A Novel Method for Ranking Group Performance

Kartikeya Mishra

Independent Researcher

Author Note

Kartikeya Mishra https://orcid.org/0009-0001-3311-3955

Correspondence concerning this article should be addressed to Kartikeya Mishra 50-D, Block E-3, Shatabdi Vihar, Sector 52, Noida, UP - 201307, India. E-mail: kartik.maxwell@gmail.com

Abstract

This paper introduces a novel non-parametric method for calculating group-level performance scores, incorporating rank sums, weight bias adjustments for unequal group sizes, and cross-collection interpretability. Theoretical derivations, Empirical validations and detailed example are discussed.

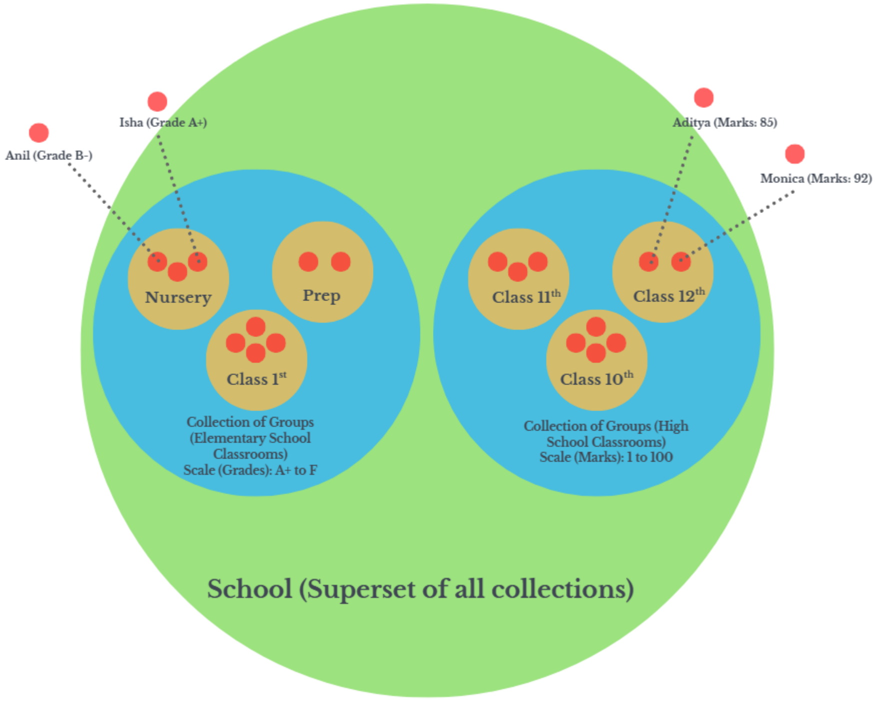
Keywords: Normalized Rank Comparison, Performance Analysis, Weighted Bias Adjustment, Group Performance Metrics, Statistical Methods

A Novel Method for Ranking Group Performance

This formula can numerically identify the performance of a group P_i within a collection of groups containing uneven number of elements. This novel approach takes into account weight bias inherent in the formula itself. The performance measure, P_i numerically falls in the normalized range of [0,1] from non-dominant to dominant performance within their collection, which gives a more interpretable and practical scale, which could be visualized as 0 to 100% dominance in performance via $P_i * 100$. This standardized scale can be used to compare within cross-collection which have different scales, see the Figure 1 to which , this formula can be applied towards.

Figure 1

Superset Collection of Elementary Grade Groups and High School Groups


Isha (Grade A+)
Anil (Grade B-)
Aditya (Marks: 85)
Monica (Marks: 92)
Nursery
Prep
Class 1st
Class 11th
Class 12th
Class 10th
Collection of Groups (Elementary School Classrooms) Scale (Grades): A+ to F
Collection of Groups (High School Classrooms) Scale (Marks): 1 to 100
School (Superset of all collections)

The inspiration for the formula came from the idea from Mann and Whitney, 1947, which uses median to calculate p-value significance. Though, the idea is inspired from their, but applied in terms of assessing performance comparison for groups having uneven elements, and expanded to create a standardize scale with cross-collection interpretability.

