# GRC Rating Algorithm Analysis

Anvi Singh
Cyber Sierra

September 2025

## 1 Introduction

Selecting appropriate Governance, Risk, and Compliance (GRC) software is a multidimensional decision-making problem. With a wide range of available tools and numerous evaluation criteria to consider, organizations face the challenge of synthesizing complex information into actionable metrics.

Multi-criteria decision-making (MCDM) methods offer a formal approach for addressing such challenges. These methods are designed to support decision-makers in aggregating diverse and sometimes conflicting criteria into a single composite score or ranking. This is particularly relevant in the context of GRC software evaluation, where performance can vary significantly across different areas of functionality.

The objective of this document is to explore and compare a selection of MCDM algorithms that are suitable for use in a small-scale analysis involving 50 software products and 10 evaluation criteria. Each criterion has been rated on a 1–5 scale, and the ultimate goal is to compute an overall score, also on a 1–5 scale,for each software option. The analysis emphasizes practical considerations, such as computational feasibility, ease of explanation, and robustness to minor variations in input data.

Given the scope and constraints of this analysis,including the size of the dataset, the number of evaluation criteria, and the need for clear and explainable results, four MCDM algorithms were selected for comparative testing: Weighted Sum Model (WSM), Weighted Product Model (WPM), Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), and VIKOR.

Each algorithm was assessed using several meta-criteria, including ease of implementation, interpretability, robustness, and suitability for small datasets. The final algorithm selection is based on both theoretical properties and empirical performance across the 50 GRC software records.

# 2 Evaluation Criteria for Algorithms

In order to determine which MCDM method is most appropriate for this task, each algorithm will be evaluated according to the following meta-criteria:

- **Ease of Implementation:** The method should be feasible to implement without the need for specialized software or complex programming. Ideally, it should be executable using spreadsheet software or basic scripting languages.
- **Interpretability:** The output and intermediate steps should be easy to understand and explain to non-technical stakeholders. Transparency of the calculation process is a key consideration.
- **Robustness:** The method should yield stable results under small changes in input data or weighting schemes. Highly sensitive methods may not be reliable in practice.
- **Handling of Poor Performance:** The algorithm's ability to penalize low scores appropriately is important in contexts where certain minimum thresholds are non-negotiable (e.g., compliance requirements).
- **Suitability for Small Datasets:** Methods should be efficient and effective when applied to relatively small datasets (e.g., 50 alternatives), without requiring large sample sizes for reliable output.

These criteria serve as the basis for comparing the selected algorithms and justifying the final recommendation.

# 3 Overview of Selected MCDM Algorithms

This section provides an overview of five multi-criteria decision-making (MCDM) algorithms selected for evaluation in this study. Each method offers a distinct approach to aggregating multiple criteria into a single overall score. The following subsections describe the methodology, strengths, and limitations of each algorithm in the context of GRC software evaluation.

## 3.1 Weighted Sum Model (WSM)

The Weighted Sum Model (WSM) method is one of the most widely used MCDM techniques due to its simplicity and intuitive logic. In WSM, each criterion score is multiplied by a predefined weight, and the weighted scores are summed to produce a final score for each alternative.

**Methodology:**

- Normalize criterion scores to a common scale if needed
- Multiply each score by its corresponding weight

- Sum the weighted scores across all criteria

**Advantages:**

- Simple to understand and implement in spreadsheets
- Transparent and easily explainable to stakeholder.
- Efficient for small datasets

**Disadvantages:**

- Fully compensatory;high performance in one criterion can offset poor performance in another, which may not be desirable in all contexts
- Assumes linear relationships among scores and weights

### 3.2 Weighted Product Model (WPM)

The Weighted Product Model (WPM) is a multiplicative alternative to WSM. Instead of summing weighted scores, WPM involves multiplying each criterion score raised to the power of its corresponding weight.

**Methodology:**

- For each alternative, calculate the product of all criterion scores raised to the power of their respective weights
- Optionally normalize the final scores to the desired scale

**Advantages:**

- Handles score variability without requiring normalization
- Penalizes low scores more sharply, making it more sensitive to weak performance

**Disadvantages:**

- Less intuitive than additive models.
- Very low scores in any criterion can significantly reduce the final score
- Slightly more complex to implement due to exponentiation and potential numerical instability

### 3.3 Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS ranks alternatives based on their relative distance from an ideal solution (best possible scores across all criteria) and an anti-ideal solution (worst possible scores). The alternative closest to the ideal and farthest from the anti-ideal is preferred.

