DOCTORADO EN CIENCIAS EN CIENCIA DE DATOS

INFOTEC CENTRO DE INVESTIGACIÓN E
INNOVACIÓN EN TECNOLOGÍAS DE LA
INFORMACIÓN Y COMUNICACIÓN

DIRECCIÓN ADJUNTA DE INNOVACIÓN Y
CONOCIMIENTO
GERENCIA DE CAPITAL HUMANO
POSGRADOS

Analysis of Systems' Performance in Supervised Learning Challenges

Tesis
Que para obtener el grado de DOCTOR EN
CIENCIAS EN CIENCIA DE DATOS

Presenta:

Sergio Martín Nava Muñoz

Asesor:

Mario Graff Guerrero

Aguascalientes, Octubre 2024

CONACYT

INFOTEC
POSGRADOS

Autorización de impresión

Resumen

El rápido crecimiento del aprendizaje automático en los últimos años ha llevado al desarrollo de numerosos algoritmos y modelos que requieren métodos de evaluación efectivos. Las competiciones entre algoritmos, que comparan el rendimiento de los modelos en condiciones similares, se han convertido en una herramienta esencial en este proceso. Sin embargo, la mayoría de la literatura existente se centra en métodos estadísticos y métricas tradicionales para la comparación de algoritmos, sin tener en cuenta los aspectos únicos de los entornos competitivos.

Esta tesis introduce nuevas herramientas y metodologías diseñadas para abordar los desafíos específicos que plantean las competiciones algorítmicas. Los enfoques propuestos tienen como objetivo facilitar la comparación justa y precisa de algoritmos en competencia, considerando las dinámicas particulares de los entornos competitivos. A través de una revisión exhaustiva de los métodos existentes, mejoras a las técnicas actuales y la introducción de herramientas novedosas, esta investigación contribuye al desarrollo de un marco más robusto para la evaluación de algoritmos en competiciones. Se espera que los resultados mejoren el proceso de evaluación, haciéndolo más adaptable y preciso en escenarios impulsados por la competencia.

Abstract

The rapid growth of machine learning in recent years has led to the development of numerous algorithms and models that require effective evaluation methods. Competitions between algorithms, which compare the performance of models under similar conditions, have become an essential tool in this process. However, most existing literature focuses on traditional statistical methods and metrics for algorithm comparison without considering the unique aspects of competitive environments.

This thesis introduces new tools and methodologies designed to address the specific challenges posed by algorithmic competitions. The proposed approaches aim to facilitate the fair and accurate comparison of competing algorithms, considering the particular dynamics of competitive settings. Through a comprehensive review of existing methods, improvements to current techniques, and the introduction of novel tools, this research contributes to developing a more robust framework for evaluating algorithms in competitions. The results are expected to enhance the evaluation process, making it more adaptable and precise for competition-driven scenarios.

Acknowledgements

I extend my deepest gratitude to my Ph.D. advisor, *Ph.D. Mario Graff Guerrero*, for his unwavering guidance, insightful advice, and continuous support throughout my research journey.

Special thanks to *Ph.D. Hugo Jair Escalante* for his significant contributions to the development and writing of the papers. His leadership and recognition in the field were crucial in securing the datasets from the analyzed competitions, enabling the successful validation of the proposed methodologies.

We sincerely thank the organizers of the competitions analyzed in this work. Their provision of the datasets was instrumental in enabling the validation and testing of the proposed framework. These datasets formed the foundation for rigorous analysis and evaluation, making it possible to demonstrate the effectiveness of our methodologies. We appreciate their support and commitment to advancing research by creating valuable resources.

Contents

List of Figures

List of Tables

List of Algorithms

Acronyms and Abbreviations

ASR	Automatic Speech Recognition
AUC	Area Under the ROC Curve
BLUE	Bilingual evaluation understudy
FDR	False Discovery Rate
FWER	Familywise Error Rate
LOOCV	Leave-one-out cross-validation
MCMC	Markov Chain Monte Carlo
ML	Machine Learning
MT	Machine Translation
NIST	National Institute of Standard and Technology.
NLP	Natural Language Processing
SVM	Support Vector Machine

Glossary

Algorithmic Competition A structured event where algorithms or models are evaluated and compared based on their performance on a defined task under specific conditions.

Benchmarking The process of comparing systems, algorithms, or methods against a standard or a set of metrics to evaluate their performance.

Bootstrapping A statistical resampling technique that involves repeatedly sampling with replacement from a dataset to estimate the distribution of a statistic.

Classification A supervised learning task where the goal is to assign a label or category to input data based on learned patterns.

Confidence Interval A range of values, derived from data, that is believed to contain the true value of a parameter with a specified level of confidence.

Cross-Validation A resampling method used to evaluate the generalizability of a model by dividing data into training and testing sets multiple times.

Crowdsourcing the practice of engaging a group of people for a common goal, often involving innovation, problem-solving, or efficiency.

Evaluation Metric A quantitative measure used to assess the performance of an algorithm, model, or system. Examples include precision, recall, and F1-score.

