Article

Evaluation Metrics in Learning Systems: A Survey

Anahita Nouri , Baqer M. Merzah , Sahand Mosayyebpour , Ramin Mousa , Saba Hesaraki *

Posted Date: 21 August 2025

doi: 10.20944/preprints202508.1594.v1

Keywords: evaluation metrics; text mining metrics; clustering metrics; image mining metrics

Article

Evaluation Metrics in Learning Systems: A Survey

Anahita Nouri [1], Baqer M. Merzah [2], Sahand Mosayyebpour [3], Ramin Mousa [4] and Saba Hesaraki [5,*]

[1] Masters Degree in Artificial Intelligence, University of Verona, Italy
[2] Department of Computer Science, Faculty of Education, University of Kufa, Najaf, Iraq
[3] Department of Computer and Electrical Engineering, University of Victoria, Victoria, BC, Canada
[4] Department of Computer Engi3neering, University of Zanjan, University Blvd., 45371-38791, Zanjan, I. R. Iran, Zanjan, 45371-38791, Iran
[5] Department of Computer, Mechanics and Electrical Engineering, Islamic Azad University Science and Research Branch, Tehran, Iran
* Correspondence: saba.hesaraki@iau.ir

Abstract

Learning systems have played a significant role in classification, Learning systems have shaped classification, clustering, regression, and pattern learning problems within the past few decades. Machine learning and deep learning continue to churn out newer models that have transformed the areas of natural language processing, computer vision, speech recognition, recommender systems, among others. The evaluation metrics serve as verification systems and are a tool to benchmark the models presented. Depending on the field of use and kind of application, different metrics have been put forth in the literature. They form the common language between researchers. Commercial projects use these metrics to help companies evaluate the value of a model from both an economic and practical point of view. Moreover, in the absence of standards for measurement and evaluation criteria, it becomes almost impossible to compare the results obtained from similar models. This study aims to classify different metrics in learning systems. Therefore, approximately 250 evaluation metrics in text mining, clustering, image mining, and signal analysis have been presented.

Keywords: evaluation metrics; text mining metrics; clustering metrics; image mining metrics

1. Introduction

Machine Learning and Deep Learning, having become the twin pillars of artificial intelligence, have attracted attention in recent decades owing to their ability to solve complex problems and automate solutions in many fields, including Natural Language Processing [1], Computer Vision [2], Speech Recognition [3], Recommender Systems [4],medicine [5,6] to signal processing [7], video processing [8], natural language processing [9], large language modeling [10], virtual reality [11], time-series analysis [12–14], and educational systems [15]. However, with the development and tuning of a machine learning model, it extends beyond the design of algorithms and training of the algorithm itself; not only are these important steps, but the accurate and systematic evaluative assessment of the performance of these models is, in fact, one of the key important steps in the development process. Evaluation metrics assist model builders in measuring quantifiably and qualitatively the strengths and weaknesses of a model [16]. These metrics have become vital to secure the reliability, accuracy, and efficiency of a model's differential applications in the real world [17]. A model may perform very well in a hypothetical world; however, assuming no accurate metric is at hand, one cannot confidently claim that it will also do well in the real world or when confronted with newer data. Evaluation metrics allow the researchers to investigate the model from various perspectives, such as its generalization property, resistance to overfitting, and behaviour on unbalanced or noisy data. These are important attributes for a developer to consider while making design decisions that affect the improvement, learning procedure, or even choice compared to a more exemplary architecture. Adjusting parameters, providing rewards in training data-choice options to weigh their values, can substantially affect the final performance.

In addition, evaluation metrics serve as a common language among researchers [18]. In commercial projects, these metrics help businesses evaluate the value of a model from an economic and practical perspective. For example, for an image-based medical diagnosis system, the degree of correctness in disease diagnosis or False Positive Rate may directly influence the lives of patients. For text processing applications such as machine translation systems, a metric like BLEU or ROUGE may determine the quality of translation or summarization. Therefore, the choice of the right metric depends not only on the type of problem, but also on the final application of using the model. The evaluation measures used are very diverse depending on the type of problem, i.e., classification, regression, clustering, content creation, etc., and the domain of use, i.e., text, image, audio, etc. There are numerous issues in text processing, including text classification (opinion analysis, sentiment analysis, spam classification), text generation, text summarization, and machine translation. Accuracy, Precision, Recall, and F1-score are prevalent metrics for classification problems. BLEU for translation, ROUGE for summarization, or Perplexity for language models are standard metrics. These metrics help to evaluate the quality of the model's responses in terms of semantics, structure, and compliance with human natural language. In image-related problems, such as object recognition, segmentation, and image classification, metrics such as Accuracy, Intersection over Union (IoU), Mean Average Precision (mAP), and Top-k Accuracy are very widely used. Likewise, for speech processing, there are metrics such as Word Error Rate (WER), which are used to evaluate speech-to-text systems. For recommender systems, one uses a metric such as Mean Reciprocal Rank (MRR) to measure the quality of the recommendation ranking. Each of these metrics provides various information about the model's performance, depending on the type of problem and the application objective. One of the main problems with the use of evaluation metrics is the choice of a metric that most accurately reflects the project goal [19]. In an unbalanced data classification problem, accuracy is misleading on its own, since the model can predict the majority class and perform well, but perform badly in the detection of the minority class. In such cases, metrics such as the F1-score or Area Under the ROC Curve (AUC-ROC) provide more accurate information. Specific metrics are also not calibrated to the end-user's goals. For example, for a machine translation system, a high BLEU score may indicate structure similarity to the reference text, but does not guarantee the semantic quality of the translation. For this reason, human evaluation is sometimes required alongside automated metrics.

This work aims to provide a comprehensive review of learning system evaluation metrics. Machine learning evaluation metrics are not only a mechanism to measure the performance of models, but also a recipe to improve, compare, and select the optimal models. By providing quantitative and qualitative data, these metrics help developers and researchers construct effective, reliable, and feasible models. Since machine learning problems and domains are different, the use of the right metric is dependent upon a deep knowledge of problem, data, and project goals.

2. Text Mining

Text mining is a nascent and growing field that tries to extract meaningful information from unstructured and structured natural texts. Text Mining converts unstructured text into a structured format to identify meaningful patterns and new insights [20]. By using advanced analytical techniques, such as machine learning algorithms and deep learning algorithms, it is possible to discover hidden relationships in unstructured data. Text is one of the most common data types in a database. These data can be divided into three categories [21]:

1. Structured Data: These data are standardized as tables of multiple rows and columns. This makes them easier to store and process. Structured data can include entries such as name, address, and phone number.
2. Unstructured Data: These data do not have a predefined and specific format. This data can include text from social media, product reviews, or video and audio files.

3. Semi-structured Data: As the name suggests, this data combines structured and unstructured data. Examples of semi-structured data include XML, JSON, and HTML files.

Since nearly 80% of the organizational data in the world is unstructured, text mining is considered a very valuable practice in organizations [22]. Text mining tools and techniques allow us to convert unstructured documents into a structured format for high-quality analysis and insight. This, in turn, improves organizational decision-making and leads to better business outcomes [23]. Text mining includes various applications such as marketing applications, Enabling better CRM [24], security applications [25], deception detection [26], medicine and biology [27], literature-based gene identification [28], academic applications, and research stream analysis. It also includes text mining in the natural language processing of users, such as Named-entity recognition [29], Question answering [30], Automatic summarization [31], Natural language generation and understanding [32], Machine translation [33], Foreign language reading and writing [34], Speech recognition, and text proofing [35], and optical character recognition [36]. Different evaluation criteria depending on the user in text mining are defined in the literature. In this section, an attempt has been made to introduce the most critical evaluation criteria in each field.

2.1. Text Data Augmentation

Recent research on data augmentation for natural language processing, including GAN neural networks, has provided various technological advances. Frameworks, tools, and implementation of text-enhanced data pipelines are promoted. However, the lack of evaluation criteria and methods comparison standards, for various reasons including data and criteria, has made the comparison meaningless. Also, there is a lack of standardization of methods, and textual data enhancement research uses unified criteria to compare different enhancement methods. Therefore, researchers seek relevant evaluation criteria for textual data augmentation techniques. Continuing this section, we have tried to categorize the most important criteria for increasing data for natural language processing [37].

2.1.1. Machine Translation Quality

Human evaluation is the best option for evaluating translation quality in intelligent systems and is considered the ground truth. However, since human evaluation is time-consuming and expensive, it is only possible to evaluate some new output of the model by a group of people. For this purpose, a group of evaluation metrics have been introduced that make it possible to evaluate the models to some extent. In the following, we discuss some of the most important metrics [37].

BLEU Score

The **BLEU** (Bilingual Evaluation Understudy) score is an automated metric for evaluating the quality of text generated by machine translation (MT) systems by comparing it to one or more reference translations [38]. BLEU assesses how closely a machine-generated translation matches professional human translations, focusing on both accuracy and fluency.

Definition 1. *Mathematically, the BLEU score is defined as:*

$$BLEU = BP \times \exp\left(\sum_{n=1}^{N} w_n \log p_n\right) \tag{1}$$

where:

- *N is the maximum n-gram length considered (commonly $N = 4$).*
- *w_n is the weight assigned to the precision score of n-grams of order n (typically uniform weights, $w_n = \frac{1}{N}$).*
- *p_n is the modified precision for n-grams of order n.*
- *BP is the brevity penalty that penalizes overly short translations.*

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: Posted: 21 August 2025 doi:10.20944/preprints202508.1594.v1

Modified n-gram Precision (p_n) The modified n-gram precision p_n evaluates translation quality across a multi-sentence test corpus. For each source sentence, which may have multiple reference translations, we compute n-gram matches against all its references. The final precision is calculated by aggregating counts across all sentences in the corpus.

$$p_n = \frac{\sum_{C \in \{\text{Candidates}\}} \sum_{\text{n-gram} \in C} \text{Count}_{\text{clip}}(\text{n-gram})}{\sum_{C' \in \{\text{Candidates}\}} \sum_{\text{n-gram}' \in C'} \text{Count}(\text{n-gram}')} \tag{2}$$

where:

- $\{\text{Candidates}\}$ represents the set of all candidate translations in the test corpus
- $\text{Count}_{\text{clip}}(\text{n-gram})$ is computed as:

$$\text{Count}_{\text{clip}}(\text{n-gram}) = \min(\text{Count}_C(\text{n-gram}), \max_{R \in \text{Refs}(C)} \text{Count}_R(\text{n-gram})) \tag{3}$$

where $\text{Refs}(C)$ are the reference translations for the candidate C

- $\text{Count}(\text{n-gram}')$ counts each n-gram occurrence in the candidate translation

Brevity Penalty (BP) The brevity penalty penalizes candidate translations that are shorter than the reference translations to prevent overly concise translations that achieve high precision by omitting necessary content. It is calculated as follows:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

where:

- c is the total length (in words) of the candidate translations.
- r is the effective reference length, defined as the sum of the lengths of the reference translations closest to each candidate sentence's length.

Logarithmic Formulation The BLEU score can also be expressed in the logarithmic domain for clarity:

$$\log(\text{BLEU}) = \min\left(1 - \frac{r}{c}, 0\right) + \sum_{n=1}^{N} w_n \log p_n. \tag{4}$$

Note: BLEU scores range from 0 to 1, where a higher score indicates greater similarity to the reference translations. In practice, BLEU scores are often multiplied by 100 to represent them as percentages.

2.1.2. Text Generation Quality

Text generation is the task of producing text similar to human-generated texts with the goal of being indistinguishable by humans. This task is formally known in the literature as "natural language generation".

Text generation can be investigated with Markov processes or deep generative models such as GANs. The continuation of the most important metrics for evaluating the quality of text generation is given.

- **Novelty:**Novelty and Diversity are distinct. Novelty covers the difference of the produced text from the training set, while Diversity covers the difference of the produced text from other produced texts. A model can be diverse but not novel. These two criteria are listed below [39]:

$$Novelty(S_i) = 1 - max\{\varphi(S_i, C_j)\}_{j=1}^{j=|C|} \tag{5}$$

- **Diversity:** BLEU [38] and Self-BLEU [40] are common metrics for quality and diversity evaluation, respectively:
$$E_{C\sim Q}\text{NSBLEU}(C) = -\sum_{i=1}^{N} Q_i\left(1-(1-Q_i)^{|C|-1}\right) \tag{6}$$
Calculate BLEU score for every generated sentence, and define the average BLEU score to be the Self-BLEU of the document.
Discriminative BLEU [41]: For tasks with inherently diverse outputs, such as conversational response generation, where there is a wide range of plausible and acceptable responses, ΔBLEU is introduced as an enhanced version of the traditional BLEU score. Unlike standard BLEU, ΔBLEU incorporates human judgments about the quality of reference texts, making it more discriminative. It rewards matches with high-quality references and penalizes matches with low-quality ones, thereby better aligning with human evaluations. This makes ΔBLEU particularly suitable for tasks like conversational response generation, where the space of valid outputs is vast and subjective. The ΔBLEU score for a corpus is computed as:
$$\Delta\text{BLEU} = \text{BP}\cdot\exp\left(\sum_{n=1}^{N}\log p_n\right)$$
where BP is the brevity penalty, and p_n is the modified n-gram precision for n-grams of order n. The modified n-gram precision p_n is defined as:
$$p_n = \frac{\sum_i \sum_{g\in\text{n-grams}(h_i)} \max_{j:g\in r_{i,j}} w_{i,j}\cdot \#_g(h_i, r_{i,j})}{\sum_i \sum_{g\in\text{n-grams}(h_i)} \max_j w_{i,j}\cdot \#_g(h_i)}$$
where:
  - i is index over the test set, and j is index over the set of references $r_{i,j}$ for the i-th input message.
  - $\#_g(h_i, r_{i,j})$ is the minimum number of occurrences of n-gram g in the hypothesis h_i and reference $r_{i,j}$.
  - $w_{i,j}$ is the human-assigned weight for reference $r_{i,j}$.
- **Fluency:** This is a subjective and multidimensional concept that can be related to various criteria, such as readability, clarity, and naturalness of the produced text. The context, purpose and audience of the text production task influence fluency. For example, a fluent text for a news article may differ from a fluent text for a children's story. The text production fluency depends on the desired program. In technical writing, clarity and precision can be essential. The following evaluation criteria can be used to evaluate fluency [42].
SLOR:
The Syntactic Log Odds Ratio (SLOR) is a metric designed for the evaluation of fluency in natural language generation (NLG) outputs. Its primary purpose is to assess the fluency of generated sentences without relying on reference sentences, which can be particularly useful in scenarios where references are unavailable or impractical to obtain. The SLOR metric is defined mathematically as follows [43]:
$$\text{SLOR}(S) = \frac{1}{|S|}(\ln p_M(S) - \ln p_u(S)) \tag{7}$$
In this equation, S represents the sentence being evaluated, $p_M(S)$ is the probability assigned to the sentence by a language model (LM), and $p_u(S)$ is the unigram probability of the sentence, calculated as the product of the probabilities of each individual token in the sentence. The normalization by $|S|$, the length of the sentence, ensures that the metric accounts for variations in sentence length, allowing for a fair comparison between sentences of different sizes. By leveraging the probabilities from a trained language model, SLOR provides a more nuanced evaluation

of fluency, making it a valuable tool for researchers and practitioners in the field of natural language processing.

