**Fuzzy COPRAS**

This is extension over COPRAS which uses fuzzy numbers instead of direct weights for weighting the criteria in the data. This involve many decision makers giving their perspective or importance on a criteria based on linguistics which involve terms like:- good, very good, average, etc.

User are required to provide a fuzzy table to the module since for different scenarios, different fuzzy table is deployed and there is no standard fuzzy table which will generalized all scenarios nicely.

The fuzzy table consist of dictionary(if we talk in terms of python) that has key value pair. The key consist of the linguistic term and it has it’s corresponding fuzzy number in form of list. E.g.:- “VH”:-[0.1, 0.3, 0.5] where ‘VH’ represents very high.

Based on the fuzzy table we will convert all the linguistic to its respective fuzzy number according to fuzzy table and we will aggregate all the fuzzy number numbers from different decision such that each alternative have its own fuzzy number which is aggregated. For aggregation we have employed arithmetic mean. We can use other methods too like geometric mean, etc.

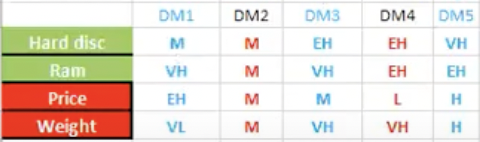
**6.1 Concept algorithm of Fuzzy-COPRAS:**

1) Understanding the inputs: - It is very crucial to know what we take as input. So we will be provided with a data set which consists of various criteria (let there be ‘a’ criteria) and several alternative(let there be ‘b’ alternatives).

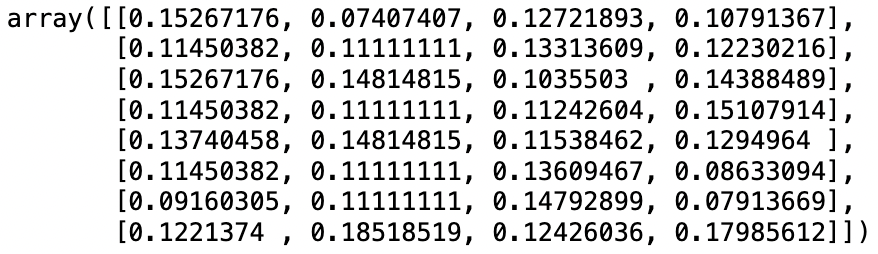


Here the alternative is laptops with criteria denoted with red and green background. Red represents cost criteria and green represents benefit criteria. Here a=4 which is number of criteria and b=8 which is cardinality of alternative.

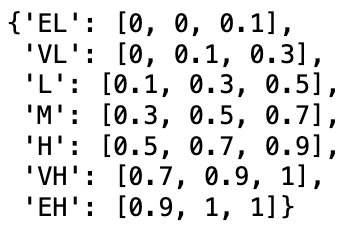
The second thing we get is different decision(their cardinality denoted by ‘c’) maker giving their perspective on different criteria based on linguistic terms.



2) Now we have to calculate the normalized data where every element is divided by sum of element in respective column. The normalized data will be multiplied with weights acquired so as to get ‘fuzzy weighted normalized matrix’.



3) Now in the decision-making dataset we map all the linguistic to its corresponding fuzzy number provided in the fuzzy table given by user.



The fuzzy table.

So the fuzzy decision making matrix looks like this:-

A table with numbers and symbols

Description automatically generated

This matrix has a shape of (a, c, 3) which is equals to (4, 5, 3) as 4 is the number of criteria, 5 is the number of decision makers and 3 is size of fuzzy list.

