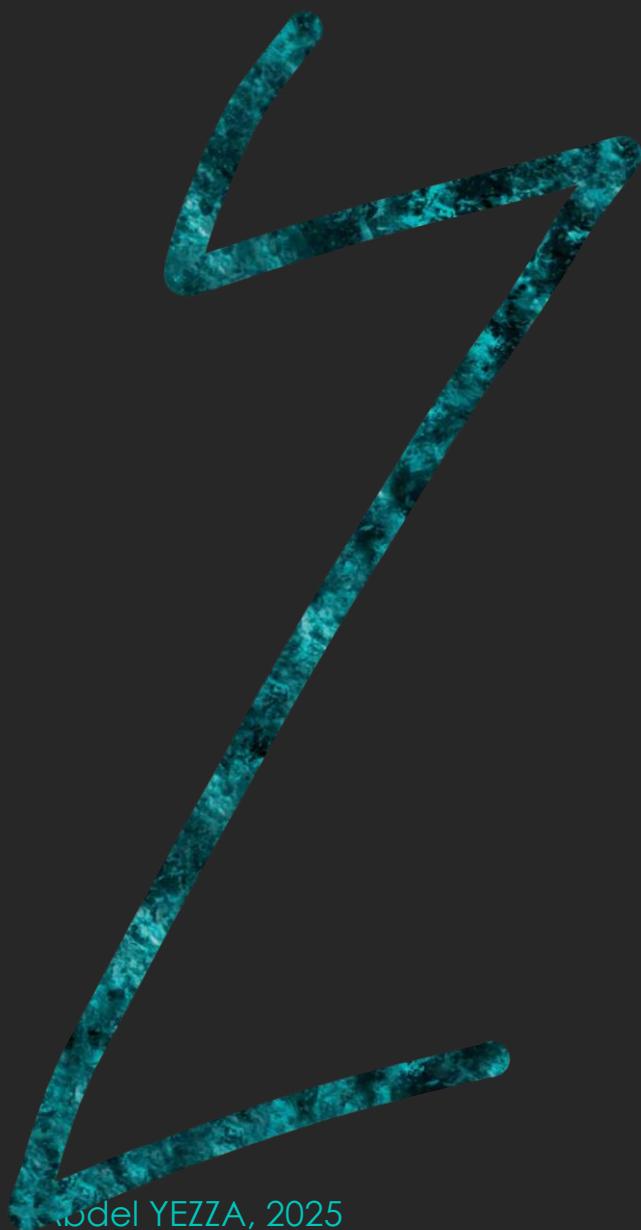


22/02/2025

Profiles assignment - GITHUB project - By Abdel YEZZA (Ph. D) – Feb. 2025



Content

Figures	1
MCAP - Profile-Activity Matching System 	2
Core Dependencies.....	2
🎯 Key Features	2
🛠 Installation	3
📦 Dependencies	3
🚀 Usage	3
📡 Command-Line Interface	3
📋 Command-Line Arguments (Console case)	4
📋 Input File Formats	4
MCA (Competency-Activity Matrix).....	4
MCP (Competency-Profile Matrix)	4
Run the previous example.....	4
📋 Web application	6
1. Backend.....	6
2. Frontend.....	6
📋 Streamlit demo application	10
📁 Project Structure	12
📋 License	12
🤝 Contributing.....	12
VIDEOS	13
🎬 WEB DEMO.....	13
🎬 STREAMLIT DEMO	13
🎬 CONSOLE DEMO	13

Figures

Fig.1. Web output_1	6
Fig.2. Web output_2	7
Fig.3. Web output_3	8
Fig.4. Web output_4 – extract of some radar graphs	9
Fig.5. Streamlit input data and options.....	10
Fig.6. Streamlit output – interactive case	11

MCAP - Profile-Activity Matching System

License MIT python 3.8+

Core Dependencies

Python 3.8+ NumPy 1.26.4+ Pandas 2.2.1+ Scikit-learn 1.3.2+ FastAPI 0.115.8+ Matplotlib 3.8.2+
React 18.0+ Node.js 18.0+ NPM 9.0+

A Python package designed for optimal profile-to-activity assignment based on skills matching, developed by Abdel YEZZA (Ph.D). This solution employs advanced algorithms to maximize the correspondence between required activity competencies and available profile skills.