Normalized Group Performance Assessment Formulae

$$\boxed{\mathbf{P_i} = \frac{-\mathbf{n_i} + \sum_{\mathbf{j=1}}^{\mathbf{n_i}} \mathbf{r_j}}{\mathbf{n_i} \cdot (\mathbf{N} - \mathbf{1})}} \tag{1}$$

- **Ranks**: Sequential ranks assigned to items within and across groups. Tied ranks are averaged out. Note ranks are in ascending order of dominance i.e. rank 1 is the lowest, with rank N^{th} as the highest. This is done to keep symmetry with usage of ranks in other statistics methods such as Mann and Whitney, 1947 etc.

- P_i: Normalized Performance Measure value of the i^{th} group in the range of [0, 1]

- $\sum r_j$: Sum of ranks for group i.

- n_i: number of elements within the i^{th} group

- N: Total number of elements across all group combined.

  There are

$$k = \text{number of groups}, \quad n_i = \text{number of items in each group}$$

$$\mathrm{N} = \sum_{i}^{k} n_i, \quad \text{where N = total number of items} \tag{2}$$

Applications

This is a list of Applications scenario, the formula was designed for:

- **Psychology**: In Psychology, we have a big arsenal of therapy techniques, all having their own benefits. If, mutually exclusive groups, each group taking a particular therapy, needs to be analyzed which group performed better, and the group sizes are even unequal. Then, Performance Measure Formula can be applied.

- **Educational**: Awarding Best Class Performance in a school, when classes have uneven students, and even different grading scales.

- **Organizational**: To identify best-performing team in the organization, and least-performing, for restructuring and guidance purposes, when number of person per team are uneven, and performance scale at various sector of an organization is different.

- **Business**: Often businesses have economy segment and luxury segment for services and products. Often, just price is not an enough indicator what segment of products fared better. If, a ranking is created, on customer feedback, return on investment, less-after-cost-maintenance, expenses etc. Then, this formula can be used to find the dominant product/services of their business, when there is uneven sale of each product/services sold.

There could be applications outside the scope of this paper. But, when designing the formula these above listed applications were being thought of.

Mathematical derivation, along with its theoretical justification, would be discussed, with its analogous visual intuition where possible. Maxima and Minima, range of normalized performance measure [0, 1], would be proven as well.

Non-normalized Performance

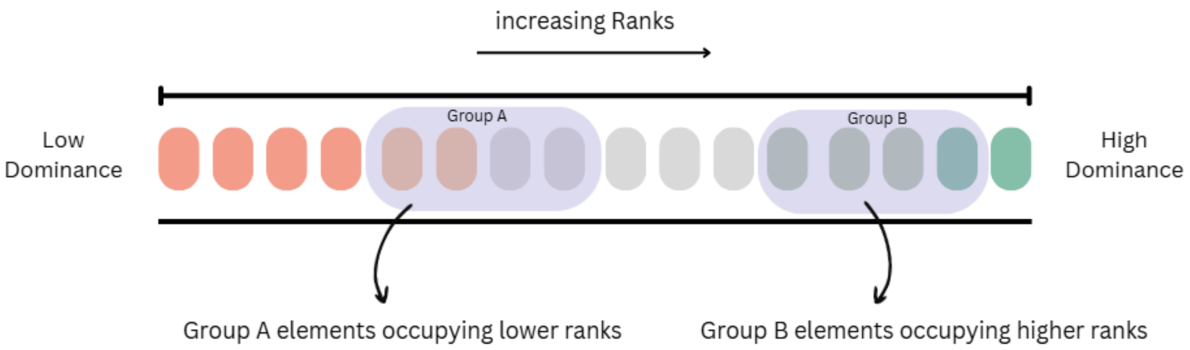
If you see, the Figure 2, is intuitively easy to understand that Group B performance is more dominant, that Group A, simply because Group B occupy the higher ranks, and Group A lower ranks.