**Methodology:**

- Normalize the decision matrix using vector normalization.
- Multiply normalized scores by weights.
- Identify the ideal and anti-ideal solutions.
- Calculate the Euclidean distance of each alternative from both ideal and anti-ideal solutions.
- Compute a closeness score based on these distances.

**Advantages:**

- Balances the concept of aspiration (ideal) and risk (anti-ideal).
- Results are easily interpretable and scalable.

**Disadvantages:**

- Requires additional computational steps, especially normalization and distance calculations.
- Cannot take into account and reflect stakeholder preferences explicitly

## 3.4 VIKOR

VIKOR is a compromise-ranking method that considers both the total performance and the worst-case performance of an alternative. It is designed to identify solutions that provide a balance between group utility and individual regret.

**Methodology:**

- Determine best and worst values for each criterion.
- Compute the weighted sum (S) and the maximum regret (R) for each alternative.
- Calculate a composite index (Q) combining S and R using a compromise coefficient.
- Rank alternatives based on Q.

**Advantages:**

- Effectively captures trade-offs between overall performance and critical weaknesses.
- Useful when decision-makers seek a compromise solution.

**Disadvantages:**

- More computationally intensive than WSM or WPM.
- Less widely understood, which may affect stakeholder confidence in results.

# 4 Dataset and Preprocessing

The dataset used in this study consists of 50 Governance, Risk, and Compliance (GRC) software products. Each product is evaluated across 10 predefined criteria relevant to core functional, technical, and strategic aspects of GRC platforms. These criteria were selected based on industry practices and practical considerations in software evaluation.

## 4.1 Evaluation Criteria

Each software product is assessed using the same set of 10 criteria. While the specific criteria names may vary by organization, typical dimensions considered include:

- Risk Management
- Compliance Management
- Audit Incident Management
- Workflow Automation
- Integration APIs
- Ease of Use
- Monitoring Dashboards
- Vendor Third Party Risks
- Pricing Value
- Scalability Support

Each criterion is rated on a discrete scale from 1 to 5, where 1 indicates very poor performance, 3 indicates average or adequate performance, and 5 indicates excellent performance.

## 4.2 Data Format

The dataset is structured as a matrix with 50 rows (alternatives) and 10 columns (criteria). Each cell contains a score from 1 to 5, representing the evaluation of a given software product on a specific criterion. All scores were determined through human evaluation.

## 4.3 Weight Assignment

To account for the relative importance of different criteria, a weight vector was applied. These weights were assigned manually based on human judgement

The sum of all weights is normalized to 1. These weights are consistently applied across all algorithms that incorporate them.

## 4.4 Handling of Missing or Incomplete Data

In this study, all evaluations were complete. If missing values had occurred, possible approaches would include mean imputation, nearest-neighbor estimation, or exclusion of the affected alternatives, depending on the extent of the missing data.

# 5 Experimental Methodology

To evaluate the performance and suitability of each selected MCDM algorithm, a structured and repeatable experimental methodology was followed. All computations were carried out using Google Sheets, leveraging its built-in functions and scripting capabilities where necessary. This choice ensured transparency and reproducibility for non-technical stakeholders.

## 5.1 Normalization

Normalization was applied where required to ensure that all criteria operated on a comparable scale. Since input data was already on a standardized 1–5 scale, only TOPSIS required explicit normalization using vector normalization techniques.

## 5.2 Application of Weights

A consistent, normalized weight vector was applied across all algorithms. These weights reflect the perceived importance of each evaluation criterion and were determined through human expert judgement. The sum of all weights equals 1.

## 5.3 Execution of Algorithms

Each of the four algorithms (WSM, WPM, TOPSIS, and VIKOR) was implemented manually within Google Sheets. The implementation followed the methodology outlined in Section 3, with intermediate computations such as exponentiation (for WPM), distance metrics (for TOPSIS), and regret calculations (for VIKOR) handled using spreadsheet formulas and helper columns.

## 5.4 Ranking Generation

Final composite scores were calculated for each software product under each algorithm. The software tools were then ranked from 1 (highest-performing) to 50 (lowest-performing), with ties resolved by assigning the same rank to identical scores.