It is designed to provide a comprehensive solution for profile-activity matching, offering multiple models, flexible scaling options, complete web application, streamlit web interface, a robust logging system, a detailed validation of input data, and customizable processing of MCAP functions (sum, mean, Euclidean and any custom function).

This project is built on the two following articles presenting all the concepts and basic elements:

1. [UN NOUVEAU MODELE POUR AFFECTER LES PROFILS ADEQUATS](#) - by Abdel YEZZA (Ph.D) - 2024
2. [UNE NOUVELLE FAÇON D'AFFECTATION DES PROFILS AUX ACTIVITES](#) - by Abdel YEZZA (Ph.D) - 2022

Key Features

- **Skills Matrix Analysis:** Process and analyze competency-activity (MCA) and competency-profile (MCP) matrices
- **Multiple Model Support:** Five different matching models available (model1 through model5 or any custom function)
- **Flexible Scaling:** Support for different scale types (0-1, free)
- **Web Interface:** Built-in web application using FastAPI and Streamlit
- **Detailed Logging:** Comprehensive logging system for tracking operations
- **Data Validation:** Robust input validation and error handling
- **Customizable Processing:** Support for different MCAP functions (sum, mean, euclidean and any custom function)

🛠 Installation

1. Clone the repository:

```
git clone [repository-url]  
cd profiles_assignment
```

2. Create and activate a virtual environment (recommended):

```
# install venv on Linux/MacOS:  
python -m venv venv  
source venv/bin/activate  
# to activate on Windows:  
venv\Scripts\activate  
Install dependencies:  
pip install -r requirements.txt
```

📦 Dependencies

- streamlit >= 1.24.0
- pandas >= 1.5.0
- scikit-learn
- matplotlib
- fastapi >= 0.104.0
- uvicorn >= 0.24.0
- python-dotenv >= 1.0.0
- python-multipart >= 0.0.6
- sqlalchemy >= 2.0.23

🚀 Usage

🏁 Command-Line Interface

Basic usage:

```
python main.py
```

Advanced usage with custom parameters:

```
python main.py --mca path/to/mca.csv --mcp path/to/mcp.csv --model model_name --scale  
scale_type --mcap mcap_function
```

Example:

```
python main.py --mca .\data\input\mca.csv --mcp .\data\input\mcp.csv --model model5 --  
scale 0-1 --mcap sqrt
```

📋 Command-Line Arguments (Console case)

- --mca: Path to the MCA (Matrix Competency-Activity) file
- --mcp: Path to the MCP (Matrix Competency-Profile) file
- --model: Model selection (model1, model2, model3, model4, model5 or your own model function)
- --scale: Scale type (0-1 or free)
- --mcap: MCAP function type (sum, mean, sqrt or a custom MCAP function)

📋 Input File Formats

MCA (Competency-Activity Matrix)

```
Activity,Comp1,Comp2,Comp3
Activity1,0.8,0.6,0.7
Activity2,0.5,0.9,0.4
```

MCP (Competency-Profile Matrix)

```
Profile,Comp1,Comp2,Comp3
Profile1,0.7,0.8,0.6
Profile2,0.9,0.5,0.8
```

Run the previous example

```
python main.py --mca .\data\input\mca.csv --mcp .\data\input\mcp.csv --model model5 --
scale 0-1 --mcap sqrt
```