Familywise Error Rate (FWER) The probability of making one or more Type I errors (false positives) when conducting multiple statistical tests.

Gold Standard The authoritative or best-known standard used as a benchmark for comparison in experiments or evaluations.

Machine Learning (ML) A field of artificial intelligence that focuses on developing algorithms that can learn from and make predictions based on data.

Permutation Test A non-parametric statistical test that involves rearranging the data to assess the significance of a result without relying on specific distributional assumptions.

Resampling Techniques Methods used to assess model performance or variability by repeatedly drawing samples from a dataset. Examples include bootstrapping and cross-validation.

Statistical Significance A determination that an observed effect or relationship in data is unlikely to have occurred by chance, according to a predefined significance level.

Supervised Learning A type of machine learning where models are trained on labeled data to predict outcomes for new, unseen inputs.

Type I Error The incorrect rejection of a true null hypothesis, also known as a false positive.

Type II Error The failure to reject a false null hypothesis, also known as a false negative.

Unsupervised Learning A type of machine learning where models identify patterns or structures in data without predefined labels or categories.

Introduction

Context and Background

Building upon the concepts of Crowdsourcing and academic competitions, the next chapter delves into the methodologies and structures that define modern challenges. Exploring the critical aspects of problem definition, dataset selection, and evaluation metrics provides a comprehensive framework for organizing and analyzing competitions. This chapter highlights the significance of fair and robust evaluations, essential for fostering innovation and advancing research in competitive environments.

Crowdsourcing and Challenges

Crowdsourcing, popularized by Jeff Howe in 2006, involves leveraging a geographically dispersed group to achieve common goals such as innovation, problem-solving, or efficiency. While historically rooted, it has gained prominence in academia, business, and humanitarian efforts.

Key examples include Wikipedia for collaborative content creation, Amazon Mechanical Turk for micro-tasks, and the Galaxy Zoo project, which accelerates scientific research through public participation [14, 39, 46].

Benefits include cost efficiency, speed, and access to diverse expertise, making it invaluable for tasks requiring scalability and creativity. However, ensuring quality, coordinating contributors, and addressing ethical concerns like fair compensation and privacy remain critical [32, 62].

Crowdsourcing in Research

Academic competitions refer to contests or challenges where students or researchers compete with each other to solve problems or develop innovative projects under certain constraints [1, 53]. These competitions can be driven or enriched by the use of Crowdsourcing strategies combined with Benchmarking, where collaboration and competition intertwine to foster innovation and academic excellence. These competitions have become benchmarks that continuously push the state of the art in science and technology, such as the ImageNet [59] and VOC competitions [27].

Academic competitions, also known as challenges, are structured contests designed to address specific scientific or technological problems by leveraging the collective expertise and innovative capabilities of a global community of researchers and practitioners. These competitions provide a platform for participants to develop, test, and benchmark their solutions in a competitive environment, fostering rapid advancements and collaborative problem-solving [26, 39].

One of the most notable areas where academic competitions have had a profound impact is machine learning. These competitions have catalyzed significant breakthroughs by providing standardized datasets and evaluation metrics, thereby enabling the rigorous comparison of algorithms and techniques. Platforms such as Kaggle, the Conference on Neural Information Processing Systems (NeurIPS) competition track, and the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) have been instrumental in this regard [14, 59].

Machine learning competitions serve as a crucial driver for innovation and performance improvements. They typically involve tasks such as Image Classification, natural language processing, and predictive modeling, each associated with specific datasets and performance benchmarks. The competitive nature of these events encourages participants to push the boundaries of existing methodologies and develop novel approaches.

The NeurIPS competition track, for example, has hosted a variety of challenges that have led to advancements in areas like deep reinforcement learning, automated machine learning (AutoML), and natural language understanding. The ImageNet competition, in particular, played a significant role in developing and popularizing deep learning techniques, especially convolutional neural networks (CNNs) [42].

These competitions not only advance the state of the art but also promote reproducibility and transparency in research. Providing access to standardized datasets and clearly defined evaluation criteria ensures that results are comparable and verifiable, which is essential for scientific progress [36].

Structure and Methodology of Challenges

Challenges are organized events that bring together participants to solve specific problems or achieve predefined goals within a given timeframe. These events are characterized by their structured approach, which typically includes a clear problem statement, standardized datasets, evaluation metrics, and a framework for submission and assessment [26, 39].

Structure

The structure of challenges generally involves several key components. Firstly, the **problem definition** outlines the task to be solved. This is followed by the provision of **datasets**, which participants use to develop and test their solutions. The datasets are often divided into training and testing sets for unbiased evaluation. Next, **Evaluation Metrics** are defined to measure the submitted solutions' performance objectively. These metrics vary depending on the nature of the problem but commonly include accuracy, precision, recall, and F1-score (See appendix A.1) for Classification tasks.

