



# International Journal of Research Publication and Reviews

Journal homepage: [www.ijrpr.com](http://www.ijrpr.com) ISSN 2582-7421

## Review on Python Packages for Fuzzy MCDA Methods

*Bhavana Lankavalasa<sup>a</sup>, Kavya Sree Maripi<sup>a</sup>, Keerthi Potnuru<sup>a</sup>*

<sup>a</sup>Department of Artificial Intelligence and Machine Learning, GMR Institute of Technology, Rajam, Andhra Pradesh, India

### ABSTRACT:

Fuzzy sets and its extensions are playing crucial role in multi criteria decision making analysis (MCDA). Nowadays, Python is an excellent choice for career paths in software development, DevOps, engineering, machine learning, web development, data analytics, and testing. So, python is implemented as MCDA tool to solve fuzzy MCDA problems. In this regard, pyfdm and pyifdm packages are introduced by authors to solve fuzzy MCDA and intuitionistic fuzzy MCDA problems. In this manuscript, we analyzing about pyfdm and pyifdm packages and we verify with examples

Keywords: Python, MCDA, pyfdm, pyifdm

### Introduction

The use of decision support systems is essential for assisting decision-makers in reaching logical and sensible conclusions [1]. Multi-Criteria Decision Analysis (MCDA) techniques are employed in their development, which rely on the utilisation of original facts and their change to assess the desirability of decision variations. These should include a decision matrix that will be used to assess the options and details on the criteria that have been selected and their applicability to the issue [4]. The approach or combination of procedures to be utilised during the evaluation should be chosen because there are numerous methodologies available for multi-criteria analysis [5]. It is also feasible to select extra factors, including normalization or other metrics that have an immediate effect on the outcomes [6]. A ranking that indicates the relative attractiveness of the various decision variations can be inferred from the alternatives' computed preference values [7]. Furthermore, given the outcomes produced by the various procedures may vary, it is important to ascertain the connection of the results when employing a combination of methods [8]. Since the DSS's operation is meant to help the decision-maker with the selection process so that the data displayed in the diagrams can be more representative of the decision-maker, the visualisation of the results is also a crucial component [9, 10]. Fuzzy logic has been added to the fundamental workings of MCDA systems, expanding their scope of applicability and enabling the handling of uncertain data [11]. Due to the fact that partial data or inaccurate measurements are frequently present in real-world problems, it has led to more practical opportunities [12].

The application of Intuitionistic Fuzzy Sets (IFS) is a widely used extension for modelling uncertain data [13]. According to [14], they permit the data to be expressed as a set  $(\mu, \nu)$ , where  $\mu$  denotes the degree of membership and  $\nu$  the degree of non-membership. Furthermore, a value for the parameter  $\pi$ , which represents the degree of uncertainty, may be obtained from these parameters:  $\pi = 1 - \mu - \nu$  [15]. This allows for the determination of the problem's uncertainty distribution, which easily represents the uncertain data and offers a straightforward means of representing it [16]. It is feasible to discern if a value is more true or more false using the membership and non-membership degree parameters, which is not achievable with classical logic [17]. The increasing number of publicly available tools that permit the developed techniques to be freely employed in a multi-criteria setting is not a result of the growing popularity of MCDA methods. Basic multi-criteria analysis capabilities is offered by a few open-access tools [21]. Nevertheless, they are mostly based on conventional methods from the field, are frequently out of date, and lack recently developed techniques [22]. Commercial solutions are also available, but their cost and the requirement to purchase capabilities restrict who can use them [24]. Furthermore, there aren't many tools that ensure working with intuitionistic fuzzy sets and fuzzy data.