should have an **output** log like the following:

```
2025-02-22 11:51:32 - myLogger - INFO - Starting command-line processing
2025-02-22 11:51:32 - myLogger - INFO - Processing with parameters:
2025-02-22 11:51:32 - myLogger - INFO - - Model: model5
2025-02-22 11:51:32 - myLogger - INFO - - Scale: 0-1
2025-02-22 11:51:32 - myLogger - INFO - - MCAP: sqrt

...
2025-02-22 11:51:32 - myLogger - INFO - Initialized processor with parameters:
2025-02-22 11:51:32 - myLogger - INFO - - mca_matrix shape: (10, 10)
2025-02-22 11:51:32 - myLogger - INFO - - mcp_matrix:          Comp1  Comp2  Comp3  Comp4  Comp5
Comp6  Comp7  Comp8  Comp9  Comp10
Profile
Prof1    0.9    0.5    0.3    0.8    0.6    0.7    0.4    0.8    0.5    0.6
Prof2    0.4    0.8    0.6    0.5    0.7    0.9    0.3    0.7    0.4    0.8
Prof3    0.6    0.4    0.9    0.7    0.5    0.8    0.6    0.4    0.9    0.3
Prof4    0.8    0.7    0.4    0.9    0.3    0.6    0.8    0.5    0.7    0.4
Prof5    0.5    0.9    0.7    0.4    0.8    0.3    0.7    0.6    0.8    0.5
Prof6    0.7    0.3    0.8    0.6    0.4    0.9    0.5    0.7    0.3    0.8
Prof7    0.9    0.6    0.5    0.8    0.7    0.4    0.9    0.3    0.6    0.7
Prof8    0.3    0.8    0.7    0.5    0.9    0.6    0.4    0.8    0.5    0.9
Prof9    0.8    0.4    0.9    0.3    0.6    0.7    0.5    0.9    0.4    0.6
Prof10   0.6    0.9    0.4    0.7    0.5    0.8    0.3    0.6    0.8    0.5
2025-02-22 11:51:32 - myLogger - INFO - - mcp_matrix shape: (15, 10)
```

```

2025-02-22 11:51:32 - myLogger - INFO - - mcp_matrix:
Comp6 Comp7 Comp8 Comp9 Comp10
Profile
Prof1 0.9 0.5 0.3 0.8 0.6 0.7 0.4 0.8 0.5 0.6
Prof2 0.4 0.8 0.6 0.5 0.7 0.9 0.3 0.7 0.4 0.8
Prof3 0.6 0.4 0.9 0.7 0.5 0.8 0.6 0.4 0.9 0.3
Prof4 0.8 0.7 0.4 0.9 0.3 0.6 0.8 0.5 0.7 0.4
Prof5 0.5 0.9 0.7 0.4 0.8 0.3 0.7 0.6 0.8 0.5
Prof6 0.7 0.3 0.8 0.6 0.4 0.9 0.5 0.7 0.3 0.8
Prof7 0.9 0.6 0.5 0.8 0.7 0.4 0.9 0.3 0.6 0.7
Prof8 0.3 0.8 0.7 0.5 0.9 0.6 0.4 0.8 0.5 0.9
Prof9 0.8 0.4 0.9 0.3 0.6 0.7 0.5 0.9 0.4 0.6
Prof10 0.6 0.9 0.4 0.7 0.5 0.8 0.3 0.6 0.8 0.5
...
2025-02-22 11:51:32 - myLogger - INFO - MCAP matrix sample:
Profile Prof1 Prof2 Prof3 Prof4 Prof5 Prof6 Prof7 Prof8 Prof9 Prof10
Profile Prof11 Prof12 Prof13 Prof14 Prof15
Activity
Act1 1.844135 1.804701 1.851501 1.836739 1.767138 1.784657 1.865625 1.861451 1.797143 1.863017
1.871348 1.790639 1.831438 1.908897 1.850976
Act2 1.752379 1.800540 1.668249 1.757919 1.821477 1.703428 1.911079 1.934339 1.726670 1.745709