Methodology

A classical challenge scheme involves several critical steps. Figure 1 illustrates the challenge scheme, showcasing the flow of processes in a typical competition framework. From left to right, the flow moves from the reality or problem to solve, obtaining a reality sample until competitors receive the training and validation data (with labels) and the test data (without labels). From right to left, the path follows the predictions submitted by the competitors for evaluation, where organizers compare them with the Gold Standard. This systematic flow ensures fairness and rigor in assessing algorithmic performance.

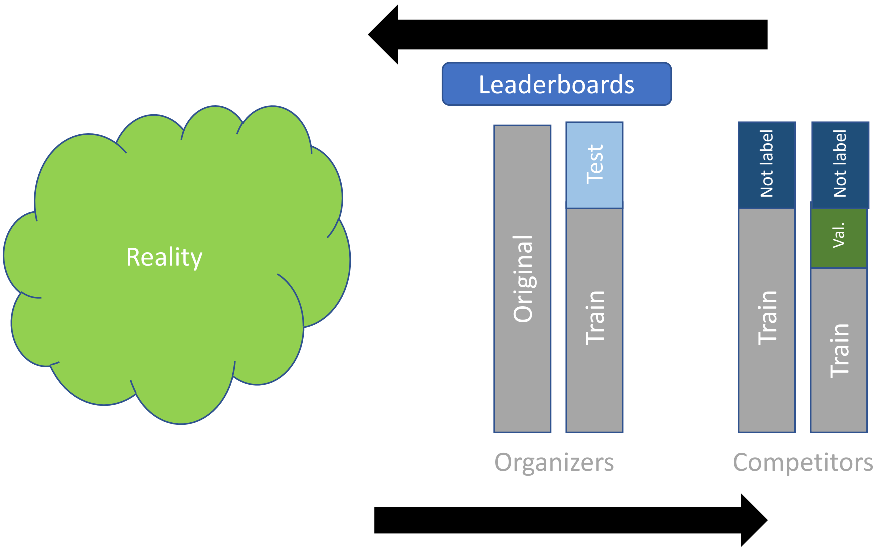


Figure 1: Challenge scheme. From left to right, the flow moves from the desired reality, obtaining a reality sample until competitors receive the training and validation data (with labels) and the test data (without labels). From right to left, the path follows the predictions for evaluation, where organizers compare them with the Gold Standard.

Initially, **designing the challenge** requires defining the problem scope, selecting appropriate datasets, and establishing evaluation criteria. This step is crucial as it sets the foundation for the entire competition [26, 36]. During this phase, organizers must clearly define the objectives and expected outcomes, ensuring that the challenge aligns with the overarching goals of the field. Choosing datasets representative of real-world problems and suitable for the tasks at hand is also essential. Evaluation criteria must be carefully crafted to accurately measure the desired performance aspects. The dataset is typically divided into 80% for the training set provided to participants and 20% for the evaluation set used to assess the submissions [12].

Following the design phase, **launching the challenge** involves disseminating information to potential participants and providing access to datasets and submission platforms. Online platforms such as Kaggle [1] or Codalab [53] often support this phase, which offers tools for managing submissions, hosting leaderboards, and facilitating participant interaction [14]. Effective communication strategies are essential to attract diverse participants, including social media outreach, email campaigns, and collaborations with academic and industry partners. Clear guidelines and robust technical support can enhance participant engagement and ensure a smooth launch.

Evaluation and scoring are performed based on the predefined metrics. Automated systems are commonly used to ensure fairness and consistency in assessment. This phase may involve multiple rounds of evaluation, where preliminary results are provided to participants for feedback and iteration. Ensuring transparency in the evaluation process is key to maintaining participant trust and competition integrity. Participants typically divide the provided training set into two subsets: 80% for training their models and 20% for validating their models before making final submissions.

Finally, the **results dissemination** phase involves announcing winners, publishing results, and often organizing workshops or conferences to discuss findings and future directions [46]. This phase is crucial for showcasing the achievements of participants, providing recognition, and fostering a community of practice. Detailed reports and publications can offer valuable insights into the challenge outcomes and contribute to the broader knowledge base of the field. Post-challenge activities, such as follow-up projects and collaborative initiatives, can further leverage the momentum generated by the competition.

Dataset

Selecting datasets for a challenge involves considering several key characteristics to ensure the competition's effectiveness, fairness, and relevance. The most important characteristics are detailed below [25, 26]:

1. **Representativity**: The dataset must represent the problem being addressed. This means the data should cover various cases and situations the algorithms might encounter in real-world applications.

2. **Adequate Size**: The dataset size should be sufficient to provide a meaningful evaluation of the algorithm's performance but not so large that it becomes an obstacle due to the computational resource limitations of the participants.

3. **Data Quality**: The quality of the dataset is crucial. The data should be clean,

well-annotated, and with minimal noise. Errors and inconsistencies should be minimized to avoid biasing the evaluation results.

4. **Accessibility**: The dataset should be accessible to all participants. This includes ensuring that the data is not proprietary or restricted and that all participants can download and use the dataset without legal or technical issues.

5. **Labeling and Annotation**: The data should be correctly labeled and annotated if the challenge involves Classification or labeling. The labels should be accurate and faithfully reflect the information contained in the data.