This non-normalized performance measure, with uneven number of groups is given by:

$$p_i^{(non\text{-}normalized)} = w_i \cdot \frac{\sum r_j}{S_{UDH}} \tag{3}$$

Figure 2

Visual intuition of the Performance Measure, Performance: Group B > Group A


increasing Ranks
Low Dominance
Group A
Group B
High Dominance
Group A elements occupying lower ranks
Group B elements occupying higher ranks

- Sum of Upper Dominant Half (S_{UDH}):

$$S_{UDH} = \frac{N(N+1)}{2} - \frac{a(a+1)}{2}, \quad a = \lfloor N/2 \rfloor \tag{4}$$

- Weight Bias (w_i):

$$w_i = \frac{N}{kn_i} \tag{5}$$

This later non-normalized dominance would be used to derive the normalized version of the formula. Therefore, non-normalized performance derivation would be discussed first.

Derivation of Non-normalized Performance

From, the understanding we gain from Figure 2, we can easily, ascertain the local dominance, if we divide the scale into two halves of lower ranks half, and upper rank half. So, the group with ranks in the upper half would have a more prominent or dominant performance. So, we need to figure out what's the ranks of the upper dominant half is.

Sum of Upper Dominant Half

1. Purpose of UDH The **Upper Dominant Half (UDH)** represents the **rank sum of the dominant half** of a dataset, providing a benchmark for performance potential in rank-based comparisons.

2. Total Rank Sum (Full Dataset) For a dataset with **N items**, the **total rank sum** is the sum of integers from **1 to N**:

$$S = 1 + 2 + 3 + \ldots + N \tag{6}$$

Using the formula for the sum of the first N integers:

$$S = \frac{N(N+1)}{2} \tag{7}$$

3. Split the Dataset into Halves

   - $a = \lfloor \frac{N}{2} \rfloor$ represents the size of the **non-dominant half** (lower half).
   - The **dominant upper half** includes the rank from $a + 1$ to N^{th} rank items.

4. Rank Sum of Non-Dominant Half (Lower Half) The **non-dominant half** (lower half) consists of the **smallest** a **ranks**. Its rank sum is:

$$S_{lower} = 1 + 2 + \ldots + a \tag{8}$$

Using the formula for sum of integers:

$$S_{lower} = \frac{a(a+1)}{2} \tag{9}$$

5. Rank Sum of Dominant Half (Upper Half) The **Upper Dominant Half** is calculated by subtracting the rank sum of the **lower half** from the **total rank sum**:

$$S_{UDH} = S - S_{lower} \tag{10}$$

Substituting the formulas:

$$S_{UDH} = \frac{N(N+1)}{2} - \frac{a(a+1)}{2} \tag{11}$$

6. Final Formula

$$S_{UDH} = \frac{N(N+1)}{2} - \frac{a(a+1)}{2} \tag{12}$$

- N: Total number of items.
- a: Size of the lower **dominant half**, calculated as the floor of half of the net total number of items across all groups, so as to be inclusive of both odd and even count of N items:

$$a = \lfloor \frac{N}{2} \rfloor \tag{13}$$

Non-normalized Performance Formulae (Biased)

So, to if we compare the ranked sum of the group with the upper dominant half, we can get a numerical understanding of performance measure value

$$p_i^{(biased)} = \frac{\sum r_j}{S_{UDH}} \tag{14}$$

- $\sum r_j$: Sum of ranks for group i.
- S_{UDH}: Sum of Upper Dominant Half

As, the formula doesn't take into account weight bias. A high number of items can still occupy lower ranks, but in account of total items in group being large, its ranked sum also gives a higher value. That's why weigh bias was added to adjust this non-normalized performance measure.

