TOPSIS

TOPSIS, known as Technique for Order of Preference by Similarity to Ideal Solution.

​It compares a set of alternatives based on a pre-specified criterion.​

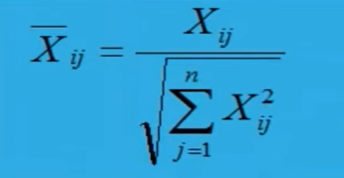
It is based on the concept that the chosen alternative should have the shortest geometric distance from the best solution and the longest geometric distance from the worst solution.

* 1. **Key concepts in TOPSIS:**

Step 1- Create a matrix consisting of M alternatives and N criteria. This matrix is usually called an “evaluation matrix”.​

Step 2- Normalize evaluation matrix.

For this we are going to use the formula –



Step 3- Calculate the weighted normalized decision matrix. It is important​ that each criterion should have its own weight so that all of them will sum up to 1. The weights can be derived randomly (not recommended) or based on Expert knowledge (industry standard).

Step 4-

Determine the best and the worst alternative for each criterion.​

The ideal best for the beneficial criteria is the entity with highest price and for non-beneficial it is the lowest and vice versa.

Step 5-

The ideal best for the beneficial criteria is the entity with highest price and for non-beneficial it is the lowest and vice versa.

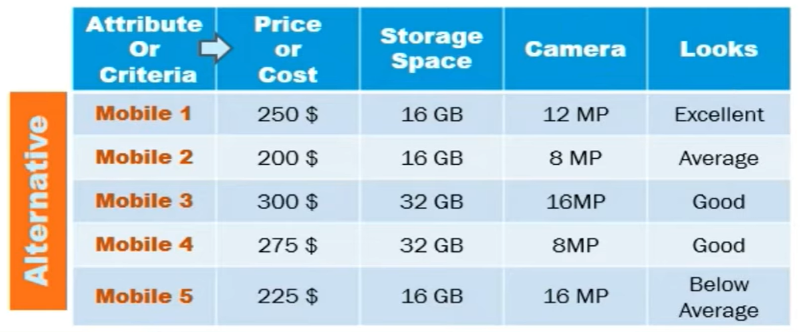
Step 6-

Finally, rank alternatives according to the TOPSIS score by descending order.

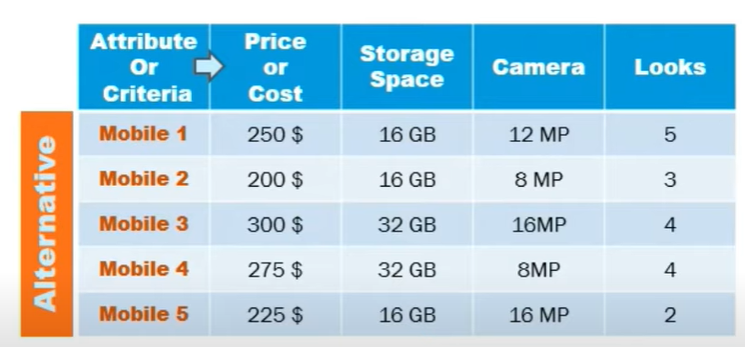
* 1. **A Practical Example: Selecting a new Mobile Set**

Given a problem to select a best Mobile Set from different mobile sets.

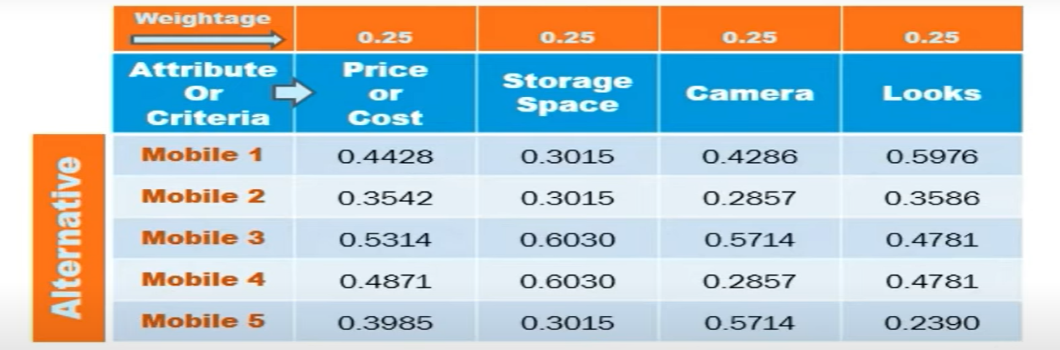
On the basis of features.