6. **Fairness**: The dataset must be impartial and should not favor any particular algorithm. This is achieved through careful design and pilot testing to identify and correct potential biases.

7. **Benchmarking**: Using datasets that are industry standards or have been used in previous challenges can help ensure the results are comparable and validated against prior work.

These characteristics help ensure that the selected dataset allows for a fair and meaningful evaluation of algorithm performance, facilitating the identification of effective and robust solutions in the context of the challenge.

Challenges Summary

A challenge involves comparing the performance of algorithms under certain constraints. The process can be detailed as follows:

- **Evaluation of Multiple Participants**: The challenge involves evaluating various participants, which can be different algorithms, methods, or systems. This ensures a broad comparison across multiple approaches to solve the given problem.

- **Performance Metrics Selection**: The organizers select the performance metrics. These metrics are crucial as they define how each participant's performance will be measured and compared. Common metrics include accuracy, precision, and recall, among others.

- **Fixed Dataset Size**: The challenge uses a fixed dataset size. This constraint ensures that all participants work with the same data volume, promoting fairness and consistency in the evaluation process.
- **Limitation on Submissions**: There is a limit on the number of submissions each participant can make. This constraint encourages participants to carefully optimize and test their solutions before submission, as they have a finite number of attempts to achieve the best possible performance.

Traditional statistical methods for inferring the significance of a particular performance metric are difficult to apply in the absence of multiple datasets or submissions. This research shows how organizers can make effective comparisons under these constraints.

Research Problem

In recent decades, collaborative challenges in science and technology have become a popular platform for evaluating and improving research methodologies through competitions. These competitions use scoring and ranking systems to compare participants' solutions. However, these systems face significant limitations, especially in tasks such as comparing the performance of Classification algorithms.

The main challenge lies in the results' validity and Statistical Significance. In many competitions, as described in the case of evaluating Classification algorithms, fixed performance metrics, a constant dataset size, and a limited number of submissions per participant are used. **These constraints make it difficult to apply classical statistics to infer the significance of performance metrics.** Traditionally, the selection of the winner is based solely on the score ranking, without considering the possibility that performance differences may result from random variability rather than genuine methodological differences.

Furthermore, the absence of multiple datasets or numerous submissions prevents effective and rigorous multiple comparisons, which can result in a misinterpretation of one method’s superiority over another. This situation poses a critical problem: the lack of robust and accessible statistical tools that allow precise and fair performance evaluation in these competitions.

Objectives

General Objective

To investigate and implement an evaluation scheme based on robust statistical methods for learning algorithms in the context of challenges, providing organizers and researchers with concrete and validated tools for more precise and well-founded decision-making.

Specific Objectives

1. **Thoroughly review the different elements used in evaluating learning algorithms**, including performance metrics, statistical methods used to infer significance, and test configurations. This review will focus on identifying the most common practices and their limitations to establish a theoretical framework that supports the design of new evaluation schemes.

2. **Examine in detail the characteristics and criteria used by the main Machine Learning competitions**, defining as main those with a high impact on the scientific community or pioneers in applying new evaluation methodologies. This objective will include an analysis of result validation processes, the transparency and fairness of evaluations, and the methodologies used for data handling and distribution.

3. **Design evaluation schemes for learning algorithms that integrate advanced statistical tools** based on the deficiencies identified in the previous objectives. This design will include prototype schemes evaluated through simulations or collaboration with existing competition organizers to validate their effectiveness

and practicality. Clear guidelines for their implementation in different competition contexts will also be developed.

Research Hypotheses and Questions

Hypotheses

- **Performance Metrics Hypothesis:** The use of advanced statistical methods, such as Bootstrapping, improves the reliability of performance evaluations in machine learning competitions compared to traditional methods.
- **Fairness and Robustness Hypothesis:** Competitions leveraging resampling techniques and robust evaluation frameworks provide fairer and more accurate participant rankings than those using basic metrics alone.
- **Applicability Hypothesis:** Statistical tools and methodologies designed for Classification tasks can be effectively adapted to other competition types, such as regression or clustering, ensuring robust evaluation across domains.

Research Questions

- **Evaluation Framework Question:** How can robust statistical methods enhance the evaluation of learning algorithms in competitive environments?
- **Method Comparison Question:** What are the trade-offs between traditional statistical evaluation methods and newer approaches like Bootstrapping and permutation tests in the context of Algorithmic Competitions?
- **Impact of Constraints Question:** How do constraints such as fixed dataset sizes and limited submissions impact the reliability of performance evaluations in competitive scenarios?
- **Generalization Question:** Can the proposed statistical methods reliably predict the performance of algorithms in unseen datasets or real-world applications?

Summary of the Work and Main Findings

This thesis addresses the challenges present in Algorithmic Competitions, proposing novel methodologies and tools for accurately comparing algorithms under competitive conditions. Through an exhaustive review of current evaluation methods, this work identifies limitations in traditional approaches when applied to competition settings. The methods proposed in this research consider the unique dynamics of competitive scenarios, allowing for fair and robust algorithm comparisons.

