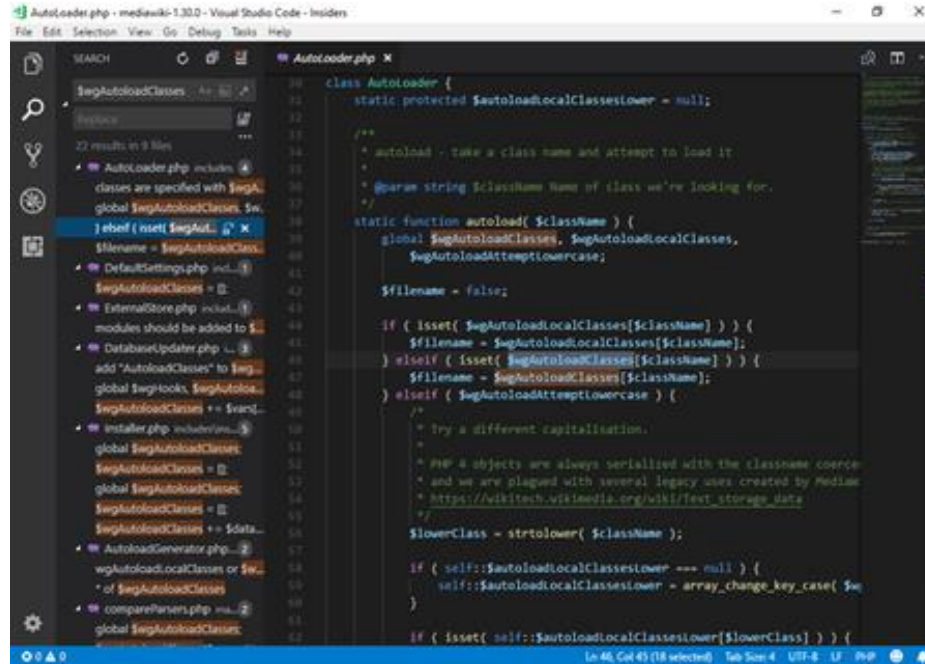
- Complex problems:
combining multiple learning types (LLMs, Go game, ...)

Main problem: classifying hyperspectral datasets
(detect the state of a plant, ...)

Problem-solving based learning approach

- Exploring new tools through minimalist tutorials
=> you can use new functions as long as you can explain them
- Student collaboration and peer learning
- Possibility to use Generative AI (e.g., Github Copilot)

- Programming environment: VSCode



The screenshot shows the Visual Studio Code editor interface. The title bar indicates the file is 'Autoloader.php' in a 'mediawiki-1.30.0' workspace. The menu bar includes File, Edit, Selection, View, Go, Debug, Tools, and Help. The left sidebar shows a search bar and a file explorer with 22 results in 9 files. The main editor area displays the code for the 'Autoloader' class. The code includes a static protected property '\$autoloadLocalClassesLower', a docblock for the 'autoload' function, and the function implementation which checks for class existence in local and global arrays and attempts to load the file. The status bar at the bottom shows 'Ln 46, Col 43 (18 selected)', 'Tab Size: 4', 'UTF-8', and 'PHP'.

```
Autoloader.php - mediawiki-1.30.0 - Visual Studio Code - Insiders
File Edit Selection View Go Debug Tools Help

SEARCH
$wgAutoloadClasses: 4x
Autoloader.php includes
classes are specified with $wgAutoloadClasses, $w
global $wgAutoloadClasses, $w
} elseif ( isset( $wgAutoloadClasses ) ) {
$filename = $wgAutoloadClasses[ $className ];
DefaultSettings.php includes
$wgAutoloadClasses = [];
ExternalStore.php includes
modules should be added to $wgAutoloadClasses
DatabaseUpdater.php includes
add "AutoloadClasses" to $wgAutoloadClasses
global $wgAutoloadClasses, $wgAutoloadClasses
$wgAutoloadClasses += $event
Installer.php includes
global $wgAutoloadClasses
$wgAutoloadClasses = [];
global $wgAutoloadClasses
$wgAutoloadClasses = [];
$wgAutoloadClasses += $data
AutoloadGenerator.php includes
wgAutoloadLocalClasses or $wgAutoloadClasses
of $wgAutoloadClasses
comparePages.php includes
global $wgAutoloadClasses

class Autoloader {
    static protected $autoloadLocalClassesLower = null;

    /**
     * autoload - Take a class name and attempt to load it
     *
     * @param string $className Name of class we're looking for.
     */
    static function autoload( $className ) {
        global $wgAutoloadClasses, $wgAutoloadLocalClasses,
            $wgAutoloadAttemptLowercase;

        $filename = false;

        if ( isset( $wgAutoloadLocalClasses[ $className ] ) ) {
            $filename = $wgAutoloadLocalClasses[ $className ];
        } elseif ( isset( $wgAutoloadClasses[ $className ] ) ) {
            $filename = $wgAutoloadClasses[ $className ];
        } elseif ( $wgAutoloadAttemptLowercase ) {
            /* Try a different capitalisation.
             *
             * PHP 4 objects were always serialized with the classname coerced
             * and we are plagued with several legacy users created by MediaWiki
             * https://wikitech.wikimedia.org/wiki/Text_storage_date
             */
            $lowerClass = strtolower( $className );

            if ( self::$autoloadLocalClassesLower === null ) {
                self::$autoloadLocalClassesLower = array_change_key_case( $wgAutoloadLocalClasses, CASE_LOWER );
            }

            if ( isset( self::$autoloadLocalClassesLower[ $lowerClass ] ) ) {
                $filename = self::$autoloadLocalClassesLower[ $lowerClass ];
            }
        }

        if ( $filename ) {
            require_once $filename;
        }
    }
}
```

- Programming Language: Python

C++

```
#include <iostream>
#include <cmath>

// For the pow function
double power(double x, double y) {
    return std::pow(x, y);
}

int main() {
    double result = power(2, 3);
    std::cout << result << std::endl;
    return 0;
}
```

Python

```
def power(x, y):
    return x ** y

# Example usage
result = power(2, 3) print(result) # Displays
8
```

➤ One 'beginner' in Python per student group

Let's start...

UE 3