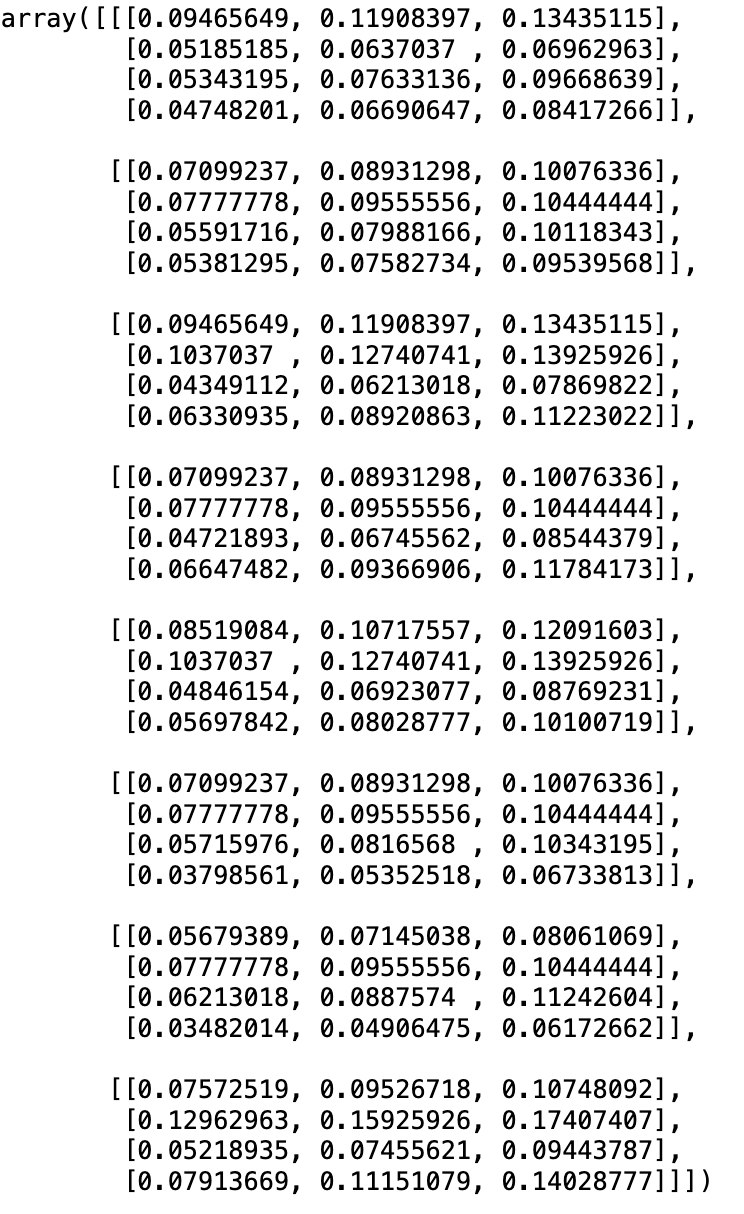
4) Now find the aggregated fuzzy decision matrix which is done on the basic of arithmetic mean.

A number of numbers and a number of numbers

Description automatically generated with medium confidence

This has a shape of (a, 3) where a is number of criteria (here 4) and 3 is size of fuzzy list.

5) Now multiply this fuzzy aggregated weight in the normalized data in order to get the ‘fuzzy weighted normalized matrix’. This will be:-



This will have a shape of (b, a, 3) where b is number of alternatives (here b=8), a is number of criteria(here a is 4) and 3 is size of fuzzy list.

6) Now we find sum of all the fuzzy number of all the criteria which is benefit and assign it to bi. Similarly we find sum of all the fuzzy number of all the criteria which is cost and assign it to ci.

So we get bi and ci which is crucial for determining the rank of different alternative. Based on this we can easily calculate the rank of the alternative and figure out the most suitable alternative. So bi and ci will look here as:-

A black and white text with numbers

Description automatically generated

Bi

A black and white text with numbers

Description automatically generated

Ci

Both have shape as (b, 1) where b is number of alternatives. Here b is 8.

6)Calculate qi, qi can be given as in term of ci and bi as:-

A math equations on a white background

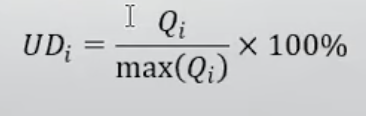
Description automatically generated

So we compute the value of qi for every alternative which turns out to be:-

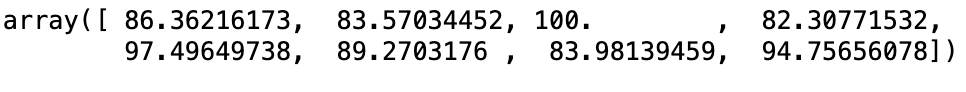
A black and white text with numbers

Description automatically generated

7)And so we can compute rank array or UDi for each alternative which can be given as: -



And once we compute this, it will turn out to be:-



8)Now all we have to do is rank the alternative based in their scores:-

A table with numbers and letters

Description automatically generated with medium confidence

The result shows that the last alternative “Lenovo” fits as the best alternative for the given criterions.

**6.2 Package Code:**

Input Field:





1. **criteria\_names**: A list of strings representing the names of the criteria.
2. **alternative\_names**: A list of strings representing the names of the alternatives.
3. **data**: A numpy array of shape (8, 4) where each row corresponds to an alternative and each column to a criterion.
4. **benificial\_cost\_mark**: A list or numpy array indicating whether each criterion is a benefit (1) or a cost (0).
5. **decision\_makers**: Either a list of strings representing the names of decision makers or simply an integer indicating the number of decision makers.
6. **criteria\_comparison**: A numpy array where each element represents a linguistic term assigned by a decision maker to each criterion.
7. **fuzzy\_table**: A dictionary mapping linguistic terms to their corresponding fuzzy numbers represented as a list with three elements: [lower, medium, upper].