The results obtained in this thesis highlight the advantages of the proposed methodologies, demonstrating significant improvements in the accuracy and adaptability of algorithm performance evaluation in competitions. These findings contribute to developing a more solid framework for competition evaluation, with positive implications for future research and the design of competitions in data science and machine learning.

Outline of the Thesis

This thesis is structured into the following chapters:

- **Chapter 1: Related Work**
  This chapter presents a thorough review of the literature on Algorithmic Competitions and machine learning, focusing on statistical methods, evaluation metrics, and frameworks tailored to competition scenarios. Key limitations in traditional approaches are identified, laying the groundwork for the contributions made in this thesis.

- **Chapter 2: Theory Framework**
  This chapter establishes the theoretical basis for this research, covering machine learning fundamentals, supervised learning approaches, and critical statistical techniques such as resampling and Bootstrapping. These methods provide the foundation for comparing algorithm performance in competitive and empirical research settings.

- **Chapter 3: Implementation of the Proposed Evaluation Framework**

  In this chapter, the implementation details of the proposed evaluation framework are outlined, with a focus on statistical methods such as Bootstrapping, hypothesis testing, and multiple comparisons. The chapter emphasizes how these techniques are applied to ensure fair and robust algorithm evaluation in competitive challenges.

- **Chapter 4: Performance Comparison in Challenge Schemes**

  This chapter explores the methodologies for comparing algorithm performance within challenge settings. It addresses challenges such as limited datasets, competition-specific constraints, and performance variability. A detailed analysis of statistical testing and inference methods is provided, highlighting their role in generating reliable results.

- **Chapter 5: Comparison of Competitions**

  This chapter compares various Algorithmic Competitions, examining their structures, objectives, and outcomes. It highlights the differences in evaluation frameworks across fields and their impact on fostering innovation and collaboration. Suggestions for improving competition design are also provided, based on observed patterns and results.

Chapter 1

Related Work

1 Related Work

Chapter 1 provides a comprehensive review of the literature relevant to the thesis topic. In the rapidly evolving field of machine learning and Algorithmic Competitions, various methods have been proposed to assess the performance and significance of models in different settings. This chapter aims to synthesize key contributions in the literature, particularly focusing on methods for evaluating Classification algorithms, statistical tests used for comparison, and approaches specifically designed for competition scenarios. By understanding these foundational works, we can better position the tools and methodologies introduced in the subsequent chapters.

1.1 Introduction

The Thesis aim is to introduce tools that facilitate the comparison of results among different competitors. The existing literature addresses the issue of comparing Classification algorithms; however, these primarily address aspects other than the competition framework. This literature review groups the summarized articles into three primary themes: statistical tests and methods, evaluation metrics and frameworks, and specific approaches for competition scenarios. Each theme addresses distinct aspects of algorithm comparison and provides insights into different methodologies and their applications.

1.2 Statistical Tests and Methods

The primary goal of statistical tests is to assess whether one algorithm significantly outperforms another using statistical tests; in this section, we mainly focus on tests to compare Classification algorithms.

Dietterich [21] reviews various statistical tests to determine algorithm performance differences. The paper discusses five closely related statistical tests for assessing whether one learning algorithm outperforms another in specific learning tasks. These tests include the *Resampled Paired t-Test*, the *K-Fold Cross-Validated t-Test*, and the 5 × 2 *Cross-Validated Paired t-Test*. However, these tests require access to the underlying algorithm because they repeatedly split the dataset for training and prediction (depending on the test) to assess algorithm variability. In a competition scenario, there is only access to the predictions, not the algorithms. This limitation makes it challenging to apply these tests directly in competitive environments where only prediction results are available.

The paired t-test is a simple method that compares the means of two related groups to determine if there is a statistically significant difference between them. However, it assumes normal distribution and equal variances, which may not always hold in practice. The cross-validated t-test, on the other hand, is an extension that uses Cross-Validation to reduce the variance of the test, making it more robust to violations of these assumptions. McNemar's test is a non-parametric method used on paired nominal data, focusing on the differences between matched pairs, thus providing a more flexible approach for binary Classification problems.

Despite their strengths, these tests are limited by the need for access to the internal mechanics of the algorithms. Access to the algorithm is necessary for conducting these tests because the algorithms need to be run on various training and test data subsets to assess performance variability accurately. This helps capture the variability in error rates across different data samples, which is essential for valid statistical comparisons. In competitive scenarios, where only prediction results are available, alternative approaches are needed, as the absence of algorithm access hinders the application of these statistical tests effectively. Therefore, while Dietterich's work provides a solid foundation for algorithm comparison, it underscores the necessity for developing methods that can operate effectively under the constraints of competition frameworks.