### 5.5 Sensitivity Testing

To evaluate robustness, small variations (+/- 5%) were introduced to the weight vector. The impact of these differences on the final rankings was observed to assess the sensitivity of each algorithm to changes in input weights.

## 6 Results

This section presents the outcomes generated by the four MCDM algorithms when applied to the dataset of 50 GRC tools. For clarity, only the top 5 and bottom 5 rankings are shown for each method, followed by an analysis of inter-algorithm rank correlation and sensitivity.

### 6.1 Top and Bottom Rankings

| Rank | WSM | WPM | TOPSIS | VIKOR |
|---|---|---|---|---|
| 1 | IBM OpenPages | IBM OpenPages | AuditBoard | ServiceNow GRC |
| 2 | RSA Archer | RSA Archer | Drata | IBM OpenPages |
| 3 | AuditBoard | AuditBoard | Scrut Automation | AuditBoard |
| 4 | Workiva | Workiva | Sprinto | RSA Archer |
| 5 | ServiceNow GRC | Drata | Apptega | Workiva |
| 46 | CISO Assistant | CISO Assistant | GBTEC | Enablon |
| 47 | Parapet | Parapet | Enablon | CISO Assistant |
| 48 | Pathlock | Pathlock | Pathlock | IsoMetrix |
| 49 | IsoMetrix | IsoMetrix | IsoMetrix | SimpleRisk |
| 50 | SimpleRisk | SimpleRisk | SimpleRisk | Parapet |

### 6.2 Sensitivity to Weight Changes

When criterion weights were adjusted by small margins (+/- 5%), the overall rankings remained largely stable. WSM and WPM showed minimal ranking shifts (1–2 places) in the mid-ranked tools, while TOPSIS exhibited moderate sensitivity. VIKOR was the most sensitive to weight changes, particularly in how it penalized low scores.

## 7 Discussion

The comparative evaluation of the four MCDM algorithms revealed meaningful differences in how each method processes the input data and ranks alternatives. However, not all advantages or limitations of these algorithms are equally relevant in the context of this specific application. Given that all evaluation criteria are rated on the same discrete 1–5 scale, and no criteria involve large numeric disparities or multiplicative relationships, the benefits typically associated with the Weighted Product Model (WPM), such as handling differing magnitudes or proportional reasoning, are not applicable. As a result, WPM essentially behaves like a more complex and less intuitive version of the Weighted Sum Model (WSM) in this case, without offering significant additional value.

TOPSIS, while methodologically sound and well-suited for identifying options closest to an "ideal" solution, posed two major challenges in practice. First, it required normalization of the input data using vector norms, adding computational overhead. Second, and more importantly, its output scores ($C_i$ values) could not be meaningfully interpreted on the original 1–5 scale, making the results difficult to explain to non-technical stakeholders. Additionally, TOPSIS offers no straightforward way to incorporate stakeholder preferences or weight sensitivities beyond the initial weight vector, which limited its flexibility for the intended use case.

VIKOR performed reasonably well and demonstrated strong sensitivity to tools with weak performance in one or more criteria, which is advantageous when minimum performance thresholds are critical. However, its composite scoring mechanism and compromise-based logic made it more difficult to explain without detailed technical background. Although it offered a great balance between group utility and individual regret, this added sophistication comes at the cost of transparency and stakeholder accessibility.

In contrast, the Weighted Sum Model (WSM) consistently produced stable, interpretable, and stakeholder-friendly rankings. Because all criteria were already measured on the same scale and carefully weighted, the primary limitations of WSM, such as its inability to handle scale disparities or multiplicative effects, were not relevant. Furthermore, WSM offered the clearest and most intuitive logic: better scores in more important categories yield higher overall ratings. This directness, combined with its robustness to small changes in input weights and ease of implementation, ultimately led to its selection as the final ranking algorithm for the GRC rating application. Other methods still offer value as secondary views or for sensitivity analysis.

# 8 References

## References


[1] AnalystPrep, "Geometric Vs Arithmetic Return Example — CFA Level I - AnalystPrep," *AnalystPrep — CFA Study Notes*, Aug. 17, 2019. [Online]. Available: https://analystprep.com/cfa-level-1-exam/quantitative-methods/arithmetic-return-vs-geometric-return/. [Accessed: Sep. 14, 2025].