Perplexity(PPL),wang2022perplexity: Perplexity (PPL) is one of the most common criteria for evaluating language models. PPL is a metric used to evaluate the fluency and quality of generated text by language models. It is derived from cross-entropy. Mathematically, the cross-entropy H for a sequence of tokens $t_1, t_2, \ldots, t_m$ can be expressed as:

$$H = -\sum_{i=1}^{m} \log P(t_i \mid t_1, t_2, \ldots, t_{i-1}),$$

where $P(t_i \mid t_1, t_2, \ldots, t_{i-1})$ is the predicted probability of the i-th token given the previous tokens. PPL is then defined as the exponentiation of the average cross-entropy:

$$PPL(s) = \exp(\frac{H}{m}) = \exp\left(-\frac{1}{m}\sum_{i=1}^{m} \log P(t_i \mid t_1, t_2, \ldots, t_{i-1})\right), \tag{8}$$

where s is the input sentence and m is the number of tokens in the sequence. A lower PPL value indicates better fluency and quality of the text generated by the model.

- **Sentiment Consistency**: In the context of text generation, maintaining sentiment consistency is crucial for ensuring that the generated text aligns with the expected emotional tone of the input prompts. In [45], two primary metrics are introduced to evaluate sentiment consistency in generated continuations: SENTSTD and SENTDIFF. These metrics help assess how well the sentiment of the generated text corresponds to the ground truth.

  1. **SENTSTD (Sentiment Standard Deviation),feng2020genaug**: This metric measures the average standard deviation of sentiment scores for each batch of generated continuations, concatenated with their respective input prompts. By calculating the standard deviation across all test examples, a lower SENTSTD value indicates greater consistency in sentiment among the generated continuations. This suggests that the generated texts exhibit a more stable emotional tone.
  2. **SENTDIFF (Sentiment Difference),feng2020genaug**: This metric evaluates the average difference in sentiment scores between each batch of continuations and their corresponding ground-truth reviews. By averaging these differences across all test examples, a lower SENTDIFF value indicates that the sentiment of the generated continuations aligns more closely with the ground-truth reviews. This metric is essential for assessing the fidelity of the generated text to the expected sentiment.

- **Semantic content preservation,zhang2019bertscore**: This evaluation metric is designed to assess the semantic equivalence between generated text and reference text. Unlike traditional metrics that rely on surface form similarity, this metric leverages contextual embeddings to evaluate how well the meaning of the content is preserved in the generated output. The BERTSCORE metric computes the similarity between tokens in the candidate sentence and those in the reference sentence using cosine similarity of their contextual embeddings. The formula for computing the recall, precision, and F1 scores is given as follows:

$$R_{BERT} = \frac{1}{|x|} \sum_{x_i \in x} \max_{x_j \in \hat{x}} \text{cosine}(e(x_i), e(\hat{x}_j)) \tag{9}$$

$$P_{BERT} = \frac{1}{|\hat{x}|} \sum_{\hat{x}_j \in \hat{x}} \max_{x_i \in x} \text{cosine}(e(x_i), e(\hat{x}_j)) \tag{10}$$

$$F_{BERT} = 2 \cdot \frac{P_{BERT} \cdot R_{BERT}}{P_{BERT} + R_{BERT}} \tag{11}$$

where:

- x is the reference sentence tokenized into k tokens $\{x_1, x_2, \ldots, x_k\}$.
- $\hat{x}$ is the candidate sentence tokenized into l tokens $\{\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_l\}$.
- $e(x_i)$ and $e(\hat{x}_j)$ represent the contextual embeddings of tokens x_i and $\hat{x}_j$, respectively.
- cosine(a, b) is the cosine similarity between vectors a and b.

This approach allows for a more nuanced evaluation of text generation, focusing on the preservation of meaning rather than mere lexical overlap, thereby addressing common pitfalls in traditional n-gram based metrics. To enhance this evaluation, Inverse Document Frequency (IDF) weighting is incorporated, which emphasizes the importance of rare and informative tokens. The Recall metric with IDF weighting is defined as follows:

$$R_{BERT} = \frac{\sum_{x_i \in x} \text{idf}(x_i) \cdot \max_{x_j \in \hat{x}} \text{cosine}(e(x_i), e(\hat{x}_j))}{\sum_{x_i \in x} \text{idf}(x_i)} \tag{12}$$

where:

- $\text{idf}(x_i) = \log\left(\frac{M}{n(x_i)}\right)$ is the IDF score of token x_i, where M is the total number of reference sentences, and $n(x_i)$ is the number of sentences containing x_i.

2.1.3. Character n-gram matches

- **Character n-gram F-score (CHRF),popovic2017chrf++**: This evaluation metric assesses the quality of machine translation outputs by leveraging character n-grams. CHRF correlates well with human assessments, particularly for morphologically rich languages. It calculates precision and recall based on character n-grams, making it language-independent and robust against tokenization issues. The general formula for the n-gram based F-score is given by:

$$\text{F}_{\text{CHRF}}(\beta) = \frac{(1+\beta^2) \cdot \text{P}_{\text{CHRF}} \cdot \text{R}_{\text{CHRF}}}{\beta^2 \cdot \text{P}_{\text{CHRF}} + \text{R}_{\text{CHRF}}} \tag{13}$$

where:

  - P_{CHRF} is the character n-gram precision, calculated as the average precision across all n-grams from $n = 1$ to N.
  - R_{CHRF} is the character n-gram recall, representing the average recall across all n-grams from $n = 1$ to N.
  - β is a parameter that assigns β times more weight to recall than to precision, with a typical value of 2 being optimal.

  The CHRF score is computed by averaging the precision and recall across n-grams of lengths from 1 to N, where N is typically set to 6 for character n-grams.
- **CHRF+**: CHRF+ enhances the original CHRF metric by incorporating word unigrams alongside character n-grams. This integration aims to improve correlation with direct human assessments by capturing both surface-level and semantic information. The precision and recall for CHRF+ ($\text{P}_{\text{CHRF+}}$ and $\text{R}_{\text{CHRF+}}$) are calculated as the combined metrics of both character and word n-grams, averaged across all n-gram lengths.
- **CHRF++**: CHRF++ further refines the CHRF+ metric by optimizing the combination of character and word n-grams, particularly by incorporating word unigrams and bigrams. This variant enhances correlation with direct human assessments by capturing more contextual information from the text. The precision and recall for CHRF++ ($\text{P}_{\text{CHRF++}}$ and $\text{R}_{\text{CHRF++}}$) are computed with enhanced weighting strategies for character and word n-grams, allowing for a more nuanced evaluation.

2.1.4. Prediction quality for classification

In the realm of machine learning and statistical analysis, the evaluation of binary classification models is crucial for understanding their performance and reliability. Various metrics have been developed to summarize the outcomes of these models, each offering unique insights into their predictive capabilities. This section focuses on several key metrics that are commonly employed to assess the quality of predictions in binary classification tasks.

The evaluation of a model's performance typically begins with the construction of a confusion matrix, which categorizes the predictions into four distinct groups: true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). These categories form the basis for calculating various performance metrics, which help researchers and practitioners gauge how well their models are performing.

Among the most widely used metrics are Accuracy, Precision, and Recall, each of which provides different perspectives on the model's effectiveness. Accuracy offers a general measure of correctness, while Precision and Recall focus on the model's performance concerning the positive class. Additionally, the Macro-F1 Score serves as a harmonic mean of Precision and Recall, providing a balanced evaluation, especially in scenarios with imbalanced class distributions.

Other important metrics include Jaccard Similarity, which quantifies the similarity between predicted and actual sets, and the Matthews Correlation Coefficient (MCC), which is particularly valuable for assessing performance on imbalanced datasets. Finally, the Area Under the Precision-Recall Curve (AUPRC), calculated using the trapezoidal rule, summarizes the model's performance across various thresholds [48].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{14}$$

where: TP = True Positives, TN = True Negatives, FP = False Positives, and FN = False Negatives.

$$\text{Precision} = \frac{TP}{\text{Actual Results}} = \frac{TP}{TP + FP} \tag{15}$$

Precision measures the accuracy of positive predictions.

$$\text{Recall} = \frac{TP}{\text{Predicted Results}} = \frac{TP}{TP + FN} \tag{16}$$

Recall measures the ability of a model to find all the relevant cases (true positives).

$$\text{Macro-F1 Score} = 2 \cdot \frac{\text{Total Precision} \cdot \text{Total Recall}}{\text{Total Precision} + \text{Total Recall}} = \frac{\sum_{i=1}^{n} F1_{\text{score}_i}}{n} \tag{17}$$

Explanation: The Macro-F1 Score is calculated as the harmonic mean of total precision and total recall. It averages the F1 scores for each class, treating all classes equally regardless of their support (number of true instances).

$$\text{Jaccard Similarity}(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \tag{18}$$

Explanation: The Jaccard Similarity measures the similarity between two sets A and B. It is defined as the size of the intersection divided by the size of the union of the sets, providing a value between 0 (no overlap) and 1 (complete overlap).

$$\text{MCC} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{19}$$

Explanation: The Matthews Correlation Coefficient (MCC) is a measure of the quality of binary classifications. It considers all four categories of the confusion matrix (True Positives, True Negatives, False Positives, False Negatives) and provides a balanced metric that can be used even when classes are of very different sizes.

$$\text{AUPRC}_{\text{trapezoid}} = \frac{1}{2}\sum_{i=1}^{n}(r_{i+1} - r_i)(p_{i+1} + p_i) \tag{20}$$

Explanation: The Area Under the Precision-Recall Curve (AUPRC) calculated using the trapezoidal rule. Here, r represents the recall and p represents the precision at different thresholds. This metric summarizes the model's performance across various levels of classification thresholds.

Confusion matrix: Confusion matrix is an $N * N$ matrix used to evaluate the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with the values predicted by the machine learning model. The main components of this matrix are shown in Figure 1, which can be expanded for more classes [49].Confusion matrices are the main tool for evaluating errors in classification problems. These graphical models encode the full specification of misclassifications. Confusion matrices are used for the following:

- Error inspection of each class.
- Setting software parameters such as detection threshold.
- Comparing software versions [50].

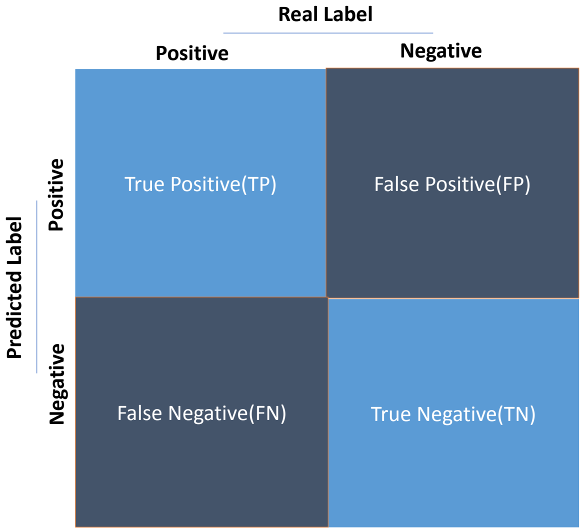

Figure 1. Visualization of Confusion Matrix

2.2. Dataset Correlations

How different datasets relate to each other can be examined by employing correlation and similarity measures. Specifically, we explore Spearman's rank correlation coefficient to capture monotonic relationships in ordinal data and Cosine similarity to measure the orientation of vectors in high-dimensional space. These metrics provide insights into the strength, direction, and nature of the relationships among the datasets.

Spearman's rank correlation coefficient: Spearman's correlation coefficient (r_s) is a non-parametric measure used to assess the strength and direction of the association between two ranked variables. It is particularly valuable when dealing with ordinal data. Unlike Pearson's correlation, which evaluates linear relationships, Spearman's correlation focuses on monotonic relationships, making it suitable for a wider range of data types. The coefficient ranges from -1 to +1, where -1 indicates a perfect negative correlation, +1 indicates a perfect positive correlation, and 0 indicates no correlation. Spearman's rank correlation coefficient is obtained by the following formula [51]:

$$r_s = 1 - 6\sum \frac{d^2}{n(n^2 - 1)}, \tag{21}$$

where

- d_i: The difference between the ranks of the two variables for each observation.

- n: The number of paired observations.

Cosine similarity: This metric is a widely used in machine learning and pattern recognition, particularly for measuring the similarity between high-dimensional vectors. It quantifies the cosine of the angle between two vectors, providing a measure of their orientation rather than their magnitude. This characteristic makes cosine similarity particularly useful in applications such as text analysis and image recognition, where the direction of the data points is often more significant than their magnitude.

The cosine similarity between two vectors $\vec{a}$ and $\vec{b}$ is defined mathematically as:

$$\cos\theta = \frac{\vec{a}\cdot\vec{b}}{||\vec{a}||\,||\vec{b}||} \tag{22}$$

where:

- $\vec{a}\cdot\vec{b}$ is the dot product of the vectors $\vec{a}$ and $\vec{b}$.
- $||\vec{a}||$ is the Euclidean norm (magnitude) of vector $\vec{a}$, calculated as $\sqrt{\sum_{i=1}^{d} a_i^2}$.
- $||\vec{b}||$ is the Euclidean norm (magnitude) of vector $\vec{b}$, calculated as $\sqrt{\sum_{i=1}^{d} b_i^2}$.
- θ is the angle between the two vectors.

The resulting value of cosine similarity ranges from -1 to 1, where 1 indicates that the vectors are identical in orientation, 0 indicates orthogonality (no similarity), and -1 indicates that the vectors are diametrically opposed. [52]:

2.3. *Automatic Speech Recognition (ASR) Performance*

Automatic Speech Recognition (ASR) is a technology that enables computers to recognize and process human speech.