Weight Bias Derivation

To derive the weight bias (w_i), we first understand, that if all the groups have equal number of items, then $w_i = 1$ i.e. no adjustment needed. That means all the groups have the same number of elements in them. The items expected in each group, can be written as:

$$X_e = \frac{N}{k} \tag{15}$$

where:

- X_e: Expected item count per group.
- k: Number of groups in collection.

- N: Total number of items across all groups.

So, now we calculate the fractional change needed to reach expected group item

$$fractional\ change = \frac{X_e - n_i}{n_i} \tag{16}$$

If we want to adjust our biased performance, $p_i^{(biased)}$, according to the fractional change, then,

$$p_i^{(unbiased)} = p_i^{(biased)} + p_i^{(biased)} * fractional\ change \tag{17}$$

in other words, taking $p_i^{(biased)}$ as common, we can rewrite it as

$$p_i^{(unbiased)} = p_i^{(biased)} * (1 + fractional\ change) \tag{18}$$

So, the term needed to adjust from $p_i^{(biased)}$ to $p_i^{(unbiased)}$, can be written as:

$$w_i = 1 + fractional\ change \tag{19}$$

substituting everything we get, the derivation of weight bias:

$$w_i = \frac{N}{kn_i} \tag{20}$$

where:

- w_i: Weight bias for group i.
- n_i: Number of items in group i.
- N: Total number of items.
- k: Number of groups.

Final Formula for unbiased (but non-normalized) Performance Measure

Therefore, from Equation 12, Equation 14, Equation 18, and Equation 20

we derived unbiased(but non-normalized) formulae, as stated in the beginning of the section, in Equation 3:

$$p_i^{(non\text{-}normalized)} = w_i \cdot \frac{\sum r_j}{S_{UDH}} \tag{21}$$

Note in this research paper, $p_i^{(non\text{-}normalized)} = p_i^{(unbiased)}$ are same, but for standardization sake, we will use $p_i^{(non\text{-}normalized)}$ moving forward.

Normalized Performance Measure

The unbiased and non-normalized formulae, is needed to be normalized, for to satisfy these following criteria:

- Explicitly provides a meaningful zero baseline for dominance measures.
- Performance scores become standardized, allowing direct comparisons across multiple studies or scenarios

Given by:

$$\boxed{\mathbf{P_i} = \frac{-\mathbf{n_i} + \sum_{\mathbf{j=1}}^{\mathbf{n_i}} \mathbf{r_j}}{\mathbf{n_i} \cdot (\mathbf{N} - \mathbf{1})}} \tag{22}$$

This formulae is the conclusion of whole paper, which is named Normalized Performance Measure.

Derivation of Normalized Performance Measure

For normalizing the performance we will find the theoretical minima and maxima of non-normalized performance.

Minima of Non-normalized Performance Measure

Thus, we start with the assumption that the net total number of items across all groups is even (odd case would be taken later in the paper), then it is safe to assume that:

$$a = \frac{N}{2}$$

Thus, the lower and upper halves would be explicitly:

$$\left[1, \frac{N}{2}\right] \quad \text{and} \quad \left[\frac{N}{2} + 1, N\right]$$

Simplifying for S_{UDH} when N is even, we get:

$$S_{UDH} = \frac{N(3N+2)}{8} \tag{23}$$

We aim to minimize the non-normalized measure $p_i^{(non\text{-}normalized)}$ defined as:

$$p_i^{(non\text{-}normalized)} = w_i \cdot \frac{\sum r_j}{S_{UDH}} \tag{24}$$

Substituting everything into our non-normalized measure equation, we get:

$$p_i^{(non\text{-}normalized)} = \frac{8}{k(3N+2)} \cdot \frac{\sum r_j}{n_i}$$

As,

$$\frac{8}{k(3N+2)} = Constant \tag{25}$$

The minima, depends on $\frac{\sum r_j}{n_i}$. And, for this to be low $\sum r_j$ should be low. That means r_j have to occupy the lowest ranks such as 1, 2, 3,... so on. We can rewrite $\frac{\sum r_j}{n_i}$ in this format below:

$$= \frac{\sum r_j}{n_i} = \frac{\frac{n_i}{2}(1+n_i)}{n_i} = \frac{1}{2} \cdot n_i + \frac{1}{2} \tag{26}$$