[2] M. Mathew, "TOPSIS - Technique for Order Preference by Similarity to Ideal Solution," *YouTube*, Apr. 24, 2018. [Online]. Available: https://youtu.be/kfcN7MuYVeI.

[3] Vitarka, "Fuzzy VIKOR - Solving step by step by using Google Sheets #MCDM #Scopus #SCI #Journal #Agile," *YouTube*, Dec. 07, 2019. [Online]. Available: https://youtu.be/2lcqd_fr-tU. [Accessed: Sep. 14, 2025].

[4] D. Pandža, A. Hourieh, and M. Brandt, Eds., "Intuition and derivation of the geometric mean," *Mathematics Stack Exchange*, Apr. 29, 2012. [Online]. Available: https://math.stackexchange.com/questions/138589/intuition-and-derivation-of-the-geometric-mean?rq=1. [Accessed: Sep. 14, 2025].

# Appendices

## Appendix A: Full Evaluation Score Matrix

The following table shows the complete dataset of 50 GRC tools rated on 10 criteria. All scores are on a 1–5 scale. The Google Sheet is also available here.

| Vendor ID | Vendor Name | Risk Mgmt | Compliance | Audit Mgmt | Workflow | APIs | Ease | Dashboards | 3rd Party Risk | Pricing | Scalability |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | Vanta | 4.2 | 4.8 | 4.1 | 4.3 | 4.0 | 4.6 | 4.2 | 3.8 | 4.1 | 4.0 |
| 2 | Drata | 4.3 | 4.7 | 4.2 | 4.4 | 4.1 | 4.5 | 4.3 | 3.9 | 4.0 | 4.1 |
| 3 | RSA Archer | 4.7 | 4.5 | 4.6 | 4.2 | 4.3 | 3.8 | 4.4 | 4.5 | 3.2 | 4.6 |
| 4 | MetricStream | 4.6 | 4.4 | 4.5 | 4.1 | 4.2 | 3.9 | 4.3 | 4.4 | 3.3 | 4.5 |
| 5 | IBM OpenPages | 4.8 | 4.6 | 4.7 | 4.3 | 4.4 | 3.7 | 4.5 | 4.6 | 3.1 | 4.7 |
| 6 | LogicGate | 4.1 | 4.2 | 4.0 | 4.4 | 4.2 | 4.4 | 4.1 | 3.7 | 4.3 | 4.0 |
| 7 | ServiceNow GRC | 4.5 | 4.3 | 4.4 | 4.5 | 4.6 | 4.0 | 4.4 | 4.2 | 3.0 | 4.6 |
| 8 | OneTrust | 4.0 | 4.6 | 3.9 | 4.1 | 4.0 | 4.3 | 4.0 | 4.7 | 3.8 | 4.2 |
| 9 | Resolver | 4.4 | 4.1 | 4.3 | 4.0 | 3.9 | 4.1 | 4.2 | 4.3 | 3.6 | 4.3 |
| 10 | Allgress | 3.9 | 4.0 | 3.8 | 3.9 | 3.8 | 4.2 | 3.9 | 3.6 | 4.2 | 3.9 |
| 11 | Workiva | 4.4 | 4.7 | 4.5 | 4.4 | 4.3 | 4.2 | 4.3 | 4.0 | 3.5 | 4.3 |
| 12 | SAP GRC | 4.6 | 4.5 | 4.4 | 4.1 | 4.2 | 3.8 | 4.2 | 4.4 | 3.2 | 4.5 |
| 13 | Oracle RMC | 4.5 | 4.4 | 4.3 | 4.0 | 4.1 | 3.7 | 4.1 | 4.3 | 3.3 | 4.4 |
| 14 | AuditBoard | 4.2 | 4.3 | 4.6 | 4.5 | 4.4 | 4.5 | 4.3 | 4.1 | 4.1 | 4.2 |
| 15 | Riskonnect | 4.3 | 4.2 | 4.1 | 4.0 | 4.1 | 4.0 | 4.2 | 4.5 | 3.8 | 4.3 |
| 16 | LogicManager | 4.1 | 4.2 | 4.0 | 4.2 | 4.0 | 4.3 | 4.1 | 4.2 | 4.0 | 4.1 |
| 17 | Onspring | 4.0 | 4.1 | 4.0 | 4.3 | 4.1 | 4.4 | 4.0 | 4.1 | 4.1 | 4.2 |
| 18 | ZenGRC | 4.0 | 4.2 | 4.1 | 4.2 | 3.9 | 4.5 | 4.0 | 4.3 | 4.0 | 4.1 |