Word Error Rate (WER): WER quantifies the accuracy of the transcribed output by comparing it to a reference transcription. The word sequence hypothesized by the ASR system is aligned with the reference transcription, and the number of errors is computed as the sum of three types of errors: Substitutions (S): Words that are incorrectly recognized, Insertions (I): Extra words added to the ASR output that do not appear in the reference, and deletions (D): Words present in the reference transcription but missing from the ASR output [53]:

$$\text{WER} = \frac{I + D + S}{N} \times 100 \tag{23}$$

Word Recognition Rate (WRR): Word Recognition Rate (WRR) is a metric used to evaluate the performance of Automatic Speech Recognition (ASR) systems. It measures the proportion of words correctly recognized by the ASR system compared to the total number of words in the reference text (ground truth). [54]:

$$WRR = 1 - WER = \frac{N - S - D - I}{N} = \frac{H - I}{N}, \tag{24}$$

where $H = N - (S + D)$ represents Number of correctly recognized words.

2.4. *Training*

In this category, various metrics that evaluate the performance of machine learning algorithms are presented. The most important learning evaluation metrics include:

Error Rate: The error rate is defined as the complement of accuracy, providing a measure of the proportion of incorrect predictions made by the model. It is expressed mathematically as:

$$\text{Error} = 1 - \text{Accuracy} \tag{25}$$

Standard Binary Cross-Entropy Loss Function: The binary cross-entropy (BCE) loss function is utilized for binary classification tasks. It quantifies the difference between the predicted probabilities and the actual binary labels. The formula is given by [55]:

$$J_{bce} = -\frac{1}{M}\sum_{m=1}^{M}[y_m \log(h_\theta(x_m)) + (1 - y_m)\log(1 - h_\theta(x_m))] \tag{26}$$

where M is the number of training samples, y_m is the target label for training sample m (0 or 1), x_m is the input for training sample m, $h_\theta(x_m)$ is the predicted probability that the label is 1 for input x_m.

Weighted Binary Cross-Entropy: The weighted binary cross-entropy loss function adjusts the standard BCE to account for class imbalance by applying a weight to the positive class. The formula is [55]:

$$J_{wbce} = -\frac{1}{M}\sum_{m=1}^{M}[w \cdot y_m \log(h_\theta(x_m)) + (1 - y_m)\log(1 - h_\theta(x_m))] \tag{27}$$

where w is the weight assigned to the positive class.

Standard Categorical Cross-Entropy: For single-label multiclass classification, the standard categorical cross-entropy loss function is expressed as [55]:

$$J_{cce} = -\frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M} y_m^k \log(h_\theta(x_m, k)) \tag{28}$$

where K is the number of classes, y_m^k is the target label for training example m for class k, $h_\theta(x_m, k)$ is the predicted probability for class k.

Standard Weighted Categorical Cross-Entropy: The weighted categorical cross-entropy loss function extends the standard categorical cross-entropy by incorporating class weights. It is defined as [55]:

$$J_{wcce} = -\frac{1}{M}\sum_{k=1}^{K}\sum_{m=1}^{M} w_k \cdot y_m^k \log(h_\theta(x_m, k)) \tag{29}$$

where w_k is the weight for class k.

Real-World Weight Cross-Entropy (RWWCE): The RWWCE loss function is designed to account for the real-world costs associated with misclassification. It incorporates weights for false negatives and false positives, expressed as [55]:

$$J_{brwwce} = -\frac{1}{M}\sum_{m=1}^{M}\left[w_{mcfn} \cdot y_m \log(h_\theta(x_m)) + w_{mcfp} \cdot (1 - y_m)\log(1 - h_\theta(x_m))\right] \tag{30}$$

where w_{mcfn} is the marginal cost of a false negative over a true positive, w_{mcfp} is the marginal cost of a false positive over a true negative.

These metrics provide a comprehensive framework for evaluating the performance of classifiers in machine learning, particularly in scenarios involving imbalanced classes and significant costs associated with mislabeling.

2.5. Discriminator Metrics

Discriminator metrics are primarily used to evaluate and select the optimal solution in classification tasks. They play a crucial role in identifying the best model that can accurately predict future evaluations of a classifier [56]. Traditional metrics, such as accuracy, are widely used due to their simplicity and ease of interpretation. However, as highlighted in [56], accuracy often fails to provide sufficient discrimination power, particularly in imbalanced datasets where it tends to favor the majority class. To address these limitations, hybrid metrics such as Optimized Precision (OP) and Optimized Accuracy with Recall and Precision (OARP) have been proposed. In addition to hybrid metrics, ranking-based metrics such as Area Under the Curve (AUC) offer a holistic view of classifier performance. Table1 summarizes these metrics, providing their formulas and highlighting their evaluation focuses.

Table 1. Evaluation Metrics for Classification: Including Threshold, Average, and Hybrid Metrics [56]

Metrics	Formula	Evaluation Focus
Accuracy (acc)	$\text{acc} = \frac{tp+tn}{tp+tn+fp+fn}$	Ratio of correct predictions over total instances.
Error Rate (err)	$\text{err} = \frac{fp+fn}{tp+tn+fp+fn}$	Ratio of incorrect predictions over total instances.
Sensitivity (sn)	$\text{sn} = \frac{tp}{tp+fn}$	Fraction of actual positives correctly classified.
Specificity (sp)	$\text{sp} = \frac{tn}{tn+fp}$	Fraction of actual negatives correctly classified.
Precision (p)	$p = \frac{tp}{tp+fp}$	Positive patterns correctly predicted from total predicted positives.
Recall (r)	$r = \frac{tp}{tp+fn}$	Fraction of positive patterns correctly classified.
F Measure (FM)	$FM = 2 \cdot \frac{p \cdot r}{p+r}$	Harmonic mean of precision and recall.
Geometric Mean (GM)	$GM = \sqrt{tp \cdot tn}$	Balances true positive and true negative rates.
Mean Square Error (MSE)	$\text{MSE} = \frac{1}{n} \sum_{j=1}^{n} (P_j - A_j)^2$	Measures difference between predicted and actual values.
AUC	$AUC = \frac{1}{n_p n_n} \sum_{i=1}^{n_p} \sum_{j=1}^{n_n} I(f(x_i) > f(x_j))$	Overall ranking performance of a classifier.
Optimized Precision (OP)	$OP = \text{acc} + \text{sn} + \text{sp}$	Hybrid metric combining accuracy, sensitivity, and specificity.
OARP	$OARP = \text{acc} + r + p$	Hybrid metric designed for classifier training.
Averaged Accuracy	$\text{Averaged Acc} = \frac{1}{C} \sum_{c=1}^{C} \text{Accuracy}_c$	Average effectiveness across all classes.
Averaged Error Rate	$\text{Averaged Err} = \frac{1}{C} \sum_{c=1}^{C} \text{Error Rate}_c$	Average error rate across all classes.
Averaged Precision	$\text{Averaged Prec} = \frac{1}{C} \sum_{c=1}^{C} \text{Precision}_c$	Average precision across all classes.
Averaged Recall	$\text{Averaged Rec} = \frac{1}{C} \sum_{c=1}^{C} \text{Recall}_c$	Average recall across all classes.
Averaged F Measure	$\text{Averaged FM} = \frac{1}{C} \sum_{c=1}^{C} FM_c$	Average F1 score across all classes.

2.6. *Agreement Metrics*

Agreement metrics are essential tools for evaluating the performance of classification models, particularly in clinical diagnostics and machine learning tasks. These metrics assess the level of agreement between predicted and actual outcomes, accounting for chance agreement. A 2x2 contingency table is often used to define these metrics, where a, b, c, and d represent the counts of true positives, false positives, false negatives, and true negatives, respectively. The total number of observations is given by $n = a + b + c + d$.

Raw agreement (I) measures the proportion of correctly classified instances ($I = \frac{a+d}{n}$), while "Agreement Due to Chance" ($\hat{I}$) estimates the proportion of agreement expected by random chance. The value of $\hat{I}$ is computed as:

$$\hat{I} = \frac{(a+c)(a+b) + (b+d)(c+d)}{n^2}.$$

This value is crucial because it provides a baseline for evaluating whether the observed agreement is better than random guessing. Cohen's kappa (κ) incorporates $\hat{I}$ and is defined as:

$$\kappa = \frac{I - \hat{I}}{1 - \hat{I}}.$$

Kappa thus quantifies the agreement between predictions and actual outcomes beyond what is expected by chance. A kappa score between 0.81 and 0.99 indicates near-perfect agreement [57,58].

Sensitivity ($Se = \frac{a}{a+c}$) and specificity ($Sp = \frac{d}{b+d}$) further evaluate the model's ability to correctly classify positive and negative cases, respectively. These metrics, along with kappa, provide a comprehensive framework for assessing model performance in both clinical and machine learning applications. Table 2 summarizes these metrics and their definitions as discussed in [57,58].

Table 2. Agreement Metrics and Their Definitions

Metric	Definition
Sensitivity (Se)	Proportion of true positives correctly identified: $Se = \frac{a}{a+c}$
Specificity (Sp)	Proportion of true negatives correctly identified: $Sp = \frac{d}{b+d}$
Raw Agreement (I)	Proportion of total agreement: $I = \frac{a+d}{n}$
Agreement Due to Chance ($\hat{I}$)	Expected agreement by chance: $\hat{I} = \frac{(a+c)(a+b)+(b+d)(c+d)}{n^2}$
Cohen's Kappa (κ)	Agreement corrected for chance: $\kappa = \frac{I-\hat{I}}{1-\hat{I}}$
Accuracy	Fraction of correct predictions: $\text{Accuracy} = \frac{\text{TP+TN}}{\text{Total Observations}}$
Precision	Proportion of true positives among predicted positives: $\text{Precision} = \frac{\text{TP}}{\text{TP+FP}}$
Recall	Proportion of true positives correctly identified: $\text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$
F1-Score	Harmonic mean of precision and recall: $\text{F1} = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}}$

2.7. Metrics for Evaluating Recommendation Systems

Recommender systems are an essential part of the largest known websites today. Without them, it will be difficult to find the right products and content for users [59]. These systems are known as a subset of NLP and ML. Recommender systems support users and developers of various computer and software systems to overcome information overload, perform information discovery tasks and approximate calculations, etc. These models have attracted a wide range of application scenarios from business process modeling to source code manipulation [60]. Due to this wide variety in application areas, different approaches and criteria have been adopted to evaluate them. In this section, we discuss the most important metrics for evaluating these models.

2.7.1. Predicting User Ratings

Recommendation systems often aim to predict how users rate items of interest. To evaluate the accuracy of these predictions, metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) are commonly used [60]. These metrics compare the predicted ratings ($\hat{r}_{ui}$) with the actual ratings (r_{ui}) provided by users. Lower values for these metrics indicate higher accuracy. Additionally, normalized versions of these metrics are used to account for differences in rating scales.RMSE and MAE differ in how they penalize errors. RMSE penalizes larger errors more heavily due to squaring the residuals, making it more sensitive to outliers. If large errors are undesirable, RMSE is preferred. Both metrics can be normalized to account for the rating range, providing scaled versions that facilitate comparison across systems with different rating scales. These metrics provide a quantitative evaluation of the system's ability to predict user ratings accurately.

Table 3. Metrics for Predicting User Ratings [60]

Metric	Formula and Description
Root Mean Squared Error (RMSE)	$RMSE = \sqrt{\frac{\sum_{(u,i)\in T}(\hat{r}_{ui} - r_{ui})^2}{N}}$ Measures the square root of the average squared differences between predicted ($\hat{r}_{ui}$) and actual (r_{ui}) ratings. Penalizes larger errors more heavily.
Mean Absolute Error (MAE)	$MAE = \frac{\sum_{(u,i)\in T} \lvert\hat{r}_{ui} - r_{ui}\rvert}{N}$ Computes the average absolute deviation of predicted ratings from actual ratings. All residuals are equally weighted.
Normalized RMSE (NRMSE)	$NRMSE = \frac{RMSE}{r_{\max} - r_{\min}}$ Scales RMSE based on the range of ratings ($r_{\max} - r_{\min}$), enabling comparisons across systems with different rating scales.
Normalized MAE (NMAE)	$NMAE = \frac{MAE}{r_{\max} - r_{\min}}$ Scales MAE similarly to NRMSE, making it comparable across systems with varying rating ranges.

2.7.2. Ranking-Based Metrics

Ranking-based metrics are essential for evaluating the performance of recommender systems. These metrics provide insights into how well a system can suggest relevant items to users based on their preferences. Below is a detailed overview of common ranking-based metrics, including Precision@K, Recall@K, Mean Reciprocal Rank (MRR@K), Mean Average Precision (MAP@K), Normalized Discounted Cumulative Gain (NDCG@K), and Hit Rate@K. Traditional classification criteria, such as accuracy, precision, recall, and f1, in the context of recommender systems can be used to see how many recommended items are relevant. But they cannot get the order in which the items are recommended. Customers have short attention spans - and it's important to check if the right things are being recommended [61]. The $MAP@K$ metric stands for Mean Average Precision at K and is related to two aspects: 1) Are the predicted items related? 2) Are the most relevant items at the top? will pay. Recommendation quality measures for implicit feedback recommender systems can be calculated as follows:

Precision@K measures the proportion of relevant items among the top-K recommendations. It is calculated as the number of relevant items retrieved in the top-K divided by K,

$$Precision@K = \frac{1}{K}\sum_{i=1}^{K} 1(relevant(i)),$$

where K is The cutoff rank (i.e., the number of top-ranked items considered) and $1(relevant(i))$ is an indicator function that returns 1 if the item at rank i is relevant, and 0 otherwise [62].

Recall@K:It is a metric used to evaluate the effectiveness of a recommendation system by measuring the proportion of relevant items successfully retrieved within the top-K recommendations. It is calculated as the number of relevant items retrieved in the top-K divided by the total number of relevant items:

$$Recall@K = \frac{1}{|R|}\sum_{i=1}^{K} 1(relevant(i)),$$

where $|R|$ is the total number of relevant items for the given user or query [63]. While *Precision@K* evaluates the proportion of relevant items among the top-K recommendations, *Recall@K* measures the proportion of all relevant items that are successfully retrieved within the top-K recommendations.