As this is a linearly increasing line equation with a positive slope $(y = mx + C)$, minima of $\frac{\sum r_j}{n_i}$ occurs when $n_i = 1$, as n_i is a natural number and can't be zero or negative, hence, substituting $n_i = 1$, we get, $\frac{\sum r_j}{n_i} = \frac{1}{2} + \frac{1}{2} = 1$, hence $\frac{\sum r_j}{n_i} = 1$

Therefore minima of Non-normalized measure is:

$$p_i^{(min,\, non\text{-}normalized)} = \frac{8}{k(3N+2)} \tag{27}$$

Maxima of Non-normalized Performance Measure

Similar to minima, being:

$$p_i^{(non\text{-}normalized)} = \frac{8}{k(3N+2)} \cdot \frac{\sum r_j}{n_i}$$

As,

$$\frac{8}{k(3N+2)} = Constant \tag{28}$$

Maxima depends on $\frac{\sum r_j}{n_i}$, which needs to be maximized. Therefore, r_j will contain higher ranks, such as N, N-1, N-2,...,k. So, The lowest rank in this arithmetic series can contain is k. Because, there are k groups, that means, each will have at least one element, occupying the lowest ranks, to maximize $\frac{\sum r_j}{n_i}$. That means, first group will contain rank 1, second group will contain rank 2, so on..., and the k^{th} group will have the rest of the elements from rank k^{th} to N^{th} element which we are trying to maximize. Then $\sum r_j$, can be expanded to:

$$\frac{(N-k+1)(N+k)}{2} \tag{29}$$

As, n_i are the number of elements in the group which is equal to $N-k+1$, then simplifying $\frac{\sum r_j}{n_i}$, we get:

$$\frac{N+k}{2} \tag{30}$$

Now to maximize, we can substitute k with N, as k are number of groups, and the max number of groups would be equal to total number of items in the collection, where each group has just one element. hence maximum value comes out to be:

$$\frac{\sum r_j}{n_i} = N \tag{31}$$

Therefore maxima of Non-normalized Measure is:

$$p_i^{(max,\, non\text{-}normalized)} = \frac{8N}{k(3N+2)} \tag{32}$$

Final formualae of Normalized Performance Measure

Normalized Performance measure would be given by:

$$P_i = \frac{p_i^{(non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}}{p_i^{(max,\, non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}} \tag{33}$$

Now substituting their corresponding value from Equation 21, Equation 23, Equation 27 and Equation 32, and simplifying, we get the final form of Normalized Performance Measure:

$$\boxed{\mathbf{P_i} = \frac{-\mathbf{n_i} + \sum\limits_{\mathbf{j=1}}^{\mathbf{n_i}} \mathbf{r_j}}{\mathbf{n_i} \cdot (\mathbf{N} - \mathbf{1})}} \tag{34}$$

- P_i: Normalized Performance Measure value of the i^{th} group in the range of [0, 1]
- $\sum r_j$: Sum of ranks for group i.
- n_i: number of elements within the i^{th} group
- N: Total number of elements across all group combined.

Normalized Performance Measure (Scenario: N is odd)

We will not go in thorough detail, but the Normalized performance measure formula is same. Because the, Sum of Upper Dominant half (S_{UDH}), maxima and minima just differ slightly in nature. But, as these terms occur in the numerator and denominator, they cancel each other out. And, **the Normalized Performance Measure remains unchanged**. Just, to be complete, we will just list down the equations, which result in the same formula as above.