Demšar [20] focuses on the statistical comparisons of classifiers across multiple datasets, which is unusual in challenge settings. The scenario involves just one dataset. Demšar presents several non-parametric methods and guidelines for conducting a proper analysis when comparing sets of classifiers. The study extends the discussion by introducing non-parametric methods for comparing classifiers across multiple datasets. Techniques such as the Friedman and Nemenyi post-hoc tests are highlighted for their robustness and ability to handle multiple comparisons without relying on parametric assumptions. These methods provide a more flexible framework for comparing classifier performance in varied scenarios.

The Friedman test is a non-parametric alternative to the repeated measures ANOVA. It ranks the algorithms for each dataset and then evaluates if there is a significant difference in the ranks. The Nemenyi post-hoc test, applied after a significant Friedman test, identifies which classifiers differ significantly. These methods are particularly useful in machine learning contexts where the normality assumptions of parametric tests are often violated.

Demšar's work also includes practical guidelines for applying these tests, emphasizing the importance of understanding the underlying assumptions and potential pitfalls. For instance, the paper discusses the issue of multiple comparisons and the increased risk of Type I errors, proposing adjustments to control the family-wise error rate. The comprehensive nature of this study makes it a valuable resource for researchers looking to perform rigorous and reliable comparisons of machine learning algorithms across diverse datasets.

García et al. [30] address a problem similar to Demšar's but focus on pairwise comparisons, specifically statistical procedures for comparing $c \times c$ classifiers. While their approach also involves scenarios with multiple datasets using the same classifiers, they emphasize pairwise comparison procedures in such contexts. The authors propose using the Wilcoxon signed-rank test and other non-parametric tests to compare classifiers, arguing that these methods offer greater reliability and interpretability when dealing with multiple datasets and classifiers.

The Wilcoxon signed-rank test is a non-parametric test that compares two related samples, matched samples, or repeated measurements on a single sample to assess whether their population mean ranks differ. It is used as an alternative to the paired t-test when the data cannot be assumed to be normally distributed. Garcia et al.

advocate for its use due to its robustness and simplicity.

Garcia et al. also discuss the practical application of these tests, including considerations for effect size and statistical power. They emphasize the need for careful experimental design to ensure meaningful and generalizable results. For example, they recommend using multiple datasets to capture the variability in algorithm performance and to avoid overfitting to specific data characteristics. This approach provides a more comprehensive assessment of classifier performance and helps identify the most robust algorithms across different contexts.

Overall, Garcia et al.'s work provides a detailed and practical guide for researchers looking to perform pairwise comparisons of classifiers, highlighting the advantages and limitations of various non-parametric methods.

1.3 Evaluation Metrics and Frameworks

Lavesson and Davidsson [45] expand on evaluating learning algorithms and classifiers by highlighting the importance of evaluation metrics in understanding model performance. They emphasize the necessity of using a variety of metrics to capture different aspects of performance, especially in prediction-only frameworks. This is crucial for ensuring that the evaluation is comprehensive and reflects the true capabilities of the models. The authors discuss metrics such as accuracy, precision, recall, F-measure, and area under the ROC curve (AUC), advocating for a multi-metric evaluation approach to better capture model performance nuances.

While widely used, accuracy often provides an incomplete picture, especially in imbalanced datasets where it might be misleading. Precision and recall provide more insight by measuring the relevance of the predictions. Precision indicates the number of true positive results divided by all positive results, while recall measures the number of true positive results divided by the number of positives that should have been retrieved.

The F-measure, the harmonic mean of precision and recall, offers a single metric that balances both concerns. The area under the ROC curve (AUC) is another important metric providing an aggregate performance measure across all Classification thresholds. It plots the true positive rate against the false positive rate, offering a comprehensive view of a model's performance.

Lavesson and Davidsson also discuss the importance of context when selecting evaluation metrics. Different applications may prioritize different aspects of performance, and a single metric may not capture all relevant dimensions. For example, recall (sensitivity) might be more critical than precision in medical diagnostics, as missing a positive case could have serious consequences. Conversely, precision might be more important in spam detection to avoid false positives.

The authors argue for a holistic approach to model evaluation, using a suite of metrics to provide a more nuanced and complete understanding of model performance. This approach can help ensure that models are evaluated fairly and thoroughly, particularly in prediction-only frameworks where access to the underlying algorithms is restricted.

Olson et al. [52] provide an overview of benchmarking in machine learning, emphasizing the importance of proper experimental design and statistical analysis. The authors propose a framework for fair algorithm comparison but assume access to the algorithms rather than just their predictions. This study highlights the need for proper experimental design and statistical analysis in benchmarking machine learning algorithms. The proposed framework includes recommendations on dataset selection, Cross-Validation strategies, and statistical tests to ensure fair and reliable comparisons. The authors stress the importance of transparency and reproducibility in benchmarking studies to foster trust and validity in the results.

The paper discusses the importance of selecting representative datasets that cover a wide range of problem domains. This helps ensure the benchmarking results are generalizable and not overly specific to a particular data type. Cross-Validation strategies, such as k-fold Cross-Validation, are recommended to provide a robust estimate of model performance. This approach divides the data into k subsets and iteratively uses one subset for testing while training on the remaining $k-1$ subsets.