The Mean Reciprocal Rank (MRR) is a widely used evaluation metric in information retrieval and recommendation systems. It measures the effectiveness of a system in ranking relevant items higher in a list of recommendations. The importance of MRR lies in its ability to capture the rank of the first relevant item in a recommendation list. A higher MRR score indicates that the system consistently places relevant items closer to the top, thereby improving user satisfaction and engagement. The MRR for a set of Q queries or users is defined as the average of the reciprocal ranks of the first relevant item for each query. Let rank_i denote the rank position of the first relevant item for the i-th query. The formulation of MRR is given as:

$$\text{MRR} = \frac{1}{Q} \sum_{i=1}^{Q} \frac{1}{\text{rank}_i}, \tag{31}$$

The reciprocal rank, $\frac{1}{\text{rank}_i}$, assigns higher scores to relevant items that appear earlier in the ranking. MRR aggregates these scores across all queries, providing a single metric that reflects the system's ranking quality [64].

The Average Reciprocal Hit Rank at K (ARHR@k) is a ranking-based evaluation metric used to measure the effectiveness of recommendation systems. It focuses on the position of the first relevant item in the top-K recommendations for each user. ARHR@k extends the Mean Reciprocal Rank (MRR) by restricting the evaluation to the top-K items, making it particularly useful in scenarios where users are only likely to consider a limited number of recommendations.

The ARHR@k is calculated as the average of the reciprocal ranks of the first relevant item for each user, considering only the top-K recommendations. The formula is [64]:

$$\text{ARHR@k} = \frac{1}{U} \sum_{u \in U} \sum_{i=1}^{K} \frac{\delta(i,u)}{i}, \tag{32}$$

where:

- U is the total number of users,
- K is the cutoff rank,
- $\delta(i,u)$ is an indicator function that equals 1 if the item at rank i is relevant to user u, and 0 otherwise,
- i is the rank position of the item.

Mean Average Precision (MAP) is a widely used ranking-based evaluation metric in recommendation systems. It measures the quality of ranked results by considering both the relevance of items and their positions in the ranking. The Mean Average Precision (MAP) is defined as the mean of the Average Precision (AP) scores across all queries. For a single query q, the Average Precision (AP) is calculated as:

$$\text{AP}(q) = \frac{1}{|R(q)|} \sum_{k=1}^{n} \text{Precision@k} \cdot \text{Rel}(k) \tag{33}$$

where $|R(q)|$ is the total number of relevant items for query q, and n is the total number of recommended items. $\text{Rel}(k)$ is a binary indicator function that equals 1 if the item at rank k is relevant to the query q, and 0 otherwise. The MAP is then computed as

$$\text{MAP} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AP}(q) \tag{34}$$

where Q is the total number of queries or users in the evaluation [64]. When the evaluation is constrained to the top K items in the ranked list, the metric becomes Mean Average Precision at K, denoted as MAP@K [65]. In this case, the summation in the AP formula is limited to the top K ranks:

$$\text{AP@K}(q) = \frac{1}{|R(q)|} \sum_{k=1}^{K} \text{Precision@k} \cdot \text{Rel}(k) \tag{35}$$

$$\text{MAP@K} = \frac{1}{Q} \sum_{q=1}^{Q} \text{AP@K}(q) \tag{36}$$

The key difference between MAP and MAP@K is that MAP considers the entire ranked list, while MAP@K evaluates only the top K items, making it more suitable for scenarios where users interact with a limited number of recommendations.

Normalized Discounted Cumulative Gain (nDCG@K) at K is a widely used metric in recommendation systems to evaluate the quality of ranked lists. It measures how well the predicted ranking of items corresponds to the ideal ranking. The DCG for a ranked list of K items is calculated as:

$$\text{DCG@K} = \sum_{i=1}^{K} \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)},$$

where rel_i is the relevance score of the item at position i. The nDCG@K is then computed as:

$$\text{nDCG@K} = \frac{\text{DCG@K}}{\text{iDCG@K}},$$

The nDCG@K is obtained by normalizing the DCG with respect to the iDCG, ensuring that the metric ranges between 0 and 1, where 1 indicates a perfect ranking. where iDCG@K is the DCG of the ideal ranking for the same set of items [64].

2.7.3. Coverage

Coverage is a critical dimension in evaluating recommendation systems, as it measures the extent to which the system can make recommendations across the available information space. It refers to the proportion of available items or users for which the system can generate recommendations. This dimension is particularly important in scenarios where new items or users are introduced, or when there is insufficient data for certain items or users. Coverage is typically categorized into two main types: *catalogue coverage*, which evaluates the proportion of items in the system's catalogue that can be recommended, and *prediction coverage*, which assesses the proportion of users or user interactions for which predictions can be made [60].

Catalogue Coverage

Catalogue coverage measures the proportion of items in the available catalogue that the system can recommend. It is calculated as follows [60]:

$$Catalogue_{Coverage} = \frac{|S_r|}{|S_a|} \tag{37}$$

where:

- S_r: The set of items recommended to users.
- S_a: The set of all available items in the catalogue.

Weighted Catalogue Coverage

Weighted catalogue coverage adjusts the catalogue coverage metric by considering the usefulness of the recommended items. It is defined as [60]:

$$WeightedCatalogue_{Coverage} = \frac{|S_r \cap S_s|}{|S_s|} \tag{38}$$

where:

- S_r: The set of items recommended to users.
- S_s: The set of items considered useful to users.

Prediction Coverage

Prediction coverage measures the proportion of users or user interactions for which the system can generate predictions. It is calculated as [60]:

$$Prediction_{Coverage} = \frac{|S_p|}{|S_u|} \tag{39}$$

where:

- S_p: The set of users for whom predictions can be made.
- S_u: The set of all users in the system.

Weighted Prediction Coverage

Weighted prediction coverage adjusts the prediction coverage metric by considering the usefulness of the recommendations for users. It is defined as [60]:

$$WeightedPrediction_{Coverage} = \frac{\sum_{i \in S_p} f(i)}{\sum_{j \in S_u} f(j)} \tag{40}$$

where:

- S_p: The set of users for whom predictions can be made.
- S_u: The set of all users in the system.
- $f(x)$: A function representing the usefulness of the recommendations for a specific user.

2.7.4. Diversity

Diversity [60,66] in recommendation systems measures the degree of variability or dissimilarity among the items in a recommendation list. High diversity ensures that the recommendations are not redundant and provides users with a broader exploration of the item space. A lack of diversity may result in repetitive or overly similar items, reducing the overall utility of the recommendation system. Several metrics have been introduced to quantify diversity, as described below.

Diversity

Diversity is calculated as the average dissimilarity between all pairs of items in a recommendation list. It is defined as [66]:

$$Diversity(c_1, \ldots, c_n) = \frac{\sum_{i=1}^{n} \sum_{j=i}^{n} (1 - Similarity(c_i, c_j))}{\frac{n \cdot (n-1)}{2}} \tag{41}$$

where:

- $c_1, \ldots, c_n$: Items in the recommendation list.
- $Similarity(c_i, c_j)$: A similarity metric between items c_i and c_j.
- n: The total number of items in the recommendation list.

Similarity

The similarity between an item c and a target query t can be calculated using a weighted sum metric [66]:

$$Similarity(t,c) = \frac{\sum_{i=1}^{n} \omega_i \cdot sim(t_i, c_i)}{\sum_{i=1}^{n} \omega_i} \tag{42}$$

where:

- t: The target query.
- c: The item being compared.
- ω_i: The weight assigned to the i-th feature.
- $sim(t_i, c_i)$: A similarity heuristic between the i-th feature of t and c.

Quality

To balance the trade-off between diversity and similarity, a quality metric is introduced. It combines the similarity of an item c with the target query t and the diversity of c relative to the items already selected in the recommendation list R [66]:

$$Quality(t,c,R) = Similarity(t,c) \cdot RelDiversity(c,R) \tag{43}$$

where:

- $Similarity(t,c)$: The similarity of item c to the target query t.
- $RelDiversity(c,R)$: The diversity of c relative to the items in R.

Relative Diversity

Relative diversity measures the diversity of a new item c relative to the items already selected in the recommendation list $R = \{r_1, \ldots, r_m\}$. It is defined as [66]:

$$RelDiversity(c,R) = \begin{cases} 0, & \text{if } R = \emptyset \\ \frac{\sum_{i=1}^{m}(1 - Similarity(c,r_i))}{m}, & \text{otherwise} \end{cases} \tag{44}$$

where:

- c: The new item being evaluated.
- R: The set of items already selected in the recommendation list.
- $Similarity(c, r_i)$: The similarity between the new item c and an item r_i in R.
- m: The number of items in R.

3. Image Mining

3.1. Evaluation Metrics for 3D Image Segmentation

3.1.1. Spatial Overlap Based Me3trics

- Basic cardinalities:

$$m_{ij} = \sum_{r=1}^{|X|} f_g^i(x_r) f_t^j(x_r) \tag{45}$$

- Generalization to fuzzy segmentations:

$$TP = \sum_{r=1}^{|X|} min(f_t^1(x_r) f_1^g(x_r)) \tag{46}$$

$$FP = \sum_{r=1}^{|X|} max(f_t^1(x_r) f_1^g(x_r), 0) \tag{47}$$

$$TN = \sum_{r=1}^{|X|} min(f_t^2(x_r)f_2^g(x_r)) \tag{48}$$

$$FN = \sum_{r=1}^{|X|} max(f_t^2(x_r)f_2^g(x_r),0) \tag{49}$$

- Calculation of overlap based metrics:

$$Dice = \frac{2|S_g^1 \bigcap S_t^1|}{|S_g^1| + |S_t^1|} = \frac{2TP}{2TP+FP+FN} = \frac{2JAC}{1+JAC} \tag{50}$$

$$jac = \frac{2|S_g^1 \bigcap S_t^1|}{|S_g^1 \bigcup S_t^1|} = \frac{2|S_g^1 \bigcap S_t^1|}{2(|S_g^1| + |S_t^1| - |S_g^1 \bigcap S_t^1|)} = \frac{Dice}{2-Dice} \tag{51}$$

$$Recall = Sensitivity = TPR = \frac{TP}{TP+FN} \tag{52}$$

$$Specificity = TNR = \frac{TN}{TN+FP} \tag{53}$$

$$Fallout = FPR = \frac{FP}{FP+TN} = 1 - TNR \tag{54}$$

$$FNR = \frac{FN}{FN+TP} = 1 - TPR \tag{55}$$

$$Precision = PPV = \frac{TP}{TP+FP} \tag{56}$$

$$FMS_\beta = \frac{(\beta^2+1).PPV.TPR}{\beta^2.PPV+TPR} \tag{57}$$

$$FMS = \frac{2.PPV.TPR}{PPV+TPR} \tag{58}$$

$$E(S_t,S_g,x) = \frac{|R(S_t,x) \setminus R(S_g,x)|}{|R(S_t,x)|} \tag{59}$$

$$GCE(S_t,S_g) = \frac{1}{n}\min\{\sum_i^n E(S_t,S_g,x_i), \sum_i^n E(S_g,S_t,x_i)\} \tag{60}$$

$$GCE = \frac{1}{n}\min\{\frac{FN(FN+2TP)}{TP+FN} + \frac{FP(FP+2TN)}{TN+FP}, \frac{FP(FP+2TP)}{TP+FP} + \frac{FN(FN+2TN)}{TN+FN}\} \tag{61}$$

- Overlap measures for multiple labels:

$$JAC_{ml} = \frac{\sum_{labels,l} \alpha l \sum_{voxels,i} MIN(A_{li},B_{li}}{\sum_{labels,l} \alpha l \sum_{voxels,i} MAX(A_{li},B_{li})} \tag{62}$$

3.1.2. Volume Based Metrics

$$VS = 1 - \frac{||S_t^1| - |S_g^1||}{|S_t^1| + |S_g^1|} = 1 - \frac{|FN-FP|}{2TP+FP+FN} \tag{63}$$

3.1.3. Pair Counting Based Metrics

- Basic cardinalities

$$a = \frac{1}{2}\sum_{i=1}^{r}\sum_{j=1}^{s} m_{ij}(m_{ij}-1) \tag{64}$$

$$b = \frac{1}{2}(\sum_{j=1}^{s} m_j^2 - \sum_{i=1}^{r}\sum_{j=1}^{s} m_{ij}^2) \tag{65}$$

$$c = \frac{1}{2}(\sum_{j=1}^{r} m_i^2 - \sum_{i=1}^{r}\sum_{j=1}^{s} m_{ij}^2) \tag{66}$$

$$d = n(n-1)/2 - (a+b+c) \tag{67}$$

$$a = \frac{1}{2}[TP(TP-1) + FP(FP-1) + TN(TN-1) + FN(FN-1)] \tag{68}$$

$$b = \frac{1}{2}[(TP+FN)^2 + (TN+FP)^2 - (TP^2 + TN^2 + FP^2 + FN^2)] \tag{69}$$

$$c = \frac{1}{2}[(TP+FP)^2 + (TN+FN)^2 - (TP^2 + TN^2 + FP^2 + FN^2)] \tag{70}$$

$$d = n(n-1)/2 - (a+b+c) \tag{71}$$

- Generalization to fuzzy segmentations:
- Calculation of pair-counting based metrics:

$$RI(S_g, S_t) = \frac{a+b}{a+b+c+d} \tag{72}$$

$$ARI = \frac{\sum_{ij}\binom{m_{ij}}{2} - \sum_i\binom{m_i}{2}\sum_j\binom{m_j}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_i\binom{m_i}{2} + \sum_j\binom{m_j}{2}] - \sum_i\binom{m_i}{2}\sum_j\binom{m_j}{2}/\binom{n}{2}} \tag{73}$$

$$ARI = \frac{2(ad-bc)}{c^2 + b^2 + 2ad + (a+d)(c+b)} \tag{74}$$

3.1.4. Information Theoretic Based Metrics

$$H(S) = -\sum_i p(S^i)\log p(S^i) \tag{75}$$

$$H(S_1, S_2) = -\sum_{ij} p(S_1^i, S_2^j)\log p(S_1^i, S_2^j) \tag{76}$$

$$p(S_g^1) = (TP+FN)/n \tag{77}$$

$$p(S_g^2) = (TN+FN)/n \tag{78}$$

$$p(S_t^1) = (TP+FN)/n \tag{79}$$

$$p(S_t^2) = (TN+FP)/n \tag{80}$$

$$p(S_i^i, S_2^j) = \frac{|S_1^i \bigcap S_2^j|}{n} \tag{81}$$

$$p(S_1^1, S_2^1) = \frac{TP}{n} \tag{82}$$

$$p(S_1^1, S_2^2) = \frac{FN}{n} \tag{83}$$

$$p(S_2^1, S_1^2) = \frac{FP}{n} \tag{84}$$

$$p(S_2^1, S_2^2) = \frac{TN}{n} \tag{85}$$

doi:10.20944/preprints202508.1594.v1

$$MI(S_g, S_t) = H(S_g) + H(S_t) - H(S_g, S_t) \tag{86}$$

$$VOI = (S_g, S_t) = H(S_g) + H(S_t) - 2MI(S_g, S_t) \tag{87}$$

3.1.5. Probabilistic Metrics

- Interclass Correlation (ICC):

$$ICC = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_\epsilon^2} \tag{88}$$

$$ICC = \frac{MS_b - MS_w}{MS_b + (k-1)MS_w} \tag{89}$$

$$MS_b = \frac{2}{n-1} \sum_x (m(x) - \mu)^2 \tag{90}$$

$$MS_w = \frac{1}{n} \sum_x (f(x) - m(x))^2 + (f_t(x) - m(x))^2 \tag{91}$$

- Probabilistic Distance (PBD):

$$PBD(A, B) = \frac{\int |P_A - P_B|}{2 \int P_{AB}} \tag{92}$$

$$PBD(S_g, S_t) = \frac{\sum_x |f_g(x) - f_t(x)|}{2 \sum_x f_g(x) f_t(x)} \tag{93}$$

$$KAP = \frac{f_a - f_c}{N - f_c} \tag{94}$$

$$f_a = TP + TN \tag{95}$$

$$f_c = \frac{(TN + FN)(TN + FP) + (FP + TP)(FN + TP)}{N} \tag{96}$$

- ROC curve (Receiver Operating Characteristic)

$$AUC = \sum_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \times \frac{TPR_i + TPR_{i+1}}{2} \tag{97}$$