As, N is odd, where $N^{(min)} = 3$, because there has to be minimum of two groups, therefore, we will take,

$$a = \frac{N-1}{2}$$

Thus, the lower and upper halves would be explicitly:

$$\left[1, \frac{N-1}{2}\right] \quad \text{and} \quad \left[\frac{N+1}{2}, N\right]$$

when N is odd:

$$S_{UDH} = \frac{(N+1)(3N+1)}{8} \tag{35}$$

$$p_i^{(min,\, non\text{-}normalized)} = \frac{8N}{k(N+1)(3N+1)} \tag{36}$$

$$p_i^{(max,\, non\text{-}normalized)} = \frac{8N^2}{k(N+1)(3N+1)} \tag{37}$$

substituting their corresponding value in Equation 33 from Equation 21, Equation 35, Equation 36 and Equation 37, and simplifying, we get the final form of Normalized Performance Measure (same as when N is even):

$$\boxed{\mathbf{P_i} = \frac{-\mathbf{n_i} + \sum_{\mathbf{j=1}}^{\mathbf{n_i}} \mathbf{r_j}}{\mathbf{n_i} \cdot (\mathbf{N} - \mathbf{1})}} \tag{38}$$

Range of Normalized Performance Measure

Minima of $P_i = 0$, which means is the least dominant performance, that means in the entire collection they have a group containing, one element with Rank 1. Mathematically, $p_i^{(non\text{-}normalized)} = p_i^{(min,\, non\text{-}normalized)}$, thus, putting the value in the Equation 33, we get:

$$P_i^{(min)} = \frac{p_i^{(min,\, non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}}{p_i^{(max,\, non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}} = 0 \tag{39}$$

Similarly, Maxima of $P_i^{(max)} = 1$, which means is the most dominant performance, that means in the entire collection they have a group containing, one element with Rank N^{th}, the highest ranked element possible. Mathematically, $p_i^{(non\text{-}normalized)} = p_i^{(max,\, non\text{-}normalized)}$, thus, putting the value in the Equation 33, we get:

$$P_i^{(max)} = \frac{p_i^{(max,\, non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}}{p_i^{(max,\, non\text{-}normalized)} - p_i^{(min,\, non\text{-}normalized)}} = 1 \tag{40}$$

Therefore, the range of normalized performance measure is [0,1]

Validation

All mathematical and numerical validation scripts, as well as the full manuscript source, are available in the GitHub repository Kartikeya, 2025. See the repository's folder structure and README for details on specific files and validation types.

Mathematical Validation: Theoretical Derivation via SymPy Library in Python

To, validate all manual work, the formula also has been derived through mathematical coding via SymPy library using Python Language (a open source replacement tool for Mathematica), written inside a Jupyter Notebook.

