UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

Python Programming for Data Science and Engineering

Ph.D. School in Information Engineering
A.Y. 2024/2025

Stefano Tortora
stefano.tortora@unipd.it
Intelligent Autonomous Systems Laboratory (IAS-Lab)
Dept. of Information Engineering (DEI)
University of Padua, Italy

Assignment 5

The aim of **Assignment 5** is to complete your game by adding a Machine learning-based *Pokemon Recommender System* that helps the player during the game. Thus, to complete the assignment you must:

1. Collect a big-enough dataset for model training
2. Implement, fit and validate an appropriate ML model "offline"
3. Include the trained model within your game
4. (optional) Improve and Test your game with a few users

Explanations on how to perform these steps are provided in the following slides.

Dataset collection

As first step, you need to collect additional battles' data that will form the dataset for training and validating our machine learning model.

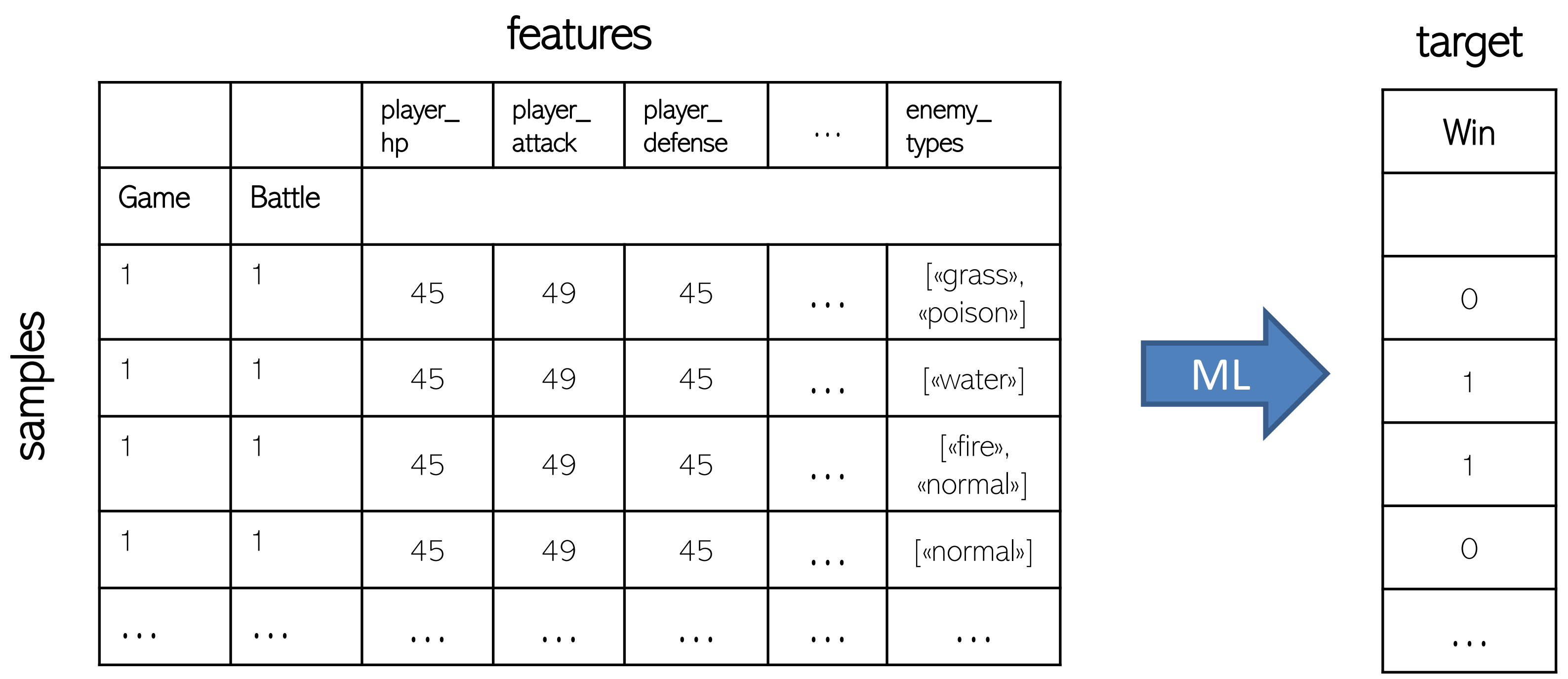
To do so, lets exploit again the *New Random Battle Mode* of assignment 4 with the following modifications:

1. Allow the player to randomly select the starter pokemon among ALL the possible pokemons in the *pokemons.json* file (and not only Bulbasaur, Charmender, etc.).
2. Run the game several times (e.g., *Ngames* = 1000 or more) and with a high-enough number of battles (e.g., *Nbattles* = 500 or more).
3. For each battle of each game, save the following statistics:
   - ❑ The *ActStats* and *types* of the player's Pokemon;
   - ❑ The *ActStats* and *types* of the opponent Pokemon;
   - ❑ If the battle was a win (1) or a lose (0).

Model implementation

Implement and fit an appropriate machine learning (ML) model that, given the *ActStats* and *types* of the player's pokemon and of the enemy pokemon as ***features*** and the battles as ***samples***, predicts the probability of winning.

Use the same dataset also to validate the model and plot some results.

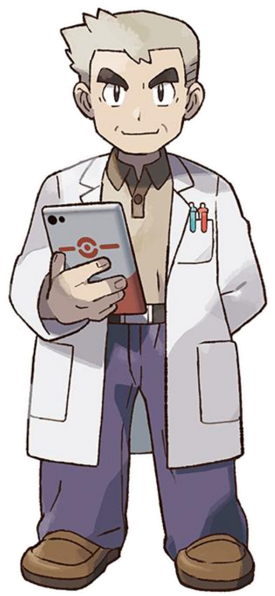


		player_ hp	player_ attack	player_ defense	…	enemy_ types
Game	Battle					
1	1	45	49	45	…	[«grass», «poison»]
1	1	45	49	45	…	[«water»]
1	1	45	49	45	…	[«fire», «normal»]
1	1	45	49	45	…	[«normal»]
…	…	…	…	…	…	…

Win
0
1
1
0
…

Pokemon Recommender System

The *Pokemon Recommender System* should assist the player in selecting, among the pokemons available within the player's Pokemon list, the one that has the highest chance of winning against the opponent Pokemon.



To do so, <u>at the beginning of each battle</u> against an opponent pokemon, your recommender system must:

1. Use the ML model to predict the probability of winning of each Pokemon in the Trainer's pokemon list;
2. If none of the available pokemon has a probability of winning above chance level (i.e., 50%), suggest the player to run away;
3. Otherwise, suggest to the player the pokemon with the highest probability of winning to be played as active pokemon for the battle.

(optional) Improve ant Test your game

To make your game more enjoyable, you can optionally add other features. Here, there are some suggested improvements that you can make:

- Pokemon Level Up: increase by 1 the level of the player's active pokemon everytime it wins a battle against a wild pokemon in the "Explore" action;
- Save Player's Progress: allow the player to save and load the data of his/her Trainer and Pokemons so that the player experience will not be lost;
- Battles against other Trainers: allow the player to battle against other Trainers (e.g., randomly encountered during the "Explore" action)

Finally, test your game with more than one user (5 to 10) playing your game both with and without the *Pokemon Recommender System*.

Collect opinions using a Likert Scale (from 1 to 7) from people about:

- Enemy Difficulty level
- Battle Engagement Level
- Fun Level

Plot some results on the collected opinions comparing the users' preferences between playing with and without the *Pokemon Recommender System*.

About the final exam

WHEN: After handing over all the five assignments on the Moodle course, send me an e-mail suggesting some dates to give your presentation (NO LATE than 31th October).

WHAT: A conference-like presentation with a time slot of 10 minutes + 5 minutes questions.

HOW: The presentation should cover the following topics:

- How have you done your assignments? Main design choices and results.
- How can Python help you in your research?

The presentation should contain at least the following information:

- your name
- your supervisor
- your PhD course cycle
- max 1-2 slides for each assignment
- a description of your research in 10 lines (more or less) and 1 image
- an explanation on how Python can help you in your research