3.1.6. Spatial Distance Based Metrics

- Distance between crisp volumes:

$$HD(A, B) = max(h(A, B), h(B, A)) \tag{98}$$

$$h(A, B) = \max_{a \in A} \min_{b \in B} ||a - b|| \tag{99}$$

- Average Hausdorff Distance (AVD):

$$AVD(A, B) = max(d(A, B), d(B, A)) \tag{100}$$

$$d(A, B) = \frac{1}{N} \sum_{a \in A} \min_{b \in B} ||a - b|| \tag{101}$$

- Mahalanobis Distance (MHD):

$$MHD(x, y) = \sqrt{(x - y)^T S^{-1} (x - y)} \tag{102}$$

$$MHD(X,Y) = \sqrt{(\mu_x - \mu_y)^T S^{-1} (\mu_x - \mu_y)} \tag{103}$$

$$S = \frac{n_1 + S_1 + n_2 S_2}{n_1 + n_2} \tag{104}$$

- Extending the distances to fuzzy volumes

$$H\hat{D}_k(A,B) = \frac{1}{k} \sum_{i=1}^{k} HD_{\frac{i}{k}}(A,B) \tag{105}$$

$$HD_\alpha(A,B) = HD(A_\alpha, B_\alpha) \tag{106}$$

$$AV\hat{D}_k(A,B) = \frac{1}{k} \sum_{i=1}^{k} AVD(A_{\frac{i}{k}}, B_{\frac{i}{k}}) \tag{107}$$

$$M\hat{H}D_k(A,B) = \frac{1}{k} \sum_{i=1}^{k} MHD(A_{\frac{i}{k}}, B_{\frac{i}{k}}) \tag{108}$$

3.1.7. Multiple Definition of Met3rics in the Literature

$$VS = 2\frac{|S_t \cap S_g|}{|S_t + S_g|}.100\% \tag{109}$$

3.2. Evaluation Metrics for Image Segmentation

3.2.1. Error Types of Image Segmentation

- CR (Correct Region):

$$CR = f(I_g \bigcap I_s) = \sum_{j=1}^{M} \sum_{i=1}^{N} |A_j \bigcap B_i| = \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji} \tag{110}$$

- MD (Miss Detection):

$$MD = f(I_g \bigcap \hat{I}_s = \sum_{j=1}^{M} |A_j| - CR = \sum_{j=1}^{M} |B_i| - \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji} \tag{111}$$

- FA (False Alarm):

$$FA = f(\hat{I}_g \bigcap I_s) = \sum_{i=1}^{N} |B_i| - CR = \sum_{i=1}^{N} |B_i| - \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji} \tag{112}$$

- BR (Background Region):

$$BR = f(\hat{I}_g \bigcap \hat{I}_s \tag{113}$$

- Perfect segmentation:

$$M = N, and \begin{cases} a_{ji} = 0, b_{ji} = |A_i| = |B_i|, c_{ji} = 0 & \text{for} \hat{\delta}(A_j \cap B_i) = 1 \\ a_{ji} = 0, b_{ji} = 0, c_{ji} = 0 & \text{for} \hat{\delta}(A_j \cap B_i) = 0 \end{cases}$$

- Completely incorrect segmentation:

$$\forall_{i,j} \hat{\delta}(A_j \bigcap B_i) = 0 \tag{114}$$

- Isolated false alarm

$$M = N - 1 > 1, and \begin{cases} i \neq N & \text{for} \hat{\delta}(A_j \bigcap B_i) = 1 \\ a_{jN} = |A_j|, b_{jN} = 0, c_{jN} = |B_N| & \text{for} \hat{\delta}(A_j \bigcap B_i) = 0 \end{cases}$$

- Isolated missed detection:

$$N = M - 1 > 1, and \begin{cases} j \neq M & \text{for} \hat{\delta}(A_j \bigcap B_i) = 1 \\ a_{jM} = |A_M|, b_{Mi} = 0, c_{Mi} = |B_i| & \text{for} \hat{\delta}(A_j \bigcap B_i) = 0 \end{cases}$$

- Partial false alarm /miss detection:

$$\exists_i \hat{\delta}(A_i \bigcap B_i) = 1 and a_{ji}, b_{ji}, c_{ji} \neq 0, N = M - 1 > 1 \tag{115}$$

- Over segmentation

$$BS_j = \{B_i | \hat{\delta}(A_i \bigcap B_i) = 1, i = 1, ..., k\}, and |A_j| > |BS_j| \tag{116}$$

- Under segmentation

3.2.2. Evaluation Metrics

- Pixel-based metric:

$$\begin{aligned} Precision = \frac{CR}{CR + FA} &= \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}}{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji} + \sum_{i=1}^{N} |B_i| - \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}} \\ &= \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}}{\sum_{i=1}^{N} |B_i|} \end{aligned} \tag{117}$$

$$\begin{aligned} Recall = \frac{CR}{CR + MD} &= \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}}{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji} + \sum_{j=1}^{M} |A_j| - \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}} \\ &= \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}}{\sum_{i=1}^{N} |B_i|} \end{aligned} \tag{118}$$

$$F = \frac{2.Precision.Recall}{Precision + Recall} = \frac{2 \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \cap B_i) b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \tag{119}$$

- Object-based metric:Martin's method

$$P_{ji} = (1 - \frac{|A_j \cap B_i|}{|A_j|}.|A_j \bigcap B_i| = (1 - \frac{b_{ji}}{a_{ji} + b_{ji}}).b_{ji} = \frac{a_{ji} + b_{ji}}{a_{ji} + b_{ji}} \tag{120}$$

$$Q_{ji} = (1 - \frac{|A_j \cap B_i|}{|B_i|}.|A_j \bigcap B_i| = (1 - \frac{b_{ji}}{c_{ji} + b_{ji}}).b_{ji} = \frac{c_{ji} + b_{ji}}{c_{ji} + b_{ji}} \tag{121}$$

$$GCE(I_g, I_s) = \frac{1}{n} \min\{\sum_{j=1}^{M} \sum_{i=1}^{N} P_{ji}, \sum_{j=1}^{M} \sum_{i=1}^{N} Q_{ji}\} \tag{122}$$

where $n = \sum_{j=1}^{M} \sum_{i=1}^{N} |A_j \bigcap B_i|$

$$LCE(I_g, I_s) = \frac{1}{n} \sum_{j=1}^{M} \sum_{i=1}^{N} \min(P_{ji}, Q_{ji}) \tag{123}$$

- Object-based metric: Polak's method:

$$W_{ji} = \frac{\hat{\delta}(A_j \bigcap B_i)|B_i|}{\sum_{K=1}^{N} \hat{\delta}(A_j \bigcap B_k)|B_k|} = \frac{\hat{\delta}(b_{ji}(b_{ji} + c_{ji}}{\sum_{k=1}^{N}} \tag{124}$$

3.2.3. Error Measures

Perfect segmentation:

$$\begin{aligned} F = \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} &= \frac{\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji} + \sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \\ &= \frac{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \end{aligned} \tag{125}$$

Completely inaccurate segmentation

$$F = \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N} 0.b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = \frac{0}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = 0 \tag{126}$$

Isolated false alarm

$$\begin{aligned} F &= \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N-1} \hat{\delta}(A_j \bigcap B_i) b_{ji} + 2\sum_{j=1}^{M} 0 \cdot b_{j,N}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \\ &= \frac{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N-1} |B_i|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = \frac{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i| - |B_N|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \\ &= 1 - \frac{|B_N|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = 1 - \frac{|B_N|}{2\sum_{i=1}^{N} |B_i| - |B_N|} \\ &= 1 - \frac{|B_N|}{2|B| - |B_N|} \end{aligned} \tag{127}$$

Isolated miss detection

$$\begin{aligned} F &= \frac{2\sum_{j=1}^{M-1} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji} + 2\sum_{i=1}^{N} 0 \cdot b_{Mi}}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} = \frac{\sum_{i=1}^{N} |B_i| + \sum_{j=1}^{M-1} |A_j|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \\ &= \frac{\sum_{i=1}^{N} |B_i| + \sum_{j=1}^{M} |A_j| - |A_M|}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \\ &= 1 - \frac{|A_M|}{2\sum_{j=1}^{M} |A_j| - |A_M|} = 1 - \frac{|A_M|}{2|A| - |A_M|} \end{aligned} \tag{128}$$

Partial false alarm/miss detection

$$F = \frac{2\sum_{j=1}^{M} \sum_{i=1}^{N} \hat{\delta}(A_j \bigcap B_i) b_{ji})}{\sum_{j=1}^{M} |A_j| + \sum_{i=1}^{N} |B_i|} \in [0, 1] \tag{129}$$

Over segmentation

$$F = \frac{2\sum_{i=1}^{k}\hat{\delta}(A_1\bigcap B_i)b_{1i}}{|A_1|+|BS|} = \frac{2\sum_{i=1}^{K} b_{1i}}{|A_1|+|BS|} = \frac{2|BS|}{|A1|+|BS|} > \frac{|BS|}{|A_1|} = \frac{CR}{CR+MD} = Recall \tag{130}$$

Under segmentation

Table 4. Comparison of error measure abilities for three error metrics.

Types	Pixel based	Martin's metrics	Polak's metrics
Perfect segmentation	✓	✓	✓
Completely inaccurate segmentation	✓	×	×
Isolated false alarm	✓	×	×
Isolated missed detection	✓	×	×
Partial false alarm/ missed detection	✓	×	□
Over-segmentation	✓	×	✓
Under-segmentation	✓	×	✓

4. Evaluating Metrics for Clustering Algorithms

The landscape of clustering algorithms spans classical geometric methods like partition-based K-Means [67] and Mini-Batch K-Means [68], hierarchical approaches such as Agglomerative [69] and BIRCH [70], density-driven techniques including DBSCAN [71] and OPTICS [72], affinity-based models like Affinity Propagation [73] and Mean Shift [74], probabilistic frameworks such as Gaussian Mixture Models [75], and spectral methods like Spectral Clustering [76], alongside modern deep learning paradigms such as Deep Embedded Clustering (DEC) [77], Deep Adaptive Clustering (DAC) [78], Information Maximizing Self-Augmented Training (IMSAT) [79], Variational Deep Embedding (VaDE) [80], and the recently proposed EDCWRN [81] that integrates representation weighting and neighborhood relationships. Evaluating these diverse methodologies necessitates metrics tailored to data characteristics: traditional criteria (e.g., inertia, silhouette scores) assume Euclidean structures, while emerging challenges—such as mixed interval-categorical data or non-Euclidean distances—demand specialized validation frameworks. This section reviews classical and recently developed metrics, emphasizing their applicability to both conventional and deep clustering architectures. To rigorously define these metrics, we first formalize the foundational elements of clustering [82]:

Dataset Definition. Let X denote a dataset containing N objects, where each object is represented as a vector in an F-dimensional space:

$$X = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}, \quad X \subseteq \mathbb{R}^F.$$

Clustering Partition. A partition (or clustering) of X divides the dataset into K distinct, non-overlapping clusters:

$$\mathcal{C} = \{C_1, C_2, \ldots, C_K\},$$

satisfying:

1. **Coverage**: $\bigcup_{k=1}^{K} C_k = X$,
2. **Disjointness**: $C_k \cap C_l = \emptyset \quad \forall k \neq l$.

Cluster and Dataset Centroids.

- The **cluster centroid** (mean vector) for C_k:

$$\boldsymbol{\mu}_k = \frac{1}{|C_k|}\sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$$

- The **dataset centroid** (global mean):

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{\mathbf{x}_i \in X} \mathbf{x}_i$$

4.1. Cluster Validity Indices (CVIs)

The evaluation of clustering results is a critical aspect of the clustering process, often achieved using cluster validity indices (CVIs) that assess the quality of partitions based on various criteria such as compactness and separation. While several comparative studies of CVIs exist, including the extensive analysis by Arbelaitz *et al.,arbelaitz2013extensive*, which systematically evaluated 30 indices across diverse configurations, this section provides a structured overview of these indices. By organizing their definitions and key properties in a tabular format, we aim to offer a clear and accessible reference for understanding the characteristics and applicability of each CVI in different clustering scenarios.

Table 5. This is a table caption. Tables should be placed in the main text near to the first time they are cited.