Olson et al. also emphasize the need for rigorous statistical tests to compare the performance of different algorithms. They recommend using methods such as the paired t-test and Wilcoxon signed-rank test to assess whether observed differences in performance are statistically significant. These tests help ensure that the conclusions drawn from benchmarking studies are robust and not due to random variation.

Transparency and reproducibility are critical themes throughout the paper. The authors argue that all aspects of the benchmarking process, from dataset selection to evaluation metrics, should be documented and made publicly available. This allows other researchers to replicate the studies and verify the results, which is crucial for building trust in the findings.

The proposed framework by Olson et al. provides a comprehensive and systematic approach to benchmarking machine learning algorithms. By following these guidelines, researchers can ensure that their comparisons are fair, reliable, and transparent, advancing the field of machine learning.

Raschka [57] delves into model evaluation, model selection, and algorithm selection in machine learning, providing insights into various evaluation metrics and resampling methods. This comprehensive study enhances the understanding of algorithm performance evaluation. Raschka covers many evaluation techniques, including k-fold Cross-Validation, bootstrap resampling, and leave-one-out Cross-Validation. The author also discusses the trade-offs between different evaluation methods and the impact of data distribution on model performance. This extensive review is a valuable resource for researchers seeking to understand the intricacies of model evaluation and selection.

K-fold Cross-Validation provides a robust estimate of model performance by averaging the results across all folds. On the other hand, Bootstrap resampling involves repeatedly sampling with replacement from the original dataset to create multiple training sets. This method allows for the estimation of the variability of model performance and is particularly useful for small datasets.

Leave-one-out Cross-Validation (LOOCV) is the extreme form of k-fold

Cross-Validation where k equals the number of data points in the dataset. This method is computationally intensive but provides an unbiased estimate of model performance. Raschka discusses the trade-offs between these methods, highlighting the balance between computational cost and the accuracy of performance estimates.

The book also covers various evaluation metrics, including accuracy, precision, recall, F-measure, and AUC. Raschka emphasizes the importance of selecting appropriate metrics based on the specific application and the nature of the data. For instance, in imbalanced datasets, accuracy might be less informative than precision and recall.

Raschka's comprehensive coverage of model evaluation techniques provides a valuable reference for researchers and practitioners in machine learning. The detailed discussion of resampling methods and evaluation metrics helps readers understand the strengths and limitations of different approaches, ultimately aiding in selecting the most appropriate methods for their specific use cases.

1.4 Approaches for Competition Scenarios

Wainer [65] compares machine learning algorithms using a rank-based method, which can be helpful in competitions where only the predictions are available. However, this approach also requires multiple datasets and does not fully address the competitive nature of single-dataset competitions. This study proposes a rank-based method for comparing algorithms suitable for scenarios with multiple datasets. The rank-based approach involves ranking the performance of each algorithm on individual datasets and then aggregating these ranks to determine overall performance. This method is beneficial in competitions where direct access to algorithms is restricted, as it relies solely on prediction results.

The rank-based method proposed by Wainer involves assigning ranks to the algorithms based on their performance on each dataset. The algorithm with the best performance on a dataset receives the highest rank, and the ranks are aggregated across all datasets to determine the overall ranking. This simple and intuitive approach makes it easy to understand and implement.

However, Wainer acknowledges the limitations of this method, particularly in scenarios with a single dataset. The method cannot leverage the variability across multiple datasets for a robust comparison. Additionally, the rank-based method does not account for the magnitude of differences in performance; it only considers the relative ranking.

To address these limitations, Wainer suggests complementing the rank-based method with statistical tests that assess the significance of performance differences. For example, the Wilcoxon signed-rank test can compare the ranks of two algorithms across multiple datasets. This combination provides a more comprehensive evaluation framework that balances simplicity and statistical rigor.

Wainer's work highlights the importance of developing methods tailored to the constraints of competition scenarios. Focusing on rank-based methods and their limitations, this study provides valuable insights for researchers and practitioners designing fair and effective competition frameworks.

Lacoste et al. [44] introduce a Bayesian approach for comparing machine learning algorithms on single and multiple datasets. This method evaluates performance differences probabilistically, explicitly modeling uncertainty and variability across datasets. Unlike frequentist methods, which rely on fixed hypothesis tests and binary outcomes, Bayesian methods provide a more flexible and interpretable framework. The authors emphasize that their framework requires access to the algorithms, not merely their predictions, as it involves modeling the algorithms' internal behavior and decision-making processes. By treating algorithm performance as random variables and using Bayesian inference, this approach offers a robust way to incorporate prior knowledge and quantify uncertainty in performance comparisons.

The Bayesian framework proposed by Lacoste et al. models the performance of each algorithm as a probability distribution, reflecting the uncertainty and variability inherent in machine learning experiments. The framework can incorporate prior information into the analysis using prior distributions based on previous knowledge or assumptions. Bayesian inference then updates these priors with the observed data to produce posterior distributions, which provide a probabilistic algorithm performance assessment.