In order to achieve this, the authors of [25, 26] present the PyFDM and PyIFDM packages, which, respectively, enable the execution of multi-criteria assessments in the fuzzy and intuitionistic fuzzy environments. These packages offer a range of methods that can be applied at any point in the decision-making process in relation to the uncertainties present in the issue. These are written in Python 3 and are available for free download through the Python Package Index (PyPI) for people working in the scientific community or on personal projects. The documentation for the suggested tool can be accessed on the Read the Docs website, which serves to familiarise new users with its fundamental presumptions and workings. Via the GitHub repository, the source code is publicly accessible [27, 28]. We examined and examined the pyfdm and pyifdm packages in detail in this study because we believe that this concept is currently popular and could pave the way for further investigation.

The paper is organized as follows. Section 2 shows the software description, including the package's architecture, functionalities, and sample code snippet. Section 3 presents the illustrative example of PyIFDM usage in two multi-criteria problems concerning smart city logistics and

green building energy-savings design projects. Section 4 describes the impact of the proposed tool in the field of decision-making. Finally, in Section 5, the conclusions from the research are drawn, and future directions of the tool development are indicated.

## Description of Packages

In this section, we presenting about the structure of pyfdm and pyifdm packages. These two packages having the same kind of relations between subsequent library modules. The packages making the MCDM evaluation to be user control and modify each step in calculation. First, library is to be helpers which helps the user to define their data and how to user the modules. Then, the user can choose the weights, normalization, distance measure and score functions. After that, the user can select the appropriate or needed MCDM methods to get the results. At last, the results are verified by various correlation methods in which user can choose the correlation technique. In addition to this, we can able clarify the results using graphs and heat flowing graphs.

## Modules of Packages

The pyfdm and pyifdm offers a comprehensive and adaptable tool for conducting multi-criteria assessments within an intuitionistic fuzzy environment. It encompasses various techniques available for user application, including F-MCDA and IF-MCDA methods, objective weighting methods, normalizations, score functions, distance measures, correlation coefficients, helper methods, and a validator. The following Table 1 and Table 2 describes the modules of pyfdm and pyifdm in detail, respectively.

Table 1 Modules in pyfdm package

Package	Module	Defined methods
Pyfdm	Methods	fVIKOR, fTOPSIS, fOCRA, fMOORA, fMAIRCA, fMABAC, fEDAS, fCOPRAS, fCODAS, fARAS
	Weights	Equal, Shannon entropy, Standard deviation, Variance
	Defuzzifications	Mean, Graded mean average, Mean area, Weighted mean
	Normalizations	Sum, Linear, Max, Min-Max, Saw normalization, vector
	Correlations	Spearman correlation coefficient, Pearson correlation Coefficient, Spearman weighted correlation coefficient and WS ranking similarity coefficient
	Distance measures	Euclidean, Weighted Euclidean, Hamming, Weighted Hamming, Vertex, Tran Duckstein, L-R, Mahdavi
	Helpers	It includes additional methods that can allow for the calculation of rankings based on obtained rank and the random generation of a Triangular Fuzzy Number decision matrix of a specified size.

Table 2 Modules in pyifdm package

Package	Module	Defined methods
Pyifdm	Methods	IF-VIKOR, IF-TOPSIS, IF-MOORA, IF-MAIRCA, IF-MABAC, IF-EDAS, IF-CORPAS, IF-CODAS, IF-ARAS
	Weights	Ye entropy, Thakur entropy, Szmidt entropy, Liu entropy, Entropy, Equal, Burillo entropy
	Score functions	Zhang Xu score 1, Zhang Xu score 2, Wei score, Wan Dong score 1, Wan Dong score 2, Thakur score, Supriya score, Liu Wang score, Kharal score 1, Kharal score 2, Chen score 1, Chen score 2
	Normalizations	Swap, Min-Max, Ecer, Supriya, Swap
	Correlations	WS ranking similarity coefficient, Spearman weighted correlation coefficient, Spearman correlation coefficient, Pearson correlation coefficient
	Distance measures	Yang Chiclana, Wang Xin 1, Wang Xin 2, Normalized Hamming, Normalized Euclidean, Lou, Hamming, Grzegorzewski, Euclidean
	Helpers	It includes methods useful for performing multi-criteria evaluations using IFs. These methods enable the calculation of rankings based on obtained preferences and allow users to

		generate a random intuitionistic decision matrix of a specified size.
--	--	---

## Illustrative Examples

To showcase the selected functionalities of the implemented library, two practical examples are provided below. These examples illustrate how the proposed tool can be utilized in multi-criteria problems based on TFNs and IFNs, respectively.