Index Name	Description/Definition	Citation
Dunn Index (D↑)	Measures the ratio of the minimum inter-cluster distance to the maximum intra-cluster distance. Higher values indicate better clustering.	[83]
Calinski-Harabasz Index (CH↑)	Ratio of the sum of between-cluster dispersion to within-cluster dispersion. Higher values indicate better clustering.	[84]
Davies-Bouldin Index (DB↓)	Measures the average similarity between each cluster and the cluster most similar to it. Lower values indicate better clustering.	[85]
Silhouette Index (Sil↑)	Combines cohesion (average intra-cluster distance) and separation (average nearest-cluster distance) into a single measure. Values range from -1 to 1, where higher values indicate better clustering.	[86]
Gamma Index (G↓)	Measures the correlation between the distances of pairs of points and their clustering assignments. Higher values indicate better clustering.	[87]
C Index (CI↓)	Normalized measure of intra-cluster distances compared to minimum and maximum possible distances. Lower values indicate better clustering.	[88]
S_Dbw Index (SDbw↓)	Combines intra-cluster variance and inter-cluster density to evaluate clustering quality. Lower values indicate better clustering.	[89]
CS Index (CS↓)	Estimates cohesion by cluster diameters and separation by the nearest neighbor distance. Lower values indicate better clustering.	[90]
Symmetry Index (Sym↑)	Measures the symmetry of clusters around their centroids using the Point Symmetry Distance. Lower values indicate better clustering.	[91]
COP Index (COP↓)	Combines intra-cluster cohesion (distance to centroid) and inter-cluster separation (furthest neighbor distance). Lower values indicate better clustering.	[92]
Negentropy Increment (NI↓)	Based on entropy, measures the increase in negentropy when clustering is applied. Higher values indicate better clustering.	[93]
Score Function (SF↑)	Combines between-cluster dispersion and within-cluster dispersion using a weighted exponential function. Higher values indicate better clustering.	[94]

Table 5. *Cont.*

Index Name	Description/Definition	Citation
Generalized Dunn Index (gD31, gD41, gD51, gD33, gD43, gD53)	Variants of the Dunn index using different measures for cohesion and separation. Higher values indicate better clustering.	[95]
Graph theory based Dunn and Davies–Bouldin variations	Dunn index variants that use graph-based measures (Minimum Spanning Tree, Relative Neighborhood Graph, Gabriel Graph) for cohesion.	[96]
Point Symmetry-Based Indices (SymDB↓, SymD↑, Sym33↑)	Variants of the Davies-Bouldin, Dunn, and Generalized Dunn indices using Point Symmetry Distance for cohesion. Lower values indicate better clustering.	[97]
Davies–Bouldinn* (DB*↓)	A modification of the Davies-Bouldin index.	[98]
SV Index (SV↑)	Combines separation (nearest neighbor distance) and cohesion (distance of border points to centroid). Lower values indicate better clustering.	[99]
OS Index (OS↑)	Combines a more complex separation estimator with cohesion (distance of border points to centroid). Lower values indicate better clustering.	[99]

4.2. *Internal Validation Methods*

Internal validation methods are used to assess the quality of clustering results without relying on external information, such as ground truth labels. These methods evaluate the clustering structure based solely on the input data, making them particularly useful in unsupervised learning tasks.

Internal validation metrics are particularly versatile and are commonly applied to two major families of clustering algorithms: *partitional clustering algorithms* (e.g., k-means, k-medoids) and *hierarchical clustering algorithms* (e.g., agglomerative or divisive methods) [100]. Below, we delve into the specifics of these metrics and their applications to these algorithmic families.

4.2.1. Metrics for Partitional Clustering Algorithms

Partitional clustering algorithms divide the data into non-overlapping groups. Internal validation metrics for these algorithms are typically based on proximity measures (e.g., distance or similarity) and aim to evaluate how well the clusters are formed. Some commonly used metrics include:

- **Sum of Squared Errors (SSE):** Measures the compactness of clusters by calculating the sum of squared distances between each data point and its cluster centroid. Lower SSE values indicate better clustering.
- **Silhouette Coefficient:** Combines cohesion and separation into a single metric. It compares the average distance of a point to other points in its cluster (cohesion) with the average distance to points in the nearest neighboring cluster (separation). Values range from -1 to 1, where higher values indicate better clustering.
- **Calinski-Harabasz Index (CH):** Also known as the variance ratio criterion, this index measures the ratio of between-cluster dispersion to within-cluster dispersion. Higher CH values indicate better-defined clusters.
- **Dunn Index:** Evaluates the ratio of the smallest distance between clusters (separation) to the largest intra-cluster distance (cohesion). A higher Dunn Index signifies better clustering.
- **Xie-Beni Index:** Originally designed for fuzzy clustering, this metric estimates the compactness and separation of clusters. It is also applicable to hard clustering methods.

4.2.2. Metrics for Hierarchical Clustering Algorithms

Hierarchical clustering algorithms produce a tree-like structure (dendrogram) that represents nested groupings of data. Internal validation metrics for these algorithms often evaluate the quality of the dendrogram itself. Key metrics include:

- **Cophenetic Correlation Coefficient (CPCC):** Measures how faithfully the dendrogram preserves the pairwise distances between data points. High CPCC values indicate that the hierarchical clustering algorithm has captured the underlying structure of the data.
- **Hubert Statistic:** Similar to CPCC, this metric evaluates the concordance between the proximity matrix and the cophenetic matrix derived from the dendrogram. Higher Hubert values suggest better clustering.
- **Proximity Matrix Correlation:** Compares the actual proximity matrix to an idealized block-diagonal matrix based on the clustering results. High correlation indicates well-formed clusters.

4.2.3. Innovative Perspective on Internal Validation

While traditional internal validation metrics like SSE and Silhouette Coefficient are widely used, they often align closely with the optimization objectives of specific clustering algorithms (e.g., k-means minimizes SSE). This alignment can introduce bias, making it challenging to compare different algorithms fairly. To address this limitation, we propose a novel approach that combines multiple metrics into a unified framework, enabling a more holistic evaluation of clustering results. By integrating cohesion, separation, and proximity measures, our method provides a comprehensive assessment of clustering quality, independent of the algorithm used.

Furthermore, we emphasize the importance of computational efficiency in large-scale clustering tasks. Metrics like the Silhouette Coefficient, with a complexity of $O(n^2)$, may become impractical for large datasets. To overcome this, we advocate for the development of approximate metrics that retain the interpretability of traditional indices while significantly reducing computational costs.

4.2.4. Conclusion

Internal validation methods play a pivotal role in unsupervised learning by enabling researchers to evaluate clustering results without relying on external labels. By focusing on cohesion and separation, these methods provide valuable insights into the quality of clustering structures. However, as clustering tasks grow in complexity and scale, there is a pressing need for innovative metrics that balance interpretability, fairness, and computational efficiency. We hope that the framework presented here inspires future research and establishes a robust foundation for evaluating clustering algorithms.

4.2.5. Partitional Methods

Partition clustering assigns a set of data to k clusters using repeated processes. In these processes, n data are classified into k-clusters. The predefined criterion function J assigns the data to the kth numerical set according to the calculation of maximization and minimization in k sets. The internal validation value of a set of K clusters can be decomposed as the sum of the validation values for each cluster as follows [101]:

$$general_{validity} = \sum_{i=1}^{K} w_i validity(C_i) \tag{131}$$

- **cohesion**:Cohesion evaluates how close the elements of a cluster are to each other. This index is defined as follows [100]:

$$cohesion(C_i) = \sum_{x \in C_i, y \in C_i} proximity(x, y) \tag{132}$$

- **Separation**:eparation measures the level of separation between clusters. This index is defined as follows [100]:

$$separation(C_i, C_j) = \sum_{x \in C_i, y \in C_j} proximity(x, y) \tag{133}$$

- **Sum of Squared Errors Within Cluster**:It should be noted that the coherence metric defined above is equivalent to the cluster Sum of Squared Errors Within Cluster(SSW). In general, this metric can be defined as follows [101]:

$$SSE(C_i) = \sum_{x \in C_i} d(c_i, x)^2 = \frac{1}{2m_i} \sum_{x \in C_i} \sum_{y \in C_i} d(x, y)^2 \tag{134}$$

- **Maximize distance between clusters**: This metric maximizes the distance between clusters and is defined as follows:

$$SSB = \sum_{i=1}^{K} m_i d(c_i, c)^2 = \frac{1}{2K} \sum_{i=1}^{k} \sum_{j=1}^{k} \frac{m}{K} d(c_i, c_j)^2 \tag{135}$$

- **Calisnki-Harabasz coefficient(CH)**: The CH, makes the final decision based on the measure of intra-cluster dispersion and inter-cluster dispersion. The goal is to choose the number of clusters that maximize CH. This criterion is known as the variance ratio criterion and is defined as follows [84]:

$$CH = \frac{\frac{SSB_m}{(M-1)}}{\frac{SSE_M}{(M)}} \tag{136}$$

- **Dunn index**: This index has been reviewed before and can also be defined as follows:

$$D = \min_{1<i<k} \{ \min_{1<j<k, i \neq j} \{ \frac{\delta(C_i, C_j)}{\max_{1<l<K} \{\Delta(C_l)\}} \} \} \tag{137}$$

$$\Delta(C_i) = \max_{x,y \in c_i} \{d(x, y)\} \tag{138}$$

$$\delta(C_i, C_j) = \min_{x \in C_i, y \in C_j} \{d(x, y)\} \tag{139}$$

- **Xie-Beni score**:This index is ratio-based, and its numerator estimates the level of data density in the same cluster, and its denominator estimates the level of separation of data from different clusters. It is designed for fuzzy clustering, but it can also be used for hard clustering. This index can be defined as follows [102]:

$$XB = \frac{\sum_{i=1}^{M} \sum_{k=1}^{M} u_{ik}^2 ||\|x_i, C_k\||^2}{N_{t \notin s} \min\{\||C_t - C_s|\|\}} \tag{140}$$

- **Ball-Hall index**: This index is a dispersion measure that performs clustering based on the quadratic distance of the cluster points concerning their center and is defined as follows [103]:

$$BH = \frac{SSE_M}{M} \tag{141}$$

- **Hartigan index**: This index performs clustering based on the logarithmic relationship between the sum of squares within the cluster and the sum of squares between the clusters and is defined as follows [104]:

$$H = \log(\frac{SSB_M}{SSE_M}) \tag{142}$$

- **Xu coefficient**: This coefficient considers the clustering of dimensions D of data, the number N of data samples, and the sum of squared errors of $SSEM$ from M clusters [105]:

$$Xu = D\log_2(\sqrt{\frac{SSE_M}{DN^2}}) + \log M \tag{143}$$

- **silhouette coefficient**This index has been reviewed before and can also be defined as follows [100]:

$$a(i) = \frac{1}{|C_a|}\sum_{j\in C_a, i\notin j} d(i,j) \tag{144}$$

$$b(i) = \min_{C_b \notin C_a} \frac{1}{|C_b|}\sum_{j\in C_b} d(i,j) \tag{145}$$

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}} \tag{146}$$

$$S = \frac{1}{n}\sum_{i=1}^{n} s(i) \tag{147}$$

4.2.6. Hierarchical Methods

This clustering is a step-by-step method of creating a hierarchy of clusters, which can be defined as agglomerative (bottom-up approach) and divisive (top-down approach). In agglomerative mode, the clustering algorithm starts by considering each data point as a separate cluster. So, if there are N data points, it starts with N clusters. At each step, the algorithm merges the two clusters that are closest to each other until all the clusters are merged into one big cluster containing all the data points. In the divisive approach, it starts with all the data points in one cluster. At each step, the algorithm divides the cluster until each cluster contains only one data point. This method is less common and more computationally intensive compared to cumulative clustering [106]. Several internal validation techniques have also been proposed and tested with hierarchical clustering algorithms, which are discussed below.

- **Cophenetic Correlation Coefficient (CPCC)**: A hierarchical clustering algorithm is used to evaluate the results. This correlation coefficient was proposed as the correlation between the cophenetic matrix P_c, containing cophenetic distances, and the proximity matrix P, containing similarities. This correlation coefficient is defined as [107].

$$CPCC = \frac{\sum_{i<j}(d_{i,j} - \hat{d})(d^*_{i,j} - \hat{d}^*)}{\sqrt{\sum_{i<j}(d_{i,j} - \hat{d})^2 \sum_{i<j}(d^*_{i,j} - \hat{d}^*{}^2}} \tag{148}$$

$$\hat{d} = \frac{\sum_{i<j} d_{ij}}{2(n^2 - n)} \tag{149}$$

$$\hat{d}^* = \sqrt{\frac{\sum_{i<j}(d_{i,j} - \hat{d})^2}{\sum_{i<j}(d^*_{ij})^2}} \tag{150}$$

- **Hubert Statistic**: Let S_+ and S_- be the number of congruent and discordant pairs, respectively. This index is defined as [100]:

$$\gamma = \frac{S_+ - S_-}{S_+ + S_-} \tag{151}$$

4.3. *External Validation Met3hods*

These validation methods are usually defined in the form of supervised learning. These methods are performed by including additional information in the clustering validation process, such as external class labels for the training samples. Since unsupervised learning techniques are generally used when such information is not available, external validation methods are not used in most clustering problems. However, they can be applied when external information is available as well as when generating synthetic data from a real data set [108]. These evaluation criteria are divided into three subsections:

4.3.1. Matching Sets

Some methods identify the relationship between each cluster identified in C and its natural correspondence to the resulting reference classes defined by P.

- **Precision:** Counts the true positives.

$$Pr = \frac{TP}{TP + FP} \tag{152}$$

- **Recall:** Calculates the percentage of elements that are correctly included in a cluster:

$$R = \frac{TP}{TP + FN} \tag{153}$$

- **F-Measure:** combines precision and recall in a single metric.

$$F_a = \frac{2 * Pr * R}{Pr + R} \tag{154}$$

- **Purity:** It is used to evaluate whether each cluster contains only instances of the same class.

$$U = \sum_i p_i (\max_j \frac{p_{ij}}{p_i} \tag{155}$$

Where $pi = \frac{n_i}{n}$, $p_j = \frac{n_j}{n}$ and $p_{ij} = \frac{n_{ij}}{n}$

4.3.2. Peer-to-Peer Correlation

- **Jaccard coefficient**:

$$J = \frac{TP}{TP + FP + FN} \tag{156}$$

$$J = \frac{\sum_{ij} \binom{n_{ij}}{2}}{\sum_i \binom{n_i}{2} + \sum_j \binom{n_j}{2} - \sum_{ij} \binom{n_{ij}}{2}} \tag{157}$$

- **Rand coefficient:**

$$Rand = \frac{TP + TN}{M} \tag{158}$$

$$Rand = \frac{\binom{n}{2} - \sum_i \binom{n_i}{2} + \sum_{ij} \binom{n_{ij}}{2} - \binom{n}{2}}{\binom{n}{2}} \tag{159}$$

- **Folkes and Mallows coefficient:**

$$FM = \sqrt{\frac{TP}{TP + FP} + \frac{TP}{TP + FN}} \tag{160}$$

$$FM == \frac{\sum_{ij} \binom{n_{ij}}{2}}{\sqrt{\sum_i \binom{n_i}{2} * \sum_j \binom{n_j}{2}}} \tag{161}$$

doi:10.20944/preprints202508.1594.v1

- **Hubert statistical coefficient:**

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X_{ij} Y_{ij} \tag{162}$$

$$\Gamma = \frac{\binom{n}{2} \sum_{ij} \binom{n_i j}{2} - \sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2}}{\sqrt{\sum_i \binom{n_i}{2} \sum_j \binom{n_j}{2} [\binom{n}{2} - \sum_i \binom{n_i}{2}][\binom{n}{2} - \sum_j \binom{n_j}{2}]}} \tag{163}$$

4.3.3. Measures Based on Information Theory

- **Entropy**:

$$H = -\sum_i p_i (\sum_i \frac{p_{ij}}{p_i} \log \frac{p_{ij}}{p_i} \tag{164}$$

- **Mutual information**:

$$MI = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{p_i p_j} \tag{165}$$

where $p_{ij} = n_{ij}/n$,$p_i = n_i/n$, and $p_j = n_j/n$.