Creating an evaluation matrix-



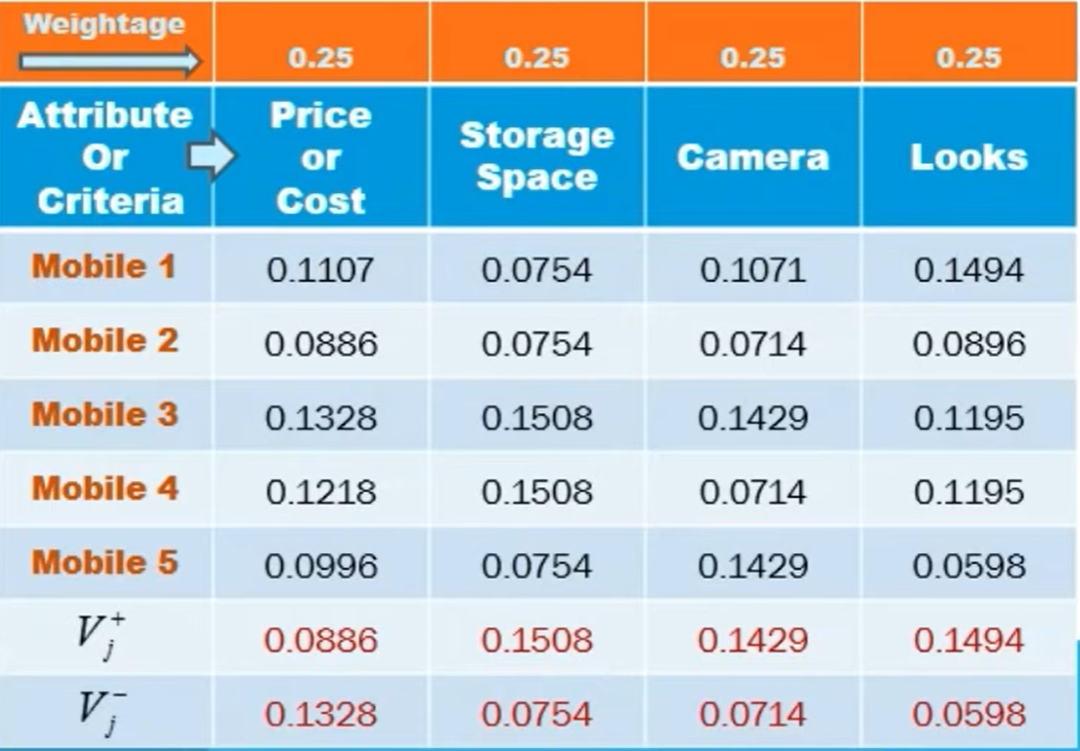
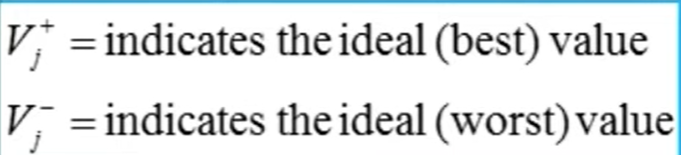
Normalizing it- We have use vector normalization.