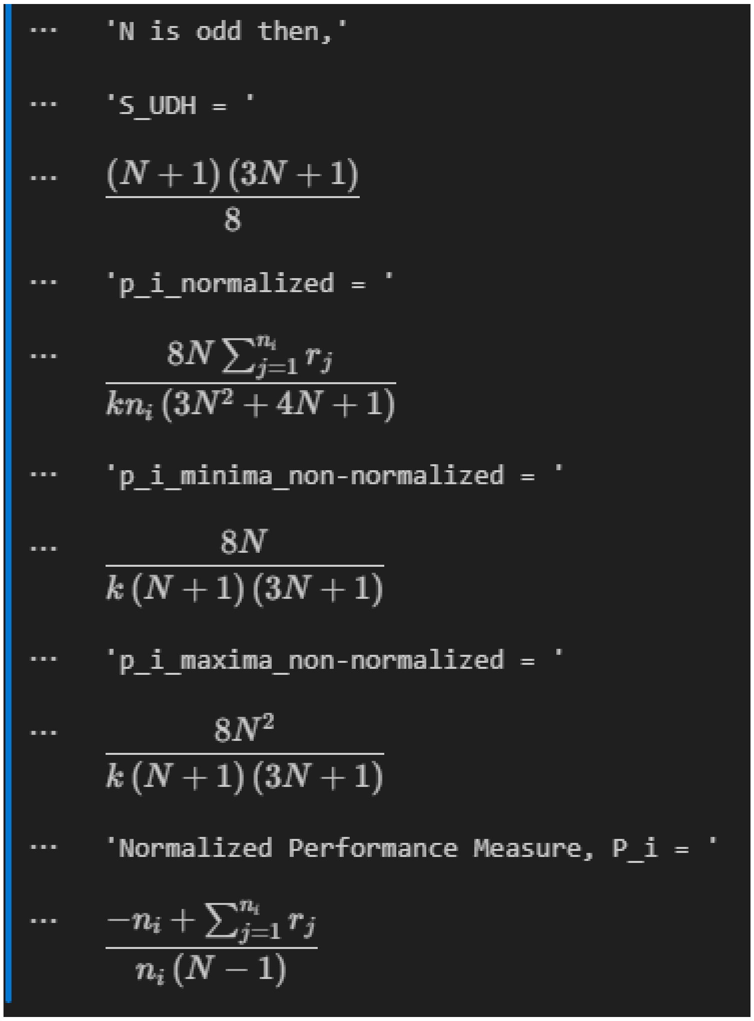
A snippet of the code is hereby attached: Python (SymPy) Code Snippet. This snippet is for: When N is odd, Mathematical Derivation of Normalized Performance Measure.

```
        # code snippet

        display("N is odd then,")

        a = (n - 1)/2
        s_udh = n*(n+1)/2 - a*(a+1)/2
        display("S_UDH = ",simplify(s_udh))

        p_i = w_i * sigma_r_j / s_udh
        display("p_i_normalized = ", simplify(p_i))

        p_i_min = p_i.subs({n_i:1}) # number of elements = 1
        p_i_min = p_i_min.doit()
        p_i_min = p_i_min.subs({r[1]:1}) # containing one element rank = 1
        display("p_i_minima_non-normalized = ",simplify(p_i_min))

        p_i_max = p_i.subs({n_i:1}) # number of elements = 1
        p_i_max = p_i_max.doit()
        p_i_max = p_i_max.subs({r[1]:n}) # containing one element rank = Nth
        display("p_i_maxima_non-normalized = ",simplify(p_i_max))

        P_i = (p_i - p_i_min)/(p_i_max - p_i_min)
        display("Normalized Performance Measure, P_i = ",
         ↪  factor(simplify(P_i)))
```

Figure 3 shows the output of the Snippet Code

Figure 3

Output: Mathematical Derivation of Normalized Performance Measure via SymPy



```
'N is odd then,'
'S_UDH = '
```

$$\frac{(N+1)\,(3N+1)}{8}$$

```
'p_i_normalized = '
```

$$\frac{8N\sum_{j=1}^{n_i} r_j}{kn_i\,(3N^2+4N+1)}$$

```
'p_i_minima_non-normalized = '
```

$$\frac{8N}{k\,(N+1)\,(3N+1)}$$

```
'p_i_maxima_non-normalized = '
```

$$\frac{8N^2}{k\,(N+1)\,(3N+1)}$$

```
'Normalized Performance Measure, P_i = '
```

$$\frac{-n_i+\sum_{j=1}^{n_i} r_j}{n_i\,(N-1)}$$

Numerical Validation: Range of Normalized Performance Measure

We have mathematically and theoretically have proven the range to [0,1]. Zero means the group has single element with Rank 1 element, lowest performance measure value possible. And, one means the group has single element with Rank N^th element, highest performance measure value possible. Any combination of elements in the group will be in the range of [0,1].

We have used this simple numerical example to validate, where value is same with their rank.