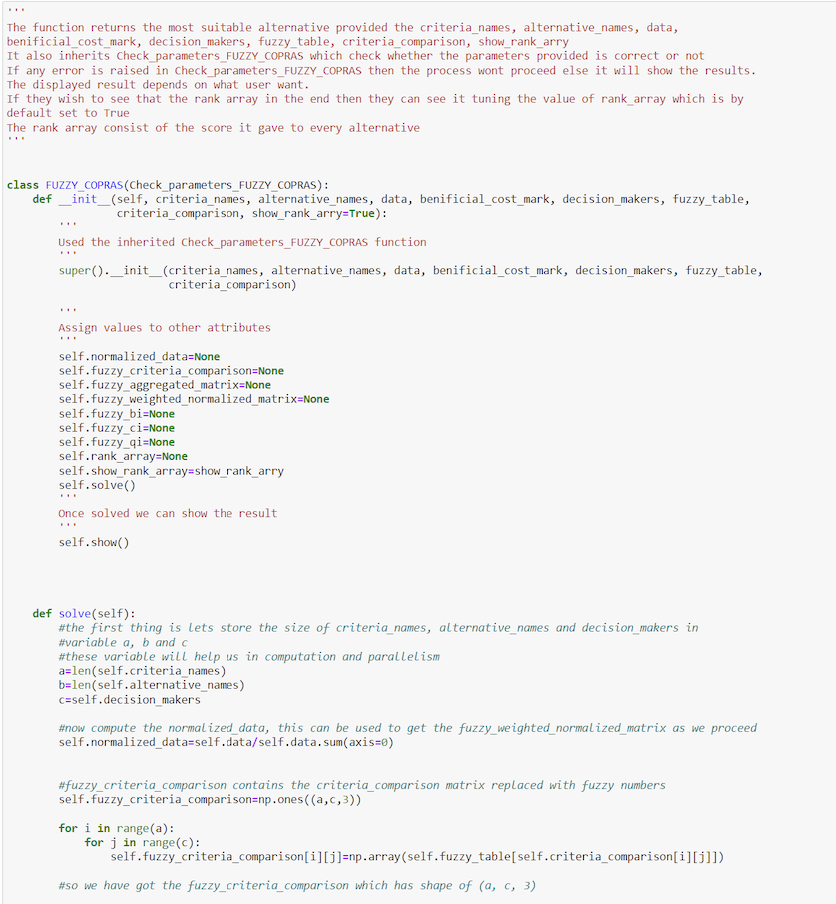
Parameter Check:-

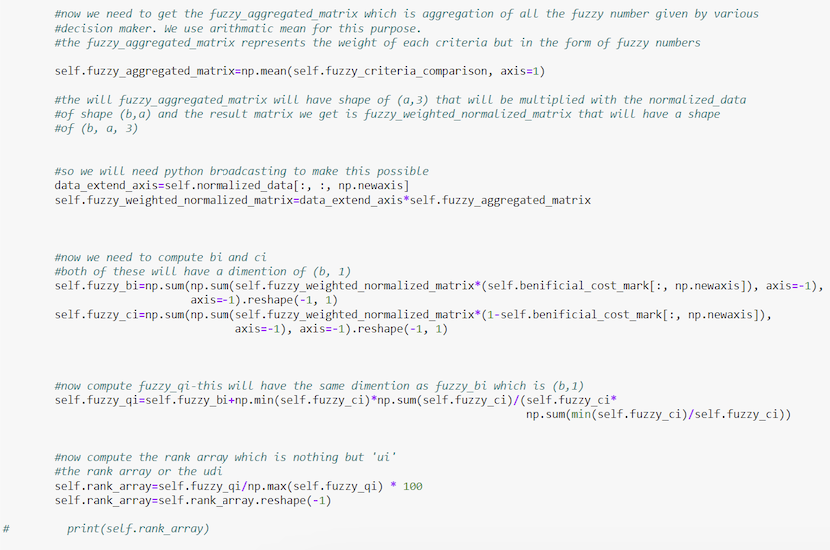




The “parameter\_check” check all the contains of the given input.

**Fuzzy Copras:-** This class find the result that computes result and figure out the best alternative.





**Output:-** The most suitable alternative is shown here.

A table with numbers and letters

Description automatically generated with medium confidence

A person can utilize different attributes according to their requirement:-

1)Self.normalized\_data :-it contains the normalized data.