### 4.1 Example 1

The first example is directed to find the best alternative from  $\{A_1, A_2, A_3, A_4\}$ . The fuzzy initial decision matrix, criteria  $\{C_1, C_2, C_3, C_4\}$  and its weights are defined in Table 3. We are going to use fCODAS method from pyfdm package. The sample code snippet is shown in Fig. 1.

Table 3 Trapezoidal fuzzy initial decision matrix

	$C_1$	$C_2$	$C_3$	$C_4$
$A_1$	[2, 8, 5]	[2, 7, 4]	[4, 2, 7]	[7, 1, 3]
$A_2$	[7, 3, 2]	[8, 4, 8]	[1, 8, 9]	[2, 7, 5]
$A_3$	[5, 5, 7]	[4, 4, 8]	[7, 2, 4]	[6, 3, 6]
$A_4$	[6, 4, 3]	[7, 5, 1]	[7, 4, 3]	[4, 7, 9]
Weights	[5, 8, 9]	[6, 7, 9]	[3, 5, 7]	[7, 8, 3]
Types	1	-1	1	1

```
[2]: from pyfdm.methods import fCODAS
from pyfdm.helpers import rank
import numpy as np
if __name__ == '__main__':
    matrix = np.array([
        [[2, 8, 5], [2, 7, 4], [4, 2, 7], [7, 1, 3]],
        [[7, 3, 2], [8, 4, 8], [1, 8, 9], [2, 7, 5]],
        [[5, 5, 7], [4, 4, 8], [7, 2, 4], [6, 3, 6]],
        [[6, 4, 3], [7, 5, 1], [7, 4, 3], [4, 7, 2]]
    ])
    weights = np.array([[5, 8, 9], [6, 7, 9], [3, 5, 7], [7, 8, 3]])
    types = np.array([1, -1, 1, 1])
    f_codas = fCODAS()
    pref = f_codas(matrix, weights, types)

    print(f'Fuzzy CODAS preferences: {pref}')
    print(f'Fuzzy CODAS ranking: {rank(pref)}')
```

Fuzzy CODAS preferences: [ 6.42955672 -17.17611973 0.02601401 10.720549 ]  
Fuzzy CODAS ranking: [2. 4. 3. 1.]

Figure 1 Code snippet for pyfdm package

Now, we applied the pyfdm package and we applied the F-CODAS module to solve this fuzzy information and code snippet is given in Figure 1. From the results, we can finalize that  $A_4 > A_1 > A_3 > A_2$ . Similarly, other methods can be applied to solve this fuzzy MCDA problem and also the graphical visualization is available in pyfdm.

### 4.2 Example 2

In the second example, let us consider the alternatives are  $\{A_1, A_2, A_3, A_4\}$  and the criteria are  $\{C_1, C_2, C_3, C_4, C_5\}$ . Intuitionistic fuzzy values for each alternatives with respect to each criteria are given in Table 4. Also, the criteria weights and its types (1 refers beneficial type and -1 refers non-beneficial type) are mentioned in Table 4.