5. Signal Mining

5.1. Similarity and Distance Met3rics

In this section, similarity and distance criteria are introduced. These criteria are used on two or more signals. Let us assume that we have two sequences, $P = p(k)$, $k = 1, 2, \ldots, K$, and $Q = q(k)$, $k = 1, 2, \ldots, K$, where $p(k)$ and $q(k)$ are the observed values of P and Q at time k, respectively. A variety of typical metrics, that are potentially available for EEG analysing, are introduced to measure the similarity and distance between P and Q.

5.1.1. Euclidean Distance (ED)

The ED between P and Q calculated as:

$$d^{ED} = \sqrt{\sum_{k=1}^{n} ((p(k) - q(k))^2} \tag{166}$$

The ED similarity calculated as follow:

$$S^{ED} = \frac{1}{d^{ED}} \tag{167}$$

5.1.2. Pearson Correlation Coefficient Distance (PCCD)

The PCCD between P and Q can be calculated by

$$d^{PCCD} = \frac{\sum_{k=1}^{k} (p(k) - \hat{p})(q(k) - \hat{q}}{\sqrt{\sum_{k=1}^{k} (p(k) - \hat{p})} \sqrt{\sum_{k=1}^{k} (q(k) - \hat{q}}} \tag{168}$$

So, the similarity defined by PCCD is then calculated by

$$S^{PCCD} = |d^{PCCD}| \tag{169}$$

5.1.3. Symmetric Kullback–Leibler Divergence (SKLD)

The SKLD between P and Q can be calculated by:

$$D(P||Q) = \sum_{k=1}^{K} (P(k) log(\frac{p(k)}{q(k)})) \tag{170}$$

$$D(Q||P) = \sum_{k=1}^{K}(q(k)log(\frac{q(k)}{p(k)})) \tag{171}$$

Kullback–Leibler distance metrics [35] is calculated by:

$$d^{SKLD} = \frac{D(P||Q) + D(Q||P)}{2} \tag{172}$$

Then, the similarity can be gotten as:

$$S^{SKLD} = \frac{1}{d^{SKLD}} \tag{173}$$

5.1.4. Hellinger Distance (HD)

The HD between P and Q can be calculated by

$$d^{HD} = \frac{1}{\sqrt{2}}||\sqrt{P} - \sqrt{Q}||_2 \tag{174}$$

5.1.5. Bhattacharyya Distance (BD)

BD which was proposed by Bhattacharyya in [40], also known as the Hellinger. The BD between P and Q is defined as:

$$s^{BD} = -ln(BC(P,Q)) \tag{175}$$

where $BC(X,Y)$ is the Bhattacharyya coefficient.

$$BC(P,Q) = \sum_{k=1}^{K}\sqrt{p(k)q(k)} \tag{176}$$

5.1.6. Minkowski

In general, the distance d_{ij} between any two points in n-dimensional space may be calculated by the equation given by Minkowski:

$$d_{ij} = [\sum_{i=k}^{n}|p(k) - q(k)|^{PP}]^{\frac{1}{PP}} \tag{177}$$

PP determining the type of distance:

- $PP = 1$: city block distance, or Manhattan distance.
- $PP = 1$: In binary data, Hamming distance.
- $PP = 2$: well-known Euclidean distance.

5.1.7. Entropy

Entropy can be used to measure the difference between two probability distributions. Considering two different probability distributions P and Q, their relative entropy can be described in the following form:

$$D_{EN}(P||Q) = \sum_{x=1}^{r}P(x).log_2\frac{P(x)}{Q(x)} \tag{178}$$

5.1.8. Kullback-Leibler

In order to make the relative entropy able to satisfy the definition of the distance, the redefined formula is rewritten as[REF]:

$$RD(P||Q) = \frac{D_{kl}(P||Q) + D_{KL}(Q||P)}{2} \tag{179}$$

From the above analysis, there is:

$$RD(P||Q) = \{\frac{\sum_{x=1}^{r} P(x).log_2[\frac{P(x)}{Q(x)}] + \sum_{x=1}^{r} Q(x).log_2[\frac{Q(x)}{P(x)}]}{2}\}$$
$$= \frac{\sum_{x=1}^{r}[P(x) - Q(x).log_2[\frac{P(x)}{Q(x)}]}{2}$$
$$= \frac{\sum_{x=1}^{r}[Q(x) - P(x)].log_2[\frac{Q(x)}{P(x)}]}{2} = RD(Q||P) \quad (180)$$

5.1.9. Angle

The angle between two vectors can be calculated using the following formula:

$$\theta = acos(\frac{\overrightarrow{P}.\overrightarrow{Q}}{|\overrightarrow{P}|.|\overrightarrow{Q}|}) \quad (181)$$

where θ represents the angle between the vectors, $\vec{u}$ and $\vec{v}$ are the two vectors, $\vec{u} \cdot \vec{v}$ is the dot product of the vectors, and $\overrightarrow{P}$ and $\overrightarrow{Q}$ denote the magnitudes of the vectors $|\overrightarrow{P}|$ and $|\overrightarrow{Q}|$ respectively.

5.1.10. Itakura-Saito

The Itakura–Saito distance (or Itakura–Saito divergence) is a measure of the difference between an original spectrum and an approximation of that spectrum. Although it is not a perceptual measure, it is intended to reflect perceptual (dis)similarity. It was proposed by Fumitada Itakura and Shuzo Saito in the 1960s while they were with NTT.

$$Dist(P,Q) = \frac{1}{2\Pi}\int_{-\Pi}^{\Pi}[\frac{P}{Q} - log\frac{P}{Q} - 1]dw \quad (182)$$

5.1.11. Dice, Cosine, and Jaccard Coefficient [Reff]

$$DiceCoefficient = 2\frac{|P \bigcap Q|}{|P| + |Q|} \quad (183)$$

$$CosineCoefficient = 2\frac{|P \bigcap Q|}{|P|^{\frac{1}{2}} + |Q|^{\frac{1}{2}}} \quad (184)$$

$$JaccardCoefficient = 2\frac{|P \bigcap Q|}{|P| + |Q| - |P \bigcap Q|} \quad (185)$$

6. Conclusions

Machine learning and deep learning are powerful tools in learning model research and are becoming increasingly pervasive. The adoption of these systems is expanding, accelerating the shift towards a more algorithmic society, meaning that algorithm-based decisions have greater potential for significant societal impact. However, most of these sophisticated decision support systems remain complex black boxes, meaning that their internal logic and inner workings are hidden from the user, and even experts cannot fully understand the logic behind their predictions. In addition, new regulations and highly regulated fields have mandated auditability and verifiability of decisions, increasing the demand for the ability to question, understand, and trust machine learning systems, for which interpretability is essential. Indeed, care must be taken when choosing metrics to evaluate model outputs and interpret results. There is no single metric that fits all. Several appropriate performance metrics should be selected based on an understanding of the dataset and the research question. Reproducibility of results is crucial for readers to trust the reported metrics, as is a discussion of potential biases in the data and model that could affect the metrics. After reporting model performance, biases should be considered. This paper extensively reviews evaluation metrics appropriate for machine learning and deep learning tasks such as classification, clustering, and ranking. Our main contribution is to systematically categorize and discuss these evaluation metrics based on the learning problem and their application context. Unlike previous reviews, we conducted a broad survey. Choosing appropriate evaluation metrics is crucial for the effective evaluation of machine learning models. Different metrics provide distinct insights depending on the problem type, data characteristics, and application goals. Hence, we attempted to review nearly 250 different evaluation metrics in different categories, including text mining, image mining, clustering, and signal mining. Future research could be devoted to evaluating criteria in large language models. This area of research has devoted itself to different criteria for interpretability.

Author Contributions: First Author: Conceptualization, Methodology, Software, Writing—original draft, Writing—review & editing. **Second Author:** Methodology, Software, Writing—review & editing. **Last Author:** Supervision, Methodology, Validation.

References

1. Chowdhary, K. Natural language processing. *Fundamentals of artificial intelligence* **2020**, pp. 603–649.
2. Zhao, X.; Wang, L.; Zhang, Y.; Han, X.; Deveci, M.; Parmar, M. A review of convolutional neural networks in computer vision. *Artificial Intelligence Review* **2024**, *57*, 99.
3. Prabhavalkar, R.; Hori, T.; Sainath, T.N.; Schlüter, R.; Watanabe, S. End-to-end speech recognition: A survey. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2023**, *32*, 325–351.
4. Roy, D.; Dutta, M. A systematic review and research perspective on recommender systems. *Journal of Big Data* **2022**, *9*, 59.
5. Ahmadi, M.; Nia, M.F.; Asgarian, S.; Danesh, K.; Irankhah, E.; Lonbar, A.G.; Sharifi, A. Comparative analysis of segment anything model and u-net for breast tumor detection in ultrasound and mammography images. *arXiv preprint arXiv:2306.12510* **2023**.
6. Wang, J.; Wang, S.; Zhang, Y. Deep learning on medical image analysis. *CAAI Transactions on Intelligence Technology* **2025**, *10*, 1–35.
7. Richard, G.; Lostanlen, V.; Yang, Y.H.; Müller, M. Model-Based Deep Learning for Music Information Research: Leveraging diverse knowledge sources to enhance explainability, controllability, and resource efficiency [Special Issue On Model-Based and Data-Driven Audio Signal Processing]. *IEEE Signal Processing Magazine* **2025**, *41*, 51–59.
8. Nia, M.F. Explore Cross-Codec Quality-Rate Convex Hulls Relation for Adaptive Streaming. *arXiv preprint arXiv:2408.09044* **2024**.
9. Kessler, R.; Béchet, N. Deep Learning and Natural Language Processing in the Field of Construction. *arXiv preprint arXiv:2501.07911* **2025**.
10. Farhadi Nia, M.; Ahmadi, M.; Irankhah, E. Transforming dental diagnostics with artificial intelligence: advanced integration of ChatGPT and large language models for patient care. *Frontiers in Dental Medicine* **2025**, *5*, 1456208.
11. Norcéide, F.S.; Aoki, E.; Tran, V.; Nia, M.F.; Thompson, C.; Chandra, K. Positional Tracking of Physical Objects in an Augmented Reality Environment Using Neuromorphic Vision Sensors. In Proceedings of the 2024 International Conference on Machine Learning and Applications (ICMLA). IEEE, 2024, pp. 1031–1036.
12. Mhamed, M.; Zhang, Z.; Hua, W.; Yang, L.; Huang, M.; Li, X.; Bai, T.; Li, H.; Zhang, M. Apple varieties and growth prediction with time series classification based on deep learning to impact the harvesting decisions. *Computers in Industry* **2025**, *164*, 104191.
13. Kordani, M.; Bagheritabar, M.; Ahmadianfar, I.; Samadi-Koucheksaraee, A. Forecasting water quality indices using generalized ridge model, regularized weighted kernel ridge model, and optimized multivariate variational mode decomposition. *Scientific Reports* **2025**, *15*, 16313.
14. Kordani, M.; Nikoo, M.R.; Fooladi, M.; Ahmadianfar, I.; Nazari, R.; Gandomi, A.H. Improving long-term flood forecasting accuracy using ensemble deep learning models and an attention mechanism. *Journal of Hydrologic Engineering* **2024**, *29*, 04024042.
15. Nia, M.F.; Callen, G.E.; An, J.; Chandra, K.; Thompson, C.; Wolkowicz, K.; Denis, M. Experiential Learning for Interdisciplinary Education on Vestibular System Models. In Proceedings of the ASEE annual conference exposition, 2023.