Table 1

Example used for Validation of Range for Normalized Performance Measure

Group	Member Value	Rank	Normalized Performance Measure
I	1	1	0.0
II	2	2	0.5
	3	3	
	4	4	
	5	5	
	6	6	
	7	7	
	8	8	
	9	9	
	10	10	
III	11	11	1.0

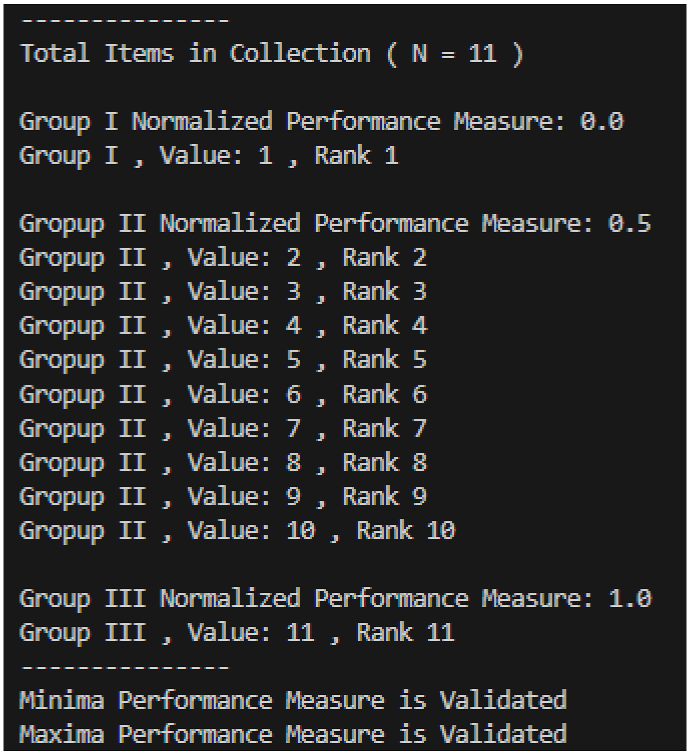
This is numerically validated via Python code, by applying the formulae discussed in the research paper, the code snippet is attached below:

```python
        def setPerformanceMeasureValue(self):
        """Calculates the normalized performance value, using the formula in the research paper
        """
        ni = self.getElementCount()
        sigma_ri = self.getRankedGroupSum()
        n = self.parent.netTotalItemsInCollection
        k = self.parent.getNumberOfGroups()
        self.performanceMeasureValue = (-ni + sigma_ri)/(ni*(n-1))
```

Figure 4 shows the output of Maxima and Minima Validation.

Figure 4

Output: Numerical Validation of Range for Normalized Performance Measure via Python



```
---------------
Total Items in Collection ( N = 11 )

Group I Normalized Performance Measure: 0.0
Group I , Value: 1 , Rank 1

Gropup II Normalized Performance Measure: 0.5
Gropup II , Value: 2 , Rank 2
Gropup II , Value: 3 , Rank 3
Gropup II , Value: 4 , Rank 4
Gropup II , Value: 5 , Rank 5
Gropup II , Value: 6 , Rank 6
Gropup II , Value: 7 , Rank 7
Gropup II , Value: 8 , Rank 8
Gropup II , Value: 9 , Rank 9
Gropup II , Value: 10 , Rank 10

Group III Normalized Performance Measure: 1.0
Group III , Value: 11 , Rank 11
---------------
Minima Performance Measure is Validated
Maxima Performance Measure is Validated
```

Detailed Example of Dominance Method

Validation of Results

- cross-validation from a different method

References

Kartikeya. (2025). Normalized performance measure [Code for Mathematical (SymPy, Jupyter Notebook, Python), Numerical Validation (SciPy, Python), and LaTeX manuscript; accessed 7 July 2025]. https://github.com/silentkarmi/normalized-rank-comparison

Mann, H. B., & Whitney, D. R. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, *18*(1), 50–60. https://doi.org/10.1214/aoms/1177730491