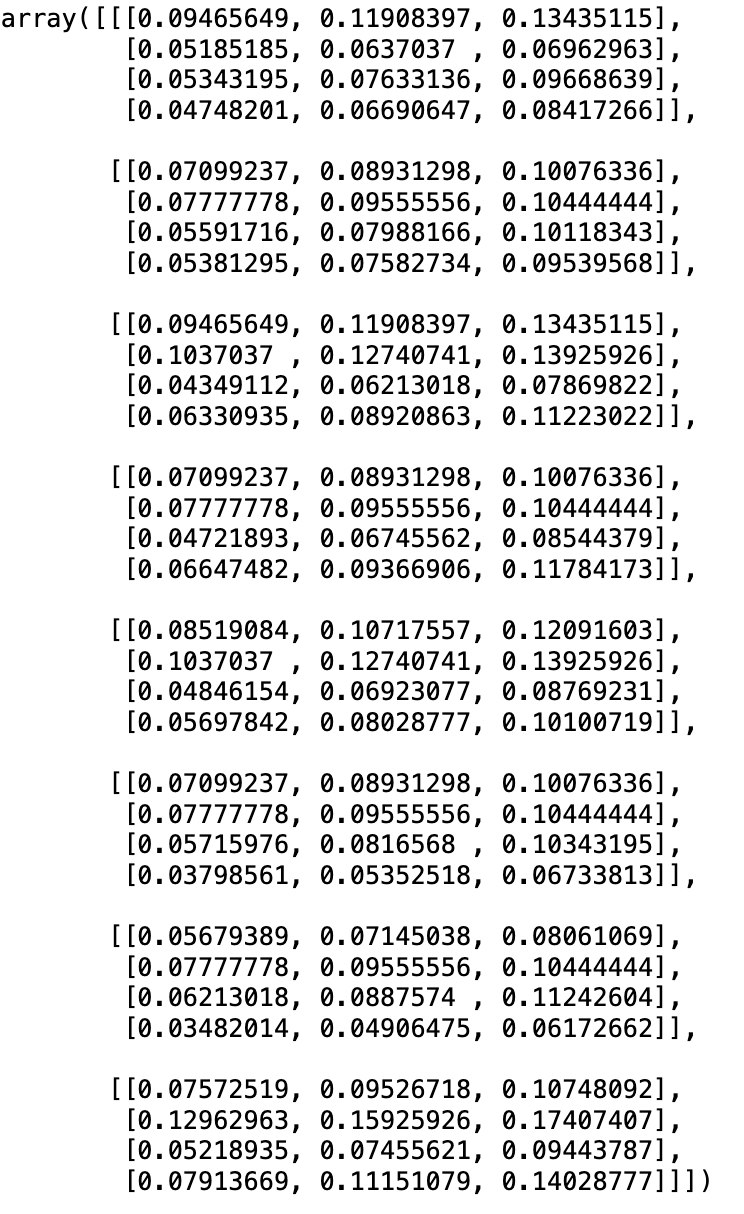
2)Self. fuzzy\_criteria\_comparison:- it contains fuzzy number for each respective linguistic provided by decision makers.

3)Self.fuzzy\_aggregated\_matrix:- it contains aggregated score of all the fuzzy number provide by each decision makers.

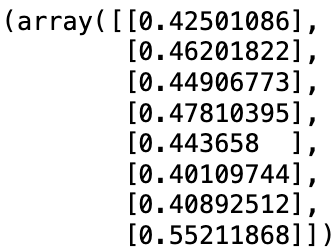
A number of numbers and symbols

Description automatically generated with medium confidence

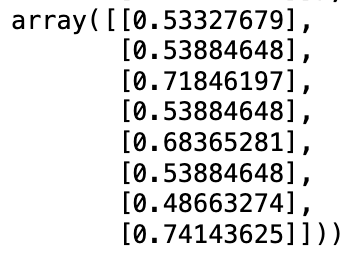
4)Self. fuzzy\_weighted\_normalized\_matrix:-



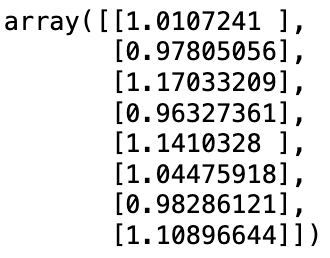
5)Self.fuzzy\_ci:-



6)Self.fuzzy\_bi:-



7)Self.fuzzy\_qi:-



8)Self.rank\_array:-