16. Liu, Y.; Chen, W.; Arendt, P.; Huang, H.Z. Toward a better understanding of model validation metrics **2011**.
17. Chatfield, C. Model uncertainty, data mining and statistical inference. *Journal of the Royal Statistical Society Series A: Statistics in Society* **1995**, *158*, 419–444.
18. Das, A.K. *Research evaluation metrics*; Vol. 4, UNESCO Publishing, 2015.
19. Caldiera, V.R.B.G.; Rombach, H.D. The goal question metric approach. *Encyclopedia of software engineering* **1994**, pp. 528–532.
20. Ampel, B.; Yang, C.H.; Hu, J.; Chen, H. Large language models for conducting advanced text Analytics Information Systems Research. *ACM Transactions on Management Information Systems* **2024**.
21. Shamshiri, A.; Ryu, K.R.; Park, J.Y. Text mining and natural language processing in construction. *Automation in Construction* **2024**, *158*, 105200.
22. Kobayashi, V.B.; Mol, S.T.; Berkers, H.A.; Kismihók, G.; Den Hartog, D.N. Text mining in organizational research. *Organizational research methods* **2018**, *21*, 733–765.
23. Sfoq, M.S.; Albeer, R.A.; Abd, E.H.; et al. A Review Of Text Mining Techniques: Trends, and Applications In Various Domains. *Iraqi Journal For Computer Science and Mathematics* **2024**, *5*, 125–141.
24. Zhang, C.; Wang, X.; Cui, A.P.; Han, S. Linking big data analytical intelligence to customer relationship management performance. *Industrial Marketing Management* **2020**, *91*, 483–494.
25. Raheja, S.; Munjal, G. Text mining for secure cyber space. *Intelligent Data Analytics for Terror Threat Prediction: Architectures, Methodologies, Techniques and Applications* **2021**, pp. 95–118.
26. Ceballos Delgado, A.A.; Glisson, W.; Shashidhar, N.; Mcdonald, J.; Grispos, G.; Benton, R. Deception detection using machine learning **2021**.
27. Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C.H.; Kang, J. BioBERT: A pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* **2020**, *36*, 1234–1240.
28. Bhasuran, B. Combining Literature Mining and Machine Learning for Predicting Biomedical Discoveries. In *Biomedical Text Mining*; Springer, 2022; pp. 123–140.
29. Martinelli, G.; Molfese, F.; Tedeschi, S.; Fernández-Castro, A.; Navigli, R. CNER: Concept and Named Entity Recognition. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 8329–8344.
30. Majumdar, A.; Ajay, A.; Zhang, X.; Putta, P.; Yenamandra, S.; Henaff, M.; Silwal, S.; Mcvay, P.; Maksymets, O.; Arnaud, S.; et al. Openeqa: Embodied question answering in the era of foundation models. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 16488–16498.
31. Kouris, P.; Alexandridis, G.; Stafylopatis, A. Text summarization based on semantic graphs: An abstract meaning representation graph-to-text deep learning approach. *Journal of Big Data* **2024**, *11*, 95.
32. Ko, H.K.; Jeon, H.; Park, G.; Kim, D.H.; Kim, N.W.; Kim, J.; Seo, J. Natural language dataset generation framework for visualizations powered by large language models. In Proceedings of the Proceedings of the CHI Conference on Human Factors in Computing Systems, 2024, pp. 1–22.
33. Pérez, C.R. Re-thinking machine translation post-editing guidelines. *The Journal of Specialised Translation* **2024**, pp. 26–47.
34. Chan, K.; Yeung, P.s.; Chung, K.K.H. The effects of foreign language anxiety on English word reading among Chinese students at risk of English learning difficulties. *Reading and Writing* **2024**, pp. 1–19.
35. Zhu, Q.; Gao, L.; Qin, L. DFNet: Decoupled Fusion Network for Dialectal Speech Recognition. *Mathematics* **2024**, *12*, 1886.
36. Batra, P.; Phalnikar, N.; Kurmi, D.; Tembhurne, J.; Sahare, P.; Diwan, T. OCR-MRD: Performance analysis of different optical character recognition engines for medical report digitization. *International Journal of Information Technology* **2024**, *16*, 447–455.
37. Amadeus, M.; Castañeda, W.A.C. Evaluation Metrics for Text Data Augmentation in NLP. *arXiv preprint arXiv:2402.06766* **2024**.
38. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002, pp. 311–318.
39. McCoy, R.T.; Smolensky, P.; Linzen, T.; Gao, J.; Celikyilmaz, A. How much do language models copy from their training data? evaluating linguistic novelty in text generation using raven. *Transactions of the Association for Computational Linguistics* **2023**, *11*, 652–670.

40. Zhu, Y.; Lu, S.; Zheng, L.; Guo, J.; Zhang, W.; Wang, J.; Yu, Y. Texygen. In Proceedings of the The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. ACM, 2018.
41. Galley, M.; Brockett, C.; Sordoni, A.; Ji, Y.; Auli, M.; Quirk, C.; Mitchell, M.; Gao, J.; Dolan, B. deltaBLEU: A discriminative metric for generation tasks with intrinsically diverse targets. *arXiv preprint arXiv:1506.06863* **2015**.
42. Morris, D.; Pennell, A.M.; Perney, J.; Trathen, W. Using subjective and objective measures to predict level of reading fluency at the end of first grade. *Reading Psychology* **2018**, *39*, 253–270.
43. Kann, K.; Rothe, S.; Filippova, K. Sentence-level fluency evaluation: References help, but can be spared! *arXiv preprint arXiv:1809.08731* **2018**.
44. Wang, Y.; Deng, J.; Sun, A.; Meng, X. Perplexity from plm is unreliable for evaluating text quality. *arXiv preprint arXiv:2210.05892* **2022**.
45. Feng, S.Y.; Gangal, V.; Kang, D.; Mitamura, T.; Hovy, E. Genaug: Data augmentation for finetuning text generators. *arXiv preprint arXiv:2010.01794* **2020**.
46. Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K.Q.; Artzi, Y. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675* **2019**.
47. Popović, M. chrF++: Words helping character n-grams. In Proceedings of the Proceedings of the second conference on machine translation, 2017, pp. 612–618.
48. Chicco, D.; Warrens, M.J.; Jurman, G. The Matthews correlation coefficient (MCC) is more informative than Cohen's Kappa and Brier score in binary classification assessment. *Ieee Access* **2021**, *9*, 78368–78381.
49. Krstinić, D.; Braović, M.; Šerić, L.; Božić-Štulić, D. Multi-label classifier performance evaluation with confusion matrix. *Computer Science & Information Technology* **2020**, *1*, 1–14.
50. Beauxis-Aussalet, E.; Hardman, L. Simplifying the visualization of confusion matrix. In Proceedings of the 26th Benelux conference on artificial intelligence (BNAIC), 2014.
51. Ali Abd Al-Hameed, K. Spearman's correlation coefficient in statistical analysis. *International Journal of Nonlinear Analysis and Applications* **2022**, *13*, 3249–3255.
52. Xia, P.; Zhang, L.; Li, F. Learning similarity with cosine similarity ensemble. *Information sciences* **2015**, *307*, 39–52.
53. Ali, A.; Renals, S. Word error rate estimation for speech recognition: e-WER. In Proceedings of the Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2018, pp. 20–24.
54. Malik, M.; Malik, M.K.; Mehmood, K.; Makhdoom, I. Automatic speech recognition: A survey. *Multimedia Tools and Applications* **2021**, *80*, 9411–9457.
55. Ho, Y.; Wookey, S. The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling. *IEEE access* **2019**, *8*, 4806–4813.
56. Hossin, M.; Sulaiman, M.N. A review on evaluation metrics for data classification evaluations. *International journal of data mining & knowledge management process* **2015**, *5*, 1.
57. Feuerman, M.; Miller, A.R. Relationships between statistical measures of agreement: Sensitivity, specificity and kappa. *Journal of evaluation in clinical practice* **2008**, *14*, 930–933.
58. Yadla, H.K.; PVRDP, R. Machine learning based text classifier centered on TF-IDF vectoriser. *Int. J. Sci. Technol. Res* **2020**, *9*, 583.
59. Berbatova, M. Overview on NLP techniques for content-based recommender systems for books. In Proceedings of the Proceedings of the Student Research Workshop Associated with RANLP 2019, 2019, pp. 55–61.
60. Avazpour, I.; Pitakrat, T.; Grunske, L.; Grundy, J. Dimensions and metrics for evaluating recommendation systems. *Recommendation systems in software engineering* **2014**, pp. 245–273.
61. Guo, J.; Deng, J.; Ran, X.; Wang, Y.; Jin, H. An efficient and accurate recommendation strategy using degree classification criteria for item-based collaborative filtering. *Expert Systems with Applications* **2021**, *164*, 113756.
62. Sujatha, P.; Dhavachelvan, P. Precision at K in Multilingual Information Retrieval. *International Journal of Computer Applications* **2011**, *24*, 40–43. https://doi.org/10.5120/2990-3929.
63. Liu, X.; Liu, Y.; Aberer, K.; Miao, C. Personalized point-of-interest recommendation by mining users' preference transition. In Proceedings of the Proceedings of the 22nd ACM International Conference on Information & Knowledge Management, New York, NY, USA, 2013; CIKM '13, p. 733–738. https://doi.org/10.1145/2505515.2505639.
64. Jadon, A.; Patil, A. A comprehensive survey of evaluation techniques for recommendation systems. In Proceedings of the International Conference on Computation of Artificial Intelligence & Machine Learning. Springer, 2024, pp. 281–304.

65. Rahman, M.M.; Roy, C.K.; Lo, D. Rack: Automatic api recommendation using crowdsourced knowledge. In Proceedings of the 2016 IEEE 23rd International Conference on Software Analysis, Evolution, and Reengineering (SANER). IEEE, 2016, Vol. 1, pp. 349–359.
66. Smyth, B.; McClave, P. Similarity vs. diversity. In Proceedings of the International conference on case-based reasoning. Springer, 2001, pp. 347–361.
67. Hamerly, G.; Elkan, C. Learning the k in k-means. *Advances in neural information processing systems* **2003**, *16*.
68. Chavan, M.; Patil, A.; Dalvi, L.; Patil, A. Mini batch K-Means clustering on large dataset. *Int. J. Sci. Eng. Technol. Res* **2015**, *4*, 1356–1358.
69. Ackermann, M.R.; Blömer, J.; Kuntze, D.; Sohler, C. Analysis of agglomerative clustering. *Algorithmica* **2014**, *69*, 184–215.
70. Zhang, T.; Ramakrishnan, R.; Livny, M. BIRCH: An efficient data clustering method for very large databases. *ACM sigmod record* **1996**, *25*, 103–114.
71. Deng, D. DBSCAN clustering algorithm based on density. In Proceedings of the 2020 7th international forum on electrical engineering and automation (IFEEA). IEEE, 2020, pp. 949–953.
72. Ankerst, M.; Breunig, M.M.; Kriegel, H.P.; Sander, J. OPTICS: Ordering points to identify the clustering structure. *ACM Sigmod record* **1999**, *28*, 49–60.
73. Dueck, D. *Affinity propagation: Clustering data by passing messages*; University of Toronto Toronto, ON, Canada, 2009.
74. Derpanis, K.G. Mean shift clustering. *Lecture Notes* **2005**, *32*, 16.
75. Ouyang, M.; Welsh, W.J.; Georgopoulos, P. Gaussian mixture clustering and imputation of microarray data. *Bioinformatics* **2004**, *20*, 917–923.
76. Von Luxburg, U. A tutorial on spectral clustering. *Statistics and computing* **2007**, *17*, 395–416.
77. Xie, J.; Girshick, R.; Farhadi, A. Unsupervised deep embedding for clustering analysis. In Proceedings of the International conference on machine learning. PMLR, 2016, pp. 478–487.
78. Chang, J.; Wang, L.; Meng, G.; Xiang, S.; Pan, C. Deep adaptive image clustering. In Proceedings of the Proceedings of the IEEE international conference on computer vision, 2017, pp. 5879–5887.
79. Hu, W.; Miyato, T.; Tokui, S.; Matsumoto, E.; Sugiyama, M. Learning discrete representations via information maximizing self-augmented training. In Proceedings of the International conference on machine learning. PMLR, 2017, pp. 1558–1567.
80. Jiang, Z.; Zheng, Y.; Tan, H.; Tang, B.; Zhou, H. Variational deep embedding: An unsupervised and generative approach to clustering. *arXiv preprint arXiv:1611.05148* **2016**.
81. Golzari Oskouei, A.; Balafar, M.A.; Motamed, C. EDCWRN: Efficient deep clustering with the weight of representations and the help of neighbors. *Applied Intelligence* **2023**, *53*, 5845–5867.
82. Arbelaitz, O.; Gurrutxaga, I.; Muguerza, J.; Pérez, J.M.; Perona, I. An extensive comparative study of cluster validity indices. *Pattern recognition* **2013**, *46*, 243–256.
83. Dunn, J.C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters **1973**.
84. Caliński, T.; Harabasz, J. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods* **1974**, *3*, 1–27.
85. Davies, D.L.; Bouldin, D.W. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* **1979**, pp. 224–227.
86. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* **1987**, *20*, 53–65.
87. Baker, F.B.; Hubert, L.J. Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association* **1975**, *70*, 31–38.
88. Hubert, L.J.; Levin, J.R. A general statistical framework for assessing categorical clustering in free recall. *Psychological bulletin* **1976**, *83*, 1072.
89. Halkidi, M.; Vazirgiannis, M. Clustering validity assessment: Finding the optimal partitioning of a data set. In Proceedings of the Proceedings 2001 IEEE international conference on data mining. IEEE, 2001, pp. 187–194.
90. Chou, C.H.; Su, M.C.; Lai, E. A new cluster validity measure and its application to image compression. *Pattern Analysis and Applications* **2004**, *7*, 205–220.
91. Bandyopadhyay, S.; Saha, S. A point symmetry-based clustering technique for automatic evolution of clusters. *IEEE Transactions on Knowledge and Data Engineering* **2008**, *20*, 1441–1457.

92. Gurrutxaga, I.; Albisua, I.; Arbelaitz, O.; Martín, J.I.; Muguerza, J.; Pérez, J.M.; Perona, I. SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index. *Pattern Recognition* **2010**, *43*, 3364–3373.
93. Lago-Fernández, L.F.; Corbacho, F. Normality-based validation for crisp clustering. *Pattern Recognition* **2010**, *43*, 782–795.
94. Saitta, S.; Raphael, B.; Smith, I.F. A bounded index for cluster validity. In Proceedings of the Machine Learning and Data Mining in Pattern Recognition: 5th International Conference, MLDM 2007, Leipzig, Germany, July 18-20, 2007. Proceedings 5. Springer, 2007, pp. 174–187.
95. Bezdek, J.C.; Pal, N.R. Some new indexes of cluster validity. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* **1998**, *28*, 301–315.
96. Pal, N.R.; Biswas, J. Cluster validation using graph theoretic concepts. *Pattern Recognition* **1997**, *30*, 847–857.
97. Saha, S.; Bandyopadhyay, S. Performance evaluation of some symmetry-based cluster validity indexes. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* **2009**, *39*, 420–425.
98. Kim, M.; Ramakrishna, R. New indices for cluster validity assessment. *Pattern Recognition Letters* **2005**, *26*, 2353–2363.
99. Žalik, K.R.; Žalik, B. Validity index for clusters of different sizes and densities. *Pattern Recognition Letters* **2011**, *32*, 221–234.
100. Palacio-Niño, J.O.; Berzal, F. Evaluation metrics for unsupervised learning algorithms. *arXiv preprint arXiv:1905.05667* **2019**.
101. Tan, P.N.; Steinbach, M.; Kumar, V. Introduction to data mining, Pearson education. *Inc., New Delhi* **2006**.
102. Xie, X.L.; Beni, G. A validity measure for fuzzy clustering. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **1991**, *13*, 841–847.
103. Ball, G.H. ISODATA, a novel method of data analysis and pattern classification. *stanford research institute* **1965**, pp. AD–699616.
104. Hartigan, J.A. Distribution problems in clustering. In *Classification and clustering*; Elsevier, 1977; pp. 45–71.
105. Xu, L. Bayesian Ying–Yang machine, clustering and number of clusters. *Pattern Recognition Letters* **1997**, *18*, 1167–1178.
106. Chavent, M.; Brito, P. Divisive clustering of histogram data. In *Analysis of Distributional Data*; Chapman and Hall/CRC, 2022; pp. 127–138.
107. Gan, G.; Ma, C.; Wu, J. *Data clustering: Theory, algorithms, and applications*; SIAM, 2020.
108. Xiong, H.; Li, Z. Clustering validation measures. In *Data clustering*; Chapman and Hall/CRC, 2018; pp. 571–606.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.