1.823077 1.809159 1.858987 1.711400 1.805393
Act3 1.688935 1.663413 1.819722 1.775372 1.822392 1.721434 1.921082 1.820256 1.780839 1.668083
1.849399 1.851501 1.750000 1.802005 1.835151
Act4 1.843005 1.838780 1.799228 1.851426 1.779123 1.815902 1.870754 1.915942 1.802467 1.852476
1.867113 1.789475 1.923538 1.842477 1.840365
Act5 1.781853 1.754360 1.875426 1.790562 1.849850 1.790019 1.965042 1.888341 1.844060 1.758787
1.922527 1.878534 1.781853 1.923033 1.911878
Act6 1.820256 1.897659 1.782476 1.777014 1.772083 1.853600 1.854649 1.984313 1.821324 1.829845
1.897659 1.772631 1.935200 1.832500 1.874463
Act7 1.698283 1.697138 1.828402 1.754597 1.843758 1.733974 1.892676 1.838478 1.837949 1.684488
1.864210 1.878755 1.735655 1.797838 1.846919
Act8 1.809926 1.893923 1.748730 1.776310 1.777951 1.814754 1.834999 1.974842 1.801311 1.822773
1.875352 1.781775 1.922455 1.776857 1.848723
Act9 1.709532 1.704976 1.810080 1.725382 1.831969 1.711237 1.902776 1.867708 1.790639 1.688935
1.830376 1.860929 1.729884 1.811690 1.815978
Act10 1.776232 1.771848 1.809850 1.861749 1.806085 1.697302 1.880972 1.854948 1.696729 1.850225
1.833333 1.764149 1.831211 1.817889 1.815749
2025-02-22 11:51:32 - myLogger - INFO - Ranking matrix generated successfully
2025-02-22 11:51:32 - myLogger - INFO - MCAP matrix shape: (10, 3)
2025-02-22 11:51:32 - myLogger - INFO - MCAP matrix sample:
Top1 Top2 Top3
Activity
Act1 Prof14 (1.909) Prof11 (1.871) Prof7 (1.866)
Act2 Prof8 (1.934) Prof7 (1.911) Prof13 (1.859)
Act3 Prof7 (1.921) Prof12 (1.852) Prof11 (1.849)
Act4 Prof13 (1.924) Prof8 (1.916) Prof7 (1.871)
Act5 Prof7 (1.965) Prof14 (1.923) Prof11 (1.923)
Act6 Prof8 (1.984) Prof13 (1.935) Prof2 (1.898)
Act7 Prof7 (1.893) Prof12 (1.879) Prof11 (1.864)
Act8 Prof8 (1.975) Prof13 (1.922) Prof2 (1.894)
Act9 Prof7 (1.903) Prof8 (1.868) Prof12 (1.861)
Act10 Prof7 (1.881) Prof4 (1.862) Prof8 (1.855)
...
2025-02-22 11:51:32 - myLogger - INFO - Ranking matrix saved to: data\output\ranking_matrix.csv
2025-02-22 11:51:32 - myLogger - INFO - Ranking matrix saved to: data\output\ranking_matrix.csv
2025-02-22 11:51:33 - myLogger - INFO - Matrice MCAP sauvegard e dans: data\output\mcap_matrix.txt
2025-02-22 11:51:33 - myLogger - INFO - Matrice MCAP sauvegard e dans: data\output\mcap_matrix.txt
2025-02-22 11:51:33 - myLogger - INFO - MCAP matrix saved to: data\output\mcap_matrix.csv
2025-02-22 11:51:41 - myLogger - INFO - Traitement termin  avec succ s