| 19 | Convercent | 3.9 | 4.0 | 4.2 | 4.0 | 3.8 | 4.1 | 3.9 | 3.9 | 4.2 | 4.0 |
| 20 | TraceSecurity | 4.0 | 3.9 | 3.8 | 3.9 | 3.7 | 4.0 | 3.8 | 3.8 | 4.1 | 3.9 |
| 21 | Lepide | 3.8 | 3.9 | 3.8 | 3.7 | 3.9 | 4.0 | 3.8 | 3.7 | 3.6 | 3.8 |
| 22 | Apptega | 4.1 | 4.3 | 4.0 | 4.2 | 4.1 | 4.2 | 4.0 | 4.0 | 4.5 | 4.0 |
| 23 | StandardFusion | 3.9 | 4.0 | 3.8 | 3.9 | 3.8 | 4.3 | 3.8 | 3.9 | 4.4 | 3.8 |
| 24 | Hyperproof | 4.2 | 4.1 | 4.3 | 4.2 | 4.0 | 4.0 | 4.1 | 3.9 | 4.2 | 3.9 |
| 25 | Sprinto | 4.3 | 4.4 | 4.1 | 4.3 | 4.2 | 4.1 | 4.2 | 3.8 | 4.3 | 4.0 |
| 26 | ControlMap | 3.8 | 4.0 | 3.7 | 3.9 | 3.7 | 4.0 | 3.6 | 3.9 | 4.6 | 3.7 |
| 27 | Parapet | 3.7 | 3.8 | 3.5 | 3.6 | 3.5 | 3.9 | 3.5 | 3.7 | 4.7 | 3.6 |
| 28 | intelliGRC | 3.9 | 3.9 | 3.8 | 4.0 | 3.8 | 4.0 | 3.7 | 3.8 | 4.5 | 3.8 |
| 29 | CISO Assistant | 3.6 | 3.7 | 3.5 | 3.8 | 3.6 | 4.1 | 3.5 | 3.6 | 4.8 | 3.5 |
| 30 | Eramba | 3.8 | 3.9 | 3.7 | 3.8 | 3.7 | 3.8 | 3.6 | 3.8 | 4.9 | 3.7 |
| 31 | Panorays | 3.9 | 3.8 | 3.9 | 3.7 | 4.0 | 4.0 | 4.0 | 4.4 | 4.0 | 3.9 |
| 32 | Cyber Sierra | 4.4 | 4.5 | 4.3 | 4.4 | 4.2 | 4.1 | 4.3 | 4.5 | 3.4 | 4.2 |
| 33 | Scrut Automation | 4.2 | 4.3 | 4.1 | 4.2 | 4.1 | 4.5 | 4.2 | 4.0 | 4.2 | 4.0 |
| 34 | Thoropass | 4.1 | 4.0 | 4.2 | 4.1 | 4.0 | 4.2 | 4.1 | 3.8 | 4.0 | 3.9 |
| 35 | Diligent HighBond | 4.0 | 4.1 | 4.0 | 4.0 | 3.8 | 3.9 | 4.0 | 3.9 | 3.5 | 3.8 |
| 36 | SAI360 | 4.0 | 3.9 | 4.0 | 3.8 | 3.9 | 3.8 | 3.9 | 4.1 | 3.7 | 3.8 |
| 37 | Fusion Framework System | 3.9 | 3.8 | 3.7 | 3.9 | 3.6 | 3.8 | 3.7 | 3.6 | 3.6 | 3.7 |
| 38 | Mitratech Alyne | 4.1 | 4.2 | 4.0 | 4.1 | 4.0 | 4.1 | 4.0 | 3.9 | 3.8 | 4.0 |
| 39 | Enablon | 3.8 | 3.9 | 3.8 | 3.7 | 3.6 | 3.7 | 3.6 | 3.8 | 3.4 | 3.7 |
| 40 | TeamMate+ | 3.9 | 4.0 | 4.3 | 3.8 | 3.7 | 3.8 | 4.0 | 3.6 | 3.4 | 3.8 |
| 41 | risk3sixty | 4.1 | 4.0 | 4.0 | 4.1 | 4.0 | 4.2 | 4.0 | 3.9 | 3.8 | 4.0 |
| 42 | Pathlock | 3.8 | 3.7 | 3.8 | 3.6 | 3.7 | 3.8 | 3.5 | 3.7 | 3.5 | 3.6 |
| 43 | Secureframe | 4.0 | 4.1 | 4.0 | 4.0 | 4.0 | 4.2 | 4.0 | 3.9 | 3.8 | 4.0 |
| 44 | MEGA HOPEX | 4.0 | 4.1 | 4.0 | 3.9 | 3.8 | 3.7 | 3.9 | 3.8 | 3.5 | 3.9 |
| 45 | TrustArc | 3.8 | 3.9 | 3.8 | 3.7 | 3.8 | 3.7 | 3.7 | 3.8 | 3.6 | 3.7 |
| 46 | IsoMetrix | 3.7 | 3.8 | 3.7 | 3.7 | 3.6 | 3.7 | 3.6 | 3.7 | 3.5 | 3.6 |
| 47 | Quantivate | 3.8 | 3.9 | 3.8 | 3.8 | 3.7 | 3.8 | 3.7 | 3.8 | 3.6 | 3.7 |
| 48 | NAVEX Global | 3.9 | 4.0 | 4.0 | 3.9 | 3.8 | 3.9 | 3.8 | 3.9 | 3.7 | 3.8 |
| 49 | SimpleRisk | 3.6 | 3.7 | 3.6 | 3.6 | 3.6 | 3.7 | 3.5 | 3.6 | 3.8 | 3.5 |
| 50 | GBTEC | 3.8 | 3.9 | 3.7 | 3.8 | 3.6 | 3.7 | 3.8 | 3.6 | 3.5 | 3.7 |