Table 4 Intuitionistic fuzzy decision matrix

	$C_1$	$C_2$	$C_3$	$C_4$	$C_5$
$A_1$	[0.6000, 0.3000]	[0.6338, 0.2649]	[0.5712, 0.3277]	[0.5885, 0.3064]	[0.7000, 0.2000]
$A_2$	[0.5384, 0.3608]	[0.7000, 0.2000]	[0.5331, 0.3662]	[0.7000, 0.2000]	[0.6674, 0.2313]
$A_3$	[0.6367, 0.2619]	[0.3659, 0.5338]	[0.4105, 0.4850]	[0.3376, 0.5621]	[0.6367, 0.2619]
$A_4$	[0.6392, 0.2594]	[0.5000, 0.4000]	[0.5384, 0.3608]	[0.2350, 0.6648]	[0.6392, 0.2594]
$A_5$	[0.6338, 0.2649]	[0.3324, 0.5673]	[0.3659, 0.5338]	[0.2000, 0.7000]	[0.5712, 0.3277]
Weights	0.42	0.25	0.12	0.13	0.08
Types	1	-1	1	-1	1

To solve this intuitionistic fuzzy information, we applied IF-EDAS module, which is defined in pyifdm package. The code snippet for IF-EDAS method is shown in Figure 2. From this, we can find the rank of alternatives  $A_5 > A_3 > A_4 > A_1 > A_2$ . Similarly, we can use the other MCDA methods to solve intuitionistic fuzzy MCDA issues.

```
[1]: import numpy as np
import random
from pyifdm.methods import ifEDAS
from pyifdm.helpers import rank
from pyifdm.methods import ifs
from pyifdm.methods.ifs.normalization import max_normalization

if __name__ == '__main__':
    matrix = np.array([
        [[0.6, 0.3], [0.6338, 0.2649], [0.5712, 0.3277], [0.5885, 0.3064], [0.7, 0.2]
        ],
        [[0.5384, 0.3608], [0.7, 0.2], [0.5331, 0.3662], [0.7, 0.2], [0.6674, 0.2313]
        ],
        [[0.6367, 0.2619], [0.3659, 0.5338], [0.4105, 0.4850], [0.3376, 0.5621], [0.6367, 0.2619]
        ],
        [[0.6392, 0.2594], [0.5, 0.4], [0.5384, 0.3608], [0.2350, 0.6648], [0.6392, 0.2594]
        ],
        [[0.6338, 0.2649], [0.3324, 0.5673], [0.3659, 0.5338], [0.2, 0.7], [0.5712, 0.3277]
        ]
    ])
    weights = np.array([0.42, 0.25, 0.12, 0.13, 0.08])
    types = np.array([1, -1, 1, -1, 1])
    if_edas = ifEDAS()
    pref = if_edas(matrix, weights, types)
    print(f'IF-EDAS Preferences: {pref}')
    print(f'IF-EDAS Ranking: {rank(pref)}')

IF-EDAS Preferences: [0.29466783 0.04291956 0.78878462 0.69576775 0.92338378]
IF-EDAS Ranking: [4. 5. 2. 3. 1.]
```

Figure 2 Code snippet for pyifdm package

## Future Scope

Python software is a very user friendly and that is used to compute the complicated problems. Now a days, lots of python packages are created to solve many fuzzy logic problems like fuzzy-logic, trapezoidal-fuzzy-number, fuzzy-set-measures, pyifdm, pyifdm, etc. Also, the researchers developing the python MCDA tools for Pythagorean and single valued intuitionistic fuzzy sets. In future, python with fuzzy extension will give better research path and these packages will be implement in real life problems.

## Conclusion

In this paper, we analyzed about the python packages (pifdm and pyifdm), which is used to solve the fuzzy MCDA problems. First we analyzed about the description of the packages and then we discussed about the described modules in the packages. Finally, we explored the usage of the pyifdm and pyifdm software by presenting illustrative examples. In future, we will apply this packages to solve real life data and we would like to develop the MCDA tools for other fuzzy extensions using python.

## Acknowledgements

We thank Dr. S. Rajareega, Department of Basic Sciences and Humanities, GMRIT, Rajam, Andhra Pradesh for her supervision and continuous support till the completion of this work.