```

Web application

1. Backend

```
# go to backend folder  
cd web/backend  
  
# Install dépendances if any  
pip install -r requirements.txt  
  
# Run the serveur with unicorn  
unicorn main:app --reload --log-level debug
```

2. Frontend

```
# go to frontend folder  
cd web/frontend  
  
# Install dépendances if any  
npm install  
  
# Run the dev server  
npm start
```

You should get a message like:

```
Local: http://localhost:3001  
On Your Network: http://192.168.1.19:3001
```

You can now view mcap-frontend in the browser.

The web main interface looks like this:

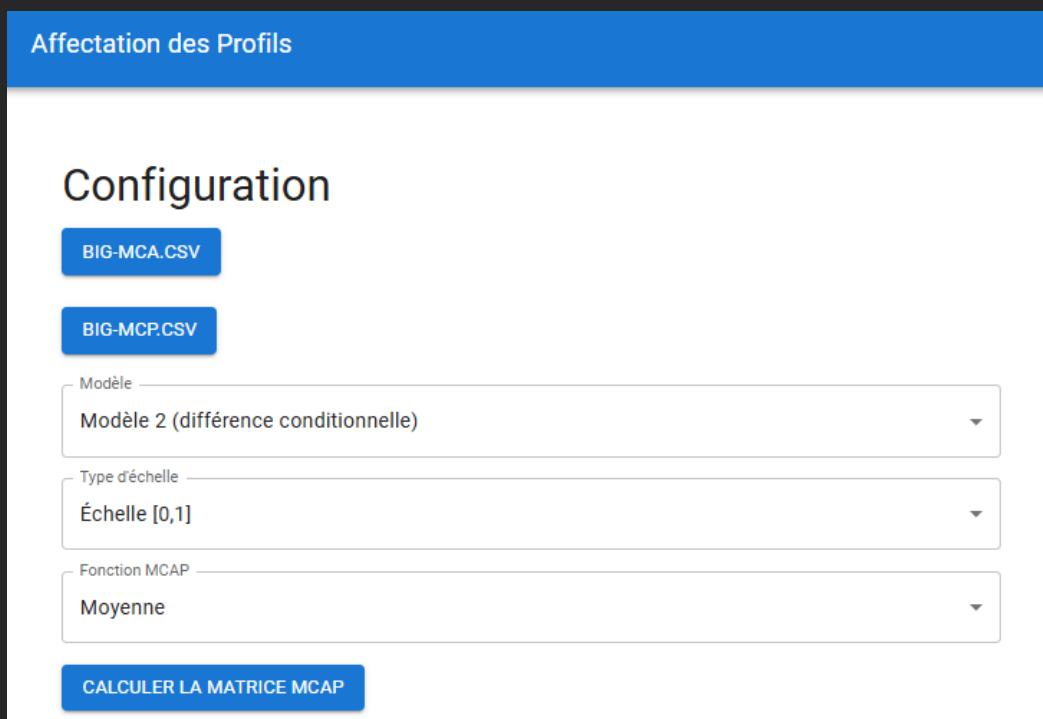


Fig.1. Web output_1

Depending on made options and input CSV matrices (mca and mcp), results look like this:

Classement des profils par activité

Activity	Top 1	Top 2	Top 3
Activ.1	Prof.4 (0.358)	Prof.1 (0.300)	Prof.5 (0.283)
Activ.2	Prof.4 (0.233)	Prof.1 (0.200)	Prof.5 (0.142)
Activ.3	Prof.4 (0.242)	Prof.5 (0.208)	Prof.1 (0.200)
Activ.4	Prof.4 (0.350)	Prof.1 (0.325)	Prof.5 (0.250)
Activ.5	Prof.5 (0.267)	Prof.4 (0.258)	Prof.1 (0.242)
Activ.6	Prof.1 (0.250)	Prof.5 (0.167)	Prof.4 (0.158)
Activ.7	Prof.4 (0.267)	Prof.1 (0.242)	Prof.5 (0.192)
Activ.8	Prof.4 (0.375)	Prof.5 (0.192)	Prof.1 (0.175)
Activ.9	Prof.4 (0.417)	Prof.1 (0.342)	Prof.5 (0.333)
Activ.10	Prof.4 (0.300)	Prof.1 (0.183)	Prof.5 (0.175)

Matrice des résultats détaillée

Activity	Prof.1	Prof.2	Prof.3	Prof.4	Prof.5
Activ.1	0.300	0.067	0.042	0.358	0.283
Activ.2	0.200	0.083	-0.125	0.233	0.142
Activ.3	0.200	0.067	-0.025	0.242	0.208
Activ.4	0.325	0.100	0.025	0.350	0.250
Activ.5	0.242	0.167	0.092	0.258	0.267
Activ.6	0.250	-0.017	-0.125	0.158	0.167
Activ.7	0.242	0.117	0.008	0.267	0.192
Activ.8	0.175	0.142	-0.075	0.375	0.192
Activ.9	0.342	0.183	0.092	0.417	0.333
Activ.10	0.183	0.083	-0.042	0.300	0.175