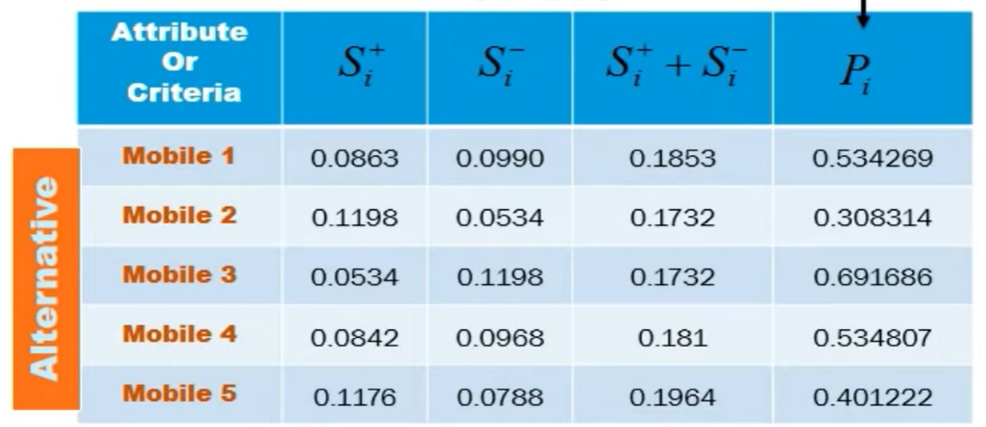
Calculating weighted normalize matrix-

Choosing the best and worst values-



Calculating the TOPSIS scores-



Where-

A black and white image of a number

Description automatically generated

A black and white image of a number

Description automatically generated

And

A black and white text

Description automatically generated

Lastly, ranking the alternatives on the basis of TOPSIS scores-



This consist of attributes along with scores(Pi) and ranks is assigned to them. Here, 3rd alternative which is mobile 3 has a rank of 1 indication it is most suitable alternative given the criteria’s.

* 1. **Package Code**

Code for Topsis is divided in two parts. First part is the parameter check and second part contains the logic of implementing Topsis.

Input Field:



The input details are-

1. Criteria Names and Alternative Names:

- Both `criteria\_names` and `alternative\_names` are lists of strings.

- `criteria\_names` contains the names of the criteria used for evaluation.

- `alternative\_names` contains the names of the alternatives being evaluated

2. Dictionary Encoding (dict\_encode):

- `dict\_encode` is a dictionary that maps non-numeric values to numeric values.

- It is used when the data provided contains non-numeric attributes, such as linguistic terms or qualitative assessments, which need to be encoded into numeric values for processing.

3. Data (DataFrame):

- The `data` variable is a pandas DataFrame representing the evaluation data.

- Each row in the DataFrame corresponds to an alternative, and each column corresponds to a criterion.

- The values in the DataFrame represent the evaluation of each alternative with respect to each criterion.

- Additionally, the index of the DataFrame is set to `alternative\_names`, and the columns are named with `criteria\_names`.

4. Beneficial/Cost Marking (benificial\_cost\_mark):

- `benificial\_cost\_mark` is a list or numpy array indicating whether each criterion is considered a benefit (1) or a cost (0).

- The length of this list/array must be equal to the number of criteria specified in `criteria\_names`.

5. Weights Criteria (weights\_criteria):

- `weights\_criteria` is a list or numpy array containing the weights assigned to each criterion.

- These weights are used to normalize the criteria weights, ensuring they sum up to 1.

- The length of this list/array must be equal to the number of criteria specified in `criteria\_names`.

- If specific weights are provided, the object automatically normalizes them by dividing every element by the sum of the array.

Parameter Check:-





Here are the checks performed in the `Check\_parameters\_TOPSIS` class:

1. Criteria Names and Alternative Names:

- Ensure that both `criteria\_names` and `alternative\_names` are lists.

- Check that every element in these lists is a string.

2. Beneficial/Cost Marking (benificial\_cost\_mark):

- Verify that `benificial\_cost\_mark` is a list or NumPy array.

- If it's a NumPy array, check that it has one dimension and the correct length.

- Ensure that each value in `benificial\_cost\_mark` is either 0 or 1.

3. Weights Criteria (weights\_criteria):

- Check that `weights\_criteria` is a list or NumPy array.

- If it's a NumPy array, check that it has one dimension and the correct length.

- Ensure that every value in `weights\_criteria` is numeric (int or float).

- Normalize the weights by dividing each element by the sum of the array.

4. Data (DataFrame):

- Ensure that `data` is a pandas DataFrame.

- Check that the number of rows in `data` matches the number of alternatives and the number of columns matches the number of criteria.

- If the data contains non-numeric values, check whether `dict\_encode` is provided and verify its format.

- Encode categorical columns in the DataFrame using the provided `dict\_encode`.

- Verify that all encoded values match the keys in `dict\_encode`.

- Convert the DataFrame to a NumPy array of float dtype for further computation.

5. Additional Checks:

- Set a flag `check\_redundancy` to 1 if any of the checks fail.