## REFERENCES

- [1] Herrmann JW. Rational decision making. Wiley StatsRef: Stat Ref Online 2014;1–9. M
- [2] Triantaphyllou E, Baig K. The impact of aggregating benefit and cost criteria in four MCDA methods. *IEEE Trans Eng Manage* 2005;52(2):213–26.
- [3] Wątróbski J, Jankowski J, Ziemia P, Karczmarczyk A, Ziolo M. Generalised framework for multi-criteria method selection. *Omega* 2019;86:107–24.
- [4] Bikmukhamedov R, Yeryomin Y, Seitz J. Evaluation of MCDA-based handover algorithms for mobile networks. In: 2016 Eighth international conference on ubiquitous and future networks. ICUFN, IEEE; 2016 p. 810–5.
- [5] Teotónio I, Cabral M, Cruz CO, Silva CM. Decision support system for green roofs investments in residential buildings. *J Clean Prod* 2020;249:119365.
- [6] Hatefi S, Torabi S. A common weight MCDA–DEA approach to construct composite indicators. *Ecol Econom* 2010;70(1):114–20.
- [7] Burnay C, Dargam F, Zarate P. Data visualization for decision-making: an important issue. *Oper Res* 2019;19:853–5.
- [8] Qin X, Luo Y, Tang N, Li G. Making data visualization more efficient and effective: a survey. *VLDB J* 2020;29:93–117.
- [9] Das B, Pal SC. Assessment of groundwater vulnerability to over-exploitation using MCDA, AHP, fuzzy logic and novel ensemble models: a case study of Goghat-I and II blocks of West Bengal, India. *Environ Earth Sci* 2020;79(5):1–16.
- [10] Pelissari R, Oliveira MC, Abackerli AJ, Ben-Amor S, Assumpção MRP. Techniques to model uncertain input data of multi-criteria decision-making problems: a literature review. *Int Trans Oper Res* 2021;28(2):523–59.
- [11] Chen T-Y. Multiple criteria decision analysis using a likelihood-based outranking method based on interval-valued intuitionistic fuzzy sets. *Inform Sci* 2014;286:188–208.
- [12] Atanassov KT. *On intuitionistic fuzzy sets theory*, vol. 283. Springer; 2012.
- [13] Szmidt E, Kacprzyk J. Distances between intuitionistic fuzzy sets. *Fuzzy Sets and Systems* 2000;114(3):505–18.
- [14] Song Y, Fu Q, Wang Y-F, Wang X. Divergence-based cross entropy and uncertainty measures of Atanassov’s intuitionistic fuzzy sets with their application in decision making. *Appl Soft Comput* 2019;84:105703.
- [15] Atanassov KT. On the concept of intuitionistic fuzzy sets. In: *On intuitionistic fuzzy sets theory*. Springer; 2012, p. 1–16.
- [16] Lin L, Yuan X-H, Xia Z-Q. Multicriteria fuzzy decision-making methods based on intuitionistic fuzzy sets. *J Comput System Sci* 2007;73(1):84–8.
- [17] Xue Y, Deng Y. Decision making under measure-based granular uncertainty with intuitionistic fuzzy sets. *Appl Intell* 2021;51:6224–33.
- [18] Whitbeck VS, Kisor Jr. M. A new tool in investment decision-making. *Financ Anal J* 1963;19(3):55–62.
- [19] Baizylidayeva U, Vlasov O, Kuandykov AA, Akhmetov TB, et al. Multicriteria decision support systems. Comparative analysis. *Middle-East J Sci Res* 2013;16(12):1725–30.
- [20] Jakub Więckowski, Bartłomiej Kizielewicz, Wojciech Sałabun, Handling decision-making in Intuitionistic Fuzzy environment: PyIFDM package, *SoftwareX* 22 (2023) 101344
- [21] Więckowski J. PyIFDM library package repository. 2022, <https://github.com/jwieckowski/pyifdm>. [Online; accessed 26-September-2022].
- [22] Więckowski J. Pyifdm library package repository. 2023, <https://github.com/jwieckowski/pyifdm>, [Online; accessed 16-January-2023].