Fig.2. Web output_2

Résultats de l'analyse MCAP

Parameters Used

Model: model2

Scale Type: 0-1

MCAP Function: mean

Graphiques

Distribution des poids

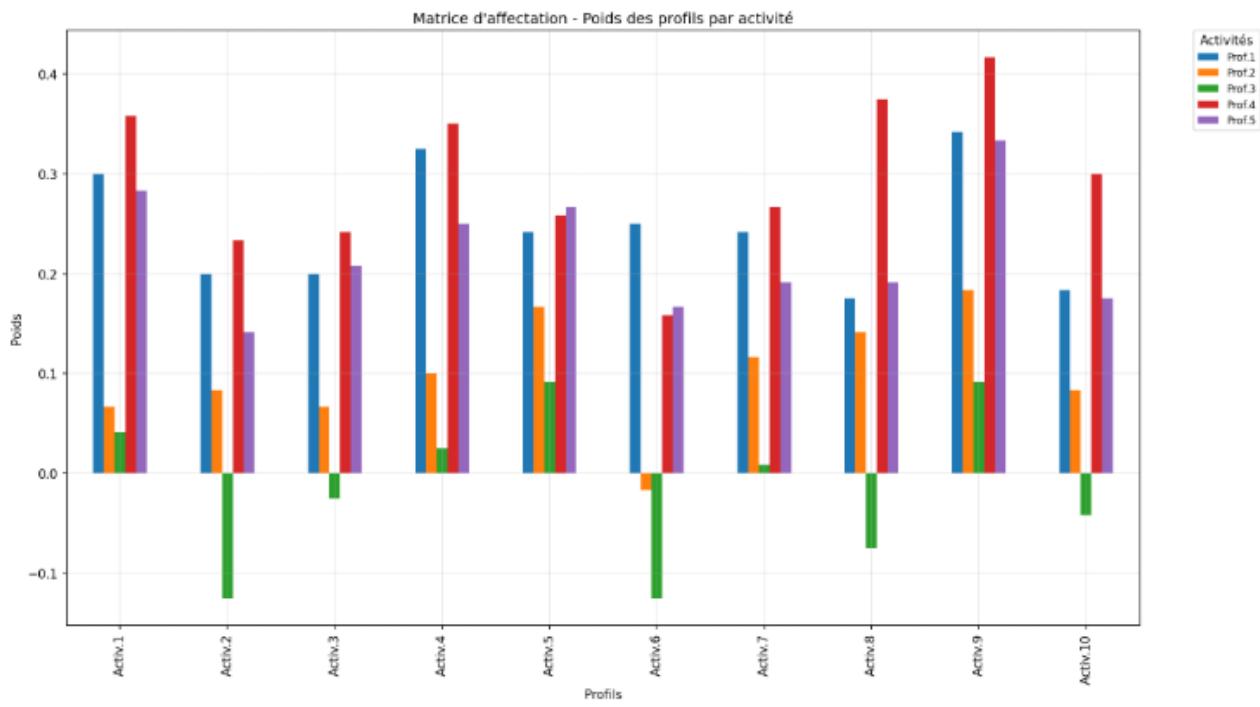
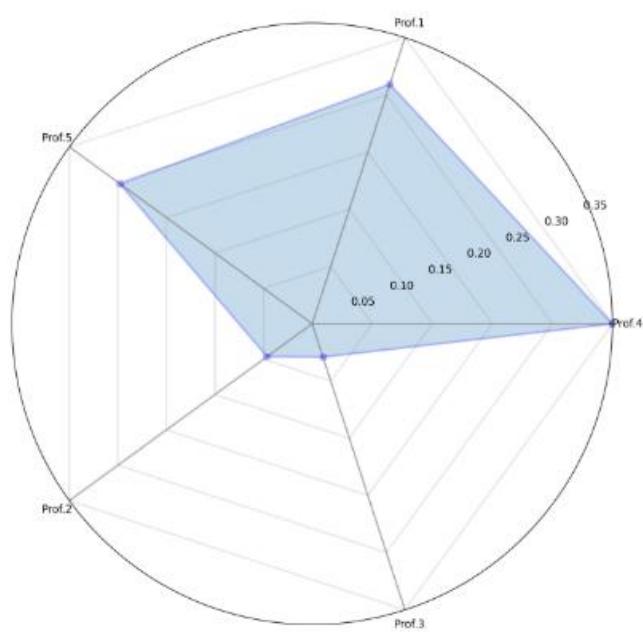