- Display the message "Parameter check: Passed" if all checks pass.

These checks ensure that the input parameters provided for TOPSIS analysis are valid and consistent, preventing unexpected errors during computation.

Topsis code:

Below is the code for TOPSIS-





Here's a breakdown of its functionality:

1. Initialization:

- Inherits the `Check\_parameters\_TOPSIS` class, ensuring that the input parameters (`criteria\_names`, `alternative\_names`, `data`, `weights\_criteria`, `benificial\_cost\_mark`, and `dict\_encode`) are correctly formatted and appropriate for TOPSIS analysis.

- Assigns the provided input parameters to class attributes.

- Sets default values for additional attributes such as `rank\_array`, `normalized\_matrix`, `vj\_plus`, `vj\_minus`, `si\_plus`, `si\_minus`, `most\_suitable\_alternative`, and `show\_rank\_array`.

2. Solving the TOPSIS:

- Performs the following computations to execute the TOPSIS method:

- Normalization: Normalizes the data matrix `a` using vector normalization, which involves dividing each element by the square root of the sum of squares of the corresponding column.

- Weighted Normalization: Multiplies each element of the normalized matrix by the corresponding weight provided in `weights\_criteria`.

- Ideal Best and Worst Values (`vj\_plus` and `vj\_minus`): Identifies the ideal best and worst values for each criterion by finding the maximum and minimum values along each column of the weighted normalized matrix `t`.

- Handling Cost Criteria: Checks if any criteria are costs based on the `benificial\_cost\_mark` array. If a criterion is a cost, it swaps the `vj\_plus` and `vj\_minus` values.

- Distance Calculation (`si\_plus` and `si\_minus`): Calculates the Euclidean distance between each alternative and the ideal best and worst values for each criterion.

- Ranking (`rank\_array`): Computes the TOPSIS score for each alternative using the formula `(si\_minus / (si\_plus + si\_minus))`. A higher score indicates a better rank.

3. Displaying Results:

- If `show\_rank\_array` is set to `True`, prints the `rank\_array` which contains the TOPSIS score for each alternative.

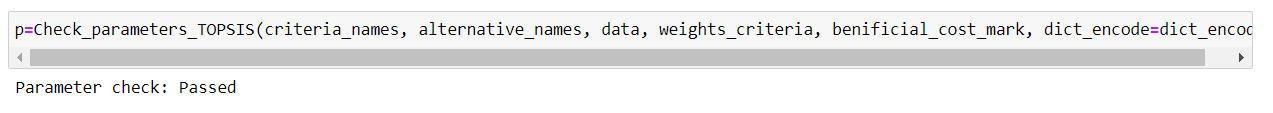
- Calculates the rank of each alternative based on the `rank\_array` and prints both the score and rank of each alternative.

- Determines the most suitable alternative by finding the alternative with the highest score in the `rank\_array` and prints it along with its score.

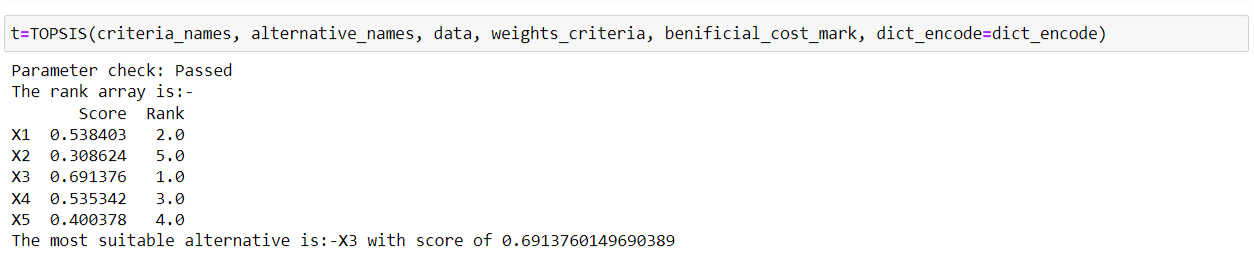
These steps ensure that the TOPSIS analysis is performed accurately and that the results are presented clearly for easy interpretation by the user.

* 1. **Results**

Output of ‘Check\_parameters\_TOPSIS’ –



Result of ‘TOPSIS’-



Other attributes-