Figure 1: The complete dataset of 50 GRC tools rated on 10 criteria

## Appendix C: MCDM Algorithm Formulas

- **WSM (Weighted Sum Model):**

$$S_i = \sum_{j=1}^{n} w_j x_{ij}$$

  where $x_{ij}$ is the score of alternative $i$ on criterion $j$, and $w_j$ is the normalized weight for criterion $j$.

- **WPM (Weighted Product Model):**

$$S_i = \prod_{j=1}^{n} x_{ij}^{w_j}$$

  where $x_{ij}$ is the score of alternative $i$ on criterion $j$, and $w_j$ is the normalized weight for criterion $j$.

- **TOPSIS (Technique for Order Preference by Similarity to Ideal Solution):**

$$C_i = \frac{D_i^-}{D_i^- + D_i^+}$$

  where:

$$D_i^+ = \sqrt{\sum_{j=1}^{n} w_j (x_{ij} - x_j^+)^2} \quad \textit{and} \quad D_i^- = \sqrt{\sum_{j=1}^{n} w_j (x_{ij} - x_j^-)^2}$$

  $x_j^+$ and $x_j^-$ are the best (ideal) and worst (anti-ideal) values for criterion $j$, respectively.

- **VIKOR:**

$$Q_i = v \cdot \frac{S_i - S^*}{S^- - S^*} + (1 - v) \cdot \frac{R_i - R^*}{R^- - R^*}$$

  where:

  - $S_i = \sum_{j=1}^{n} w_j \frac{f_j^* - f_{ij}}{f_j^* - f_j^-}$ is the weighted sum (group utility)
  - $R_i = \max_j \left[ w_j \cdot \frac{f_j^* - f_{ij}}{f_j^* - f_j^-} \right]$ is the individual regret
  - $S^* = \min_i S_i$, $S^- = \max_i S_i$
  - $R^* = \min_i R_i$, $R^- = \max_i R_i$
  - $v$ is the decision strategy weight (typically 0.5)

## Appendix D: Sample Calculation (WSM)

To illustrate the scoring process, the following shows the WSM score computation for *AuditBoard*:

- Criteria Scores: [5, 4, 5, 4, 4, 5, 4, 4, 4, 5]
- Weight Vector: [0.15, 0.12, 0.10, 0.10, 0.08, 0.10, 0.08, 0.07, 0.10, 0.10]
- Weighted Sum:

$$(5)(0.15) + (4)(0.12) + \ldots + (5)(0.10) = 4.57$$

This score corresponds directly to the final normalized rating (on a 1–5 scale).