Fig.3. Web output_3

Graphiques radar par activité

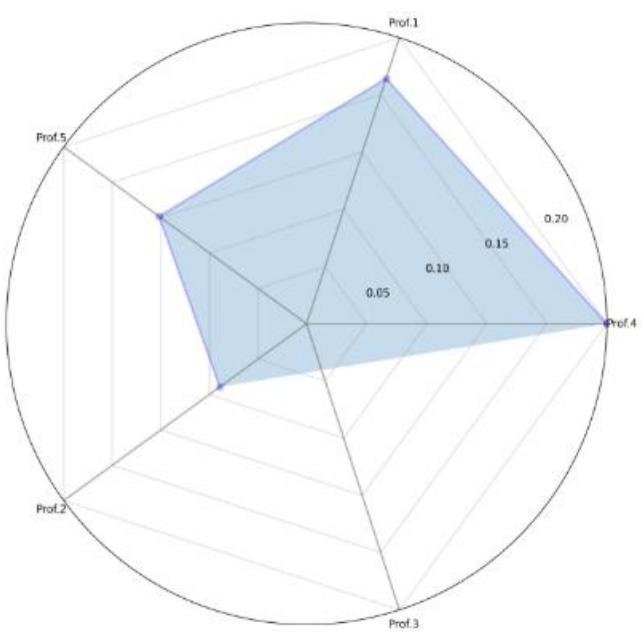
Activité Activ.1

Radar Plot - Activ.1



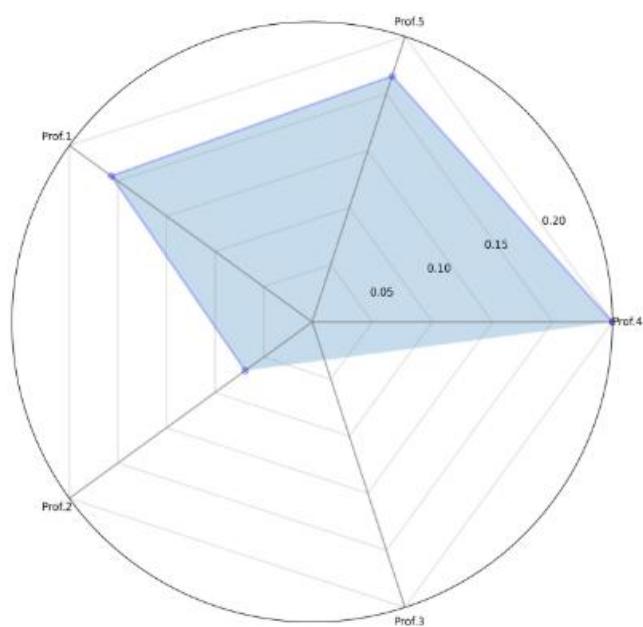
Activité Activ.2

Radar Plot - Activ.2



Activité Activ.3

Radar Plot - Activ.3



Activité Activ.4

Radar Plot - Activ.4

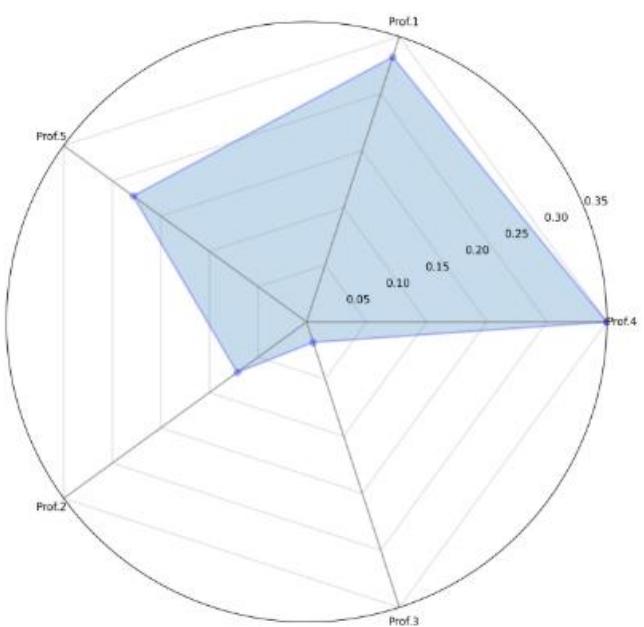


Fig.4. Web output_4 – extract of some radar graphs

Streamlit demo application

```
streamlit.cmd run .\src\streamlit\app.py
```

You should have three menu items:

- Start page
- Test application
- Interactive application

Testing panel is equivalent to the WEB application and looks like this:

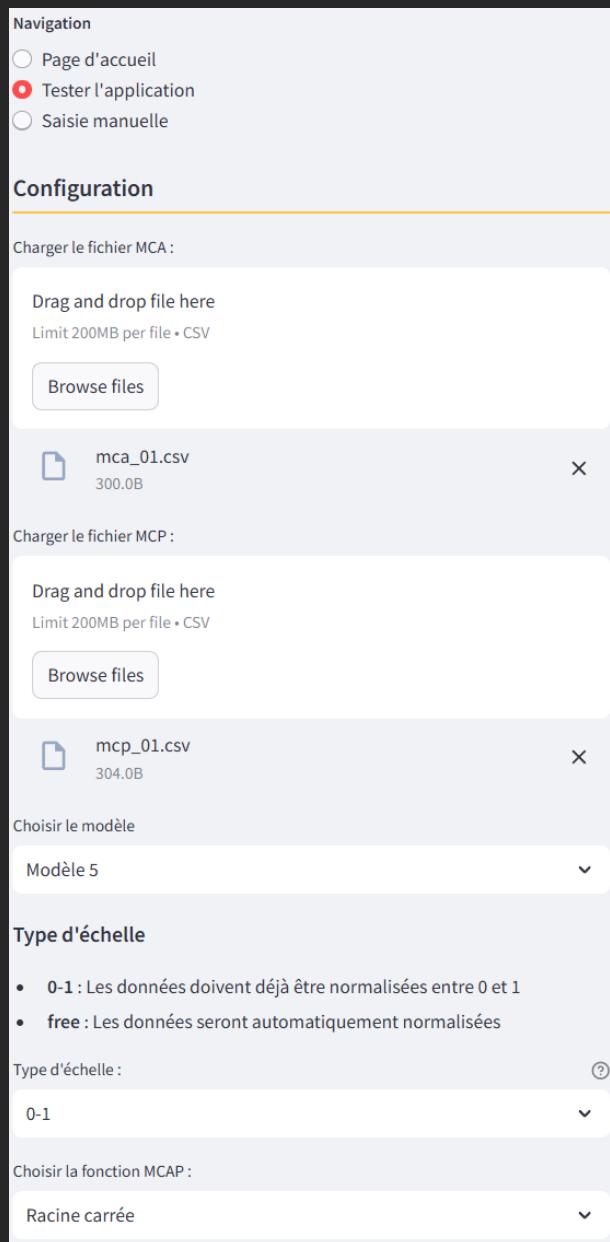


Fig.5. Streamlit input data and options

Interactive application panel looks like this:

Affectation des profils aux activités

Navigation

- Page d'accueil
- Tester l'application
- Saisie manuelle

Configuration

Choisir le modèle :

Modèle 5

Type d'échelle

free

Choisir la fonction MCAP

Moyenne

Saisie manuelle des données

Nombre d'activités

4

-
+

Nombre de profils

4

-
+

Nombre de compétences

4

Matrice MCA (Compétences des Activités)

Exporter MCA

	Comp1	Comp2	Comp3	Comp4
Act1	2	1	2	4
Act2	5	4	4	2
Act3	2	0	1	3
Act4	4	0	4	4

Matrice MCP (Compétences des Profils)

Exporter MCP

	Comp1	Comp2	Comp3	Comp4
Prof1	5	4	3	1
Prof2	1	3	3	2
Prof3	3	2	3	3
Prof4	4	1	2	0

Dimensions de la matrice MCA : (4, 4)

Dimensions de la matrice MCP : (4, 4)

Exporter MCP

Lancer le traitement

Traitement en cours...

Fig.6. Streamlit output – interactive case

11

📁 Project Structure

```
profiles_assignment/
├── src/                      # Source code
│   ├── core/                  # Core processing logic
│   │   ├── __init__.py
│   │   └── mcap_processor.py
│   ├── models/                 # Model implementations
│   │   ├── __init__.py
│   │   ├── mcap_functions.py
│   │   └── model_functions.py
│   ├── utils/                  # Utility functions
│   │   ├── __init__.py
│   │   └── logger.py
│   └── streamlit/              # Streamlit app components
│       └── app.py
└── web/                       # Web application
    ├── backend/                # FastAPI backend
    │   ├── app/
    │   │   ├── routes.py
    │   │   ├── models.py
    │   │   └── database.py
    │   ├── config/
    │   └── main.py
    └── frontend/                # React frontend
        ├── public/
        └── src/
├── config/                    # Configuration files
│   └── mylogger.ini            # Logging configuration
├── data/                      # Data files
│   ├── input/                  # Input CSV files
│   └── output/                 # Generated outputs
│       └── figures/            # Generated plots
├── tests/                     # Test suite
├── requirements.txt            # Python dependencies
├── main.py                    # CLI entry point
└── README.md                  # Project documentation
```

📄 License

This project is open source and available under the MIT License.

🤝 Contributing

Contributions are welcome! Please feel free to submit a Pull Request.

VIDEOS

 [WEB DEMO](#)

 [STREAMLIT DEMO](#)

 [CONSOLE DEMO](#)