One key advantage of this approach is its ability to quantify uncertainty in performance estimates. Traditional methods often provide point estimates of performance metrics, which do not capture the variability and uncertainty in the data. In contrast, the Bayesian approach produces posterior distributions that reflect the range of possible performance outcomes and their associated probabilities, a process that benefits significantly from direct access to the algorithms' operations.

Lacoste et al. also discuss the practical implementation of their framework, including the selection of appropriate prior distributions and the computational challenges associated with Bayesian inference. They provide guidelines for choosing priors based on the specific context, available knowledge, and strategies for efficient computation using techniques such as Markov Chain Monte Carlo (MCMC) methods.

The Bayesian approach presented by Lacoste et al. offers a powerful and flexible framework for comparing machine learning algorithms. By incorporating prior knowledge and quantifying uncertainty, this method provides a more nuanced and informative assessment of algorithm performance, particularly in scenarios with limited data or high variability. However, this nuanced analysis relies heavily on direct access to the algorithms, as it is the algorithms' internal processes, not just their outputs, that are integral to this probabilistic framework.

1.5 Summary

The literature review on the comparison of Classification algorithms reveals a broad spectrum of methodologies and frameworks tailored to different aspects of performance evaluation. The studies reviewed can be categorized into three main themes: statistical tests and methods, evaluation metrics and frameworks, and specific approaches for competition scenarios.

Statistical tests, as explored by Dietterich [21], Demšar [20], and Garcia et al.

[30], provide robust tools for assessing algorithm performance differences. These include parametric tests like the paired t-test and non-parametric methods such as the Wilcoxon signed-rank test and the Friedman test, which are particularly useful in dealing with the diverse nature of machine learning data. However, these tests often assume access to the algorithms, which may not be available in competitive settings where only prediction outputs are provided.

Evaluation metrics and frameworks, as discussed by Lavesson and Davidsson [45], Olson et al. [52], and Raschka [57], emphasize the importance of using a variety of performance metrics and designing comprehensive benchmarking frameworks. Metrics like accuracy, precision, recall, F-measure, and AUC are crucial for capturing different performance dimensions, especially in prediction-only frameworks. Proper experimental design, including Cross-Validation and Resampling Techniques, ensures fair and reliable comparisons.

Wainer [65] and Lacoste et al. [44] highlight specific approaches for competition scenarios, focusing on methods that can operate under the constraints of competitive environments where only predictions are accessible. Rank-based methods and Bayesian approaches offer valuable insights for fair algorithm comparison without requiring access to the underlying algorithms. These methods provide flexible and robust frameworks for evaluating performance differences, accommodating the uncertainty and variability inherent in machine learning experiments.

Overall, while the existing literature provides extensive tools and methodologies for comparing Classification algorithms, there remains a significant gap in addressing the unique challenges of competition frameworks. Future research should aim to develop methods that effectively leverage prediction outputs to ensure fair and accurate comparisons in competitive settings, enhancing the reliability and applicability of algorithm evaluations in practical applications.

In the referenced works by **Dietterich (1998)** [21], **Demšar (2006)** [20], **García et al. (2008)** [30], and **Wainer (2022)** [65], access to algorithms rather than just predictions is necessary for statistical tests comparing machine learning models due to several critical reasons:

1. **Understanding Algorithm Behavior**: Dietterich (1998) emphasizes that statistical tests for comparing learning algorithms, such as Cross-Validation, require examining the variance caused by the algorithms' training process. This means that to conduct proper statistical analysis, it's crucial to understand how algorithms learn from data across different runs, which is only possible by having access to the algorithms themselves, not just their outputs. For instance, in 5x2 Cross-Validation tests, the variation between folds is influenced by the learning algorithm's structure, which cannot be assessed merely by the predictions.

2. **Resampling and Variability**: García et al. (2008) highlight that non-parametric statistical tests like the Friedman and Wilcoxon tests rely on measuring differences in the rankings of algorithms across multiple datasets. This comparison of algorithms requires access to both the internal workings and multiple iterations of the algorithms on different datasets to assess performance variability. Without the algorithm, key elements like random initialization or optimization methods that impact model performance cannot be evaluated.

3. **Post-hoc Tests**: Demšar (2006) discusses the importance of post-hoc tests (like the Nemenyi test) that allow for comparisons across multiple classifiers. These tests rely on performance metrics produced as part of the algorithmic process, meaning simply analyzing predictions without understanding the source of those predictions (the algorithm) undermines the validity of these comparisons.

4. **Bayesian Insights**: Wainer (2022) introduces the Bayesian Bradley-Terry (BBT) model, which requires access to the algorithm's internal processes to evaluate not just the ranking of algorithms but also the probability that one algorithm is better than another. This probabilistic approach allows for defining practical equivalence (ROPE) between algorithms, an assessment that cannot be determined with predictions alone. The BBT model also emphasizes the importance of understanding the internal variability of the algorithms across datasets, something that can only be achieved by accessing the algorithms themselves.

In summary, statistical comparisons depend on understanding the variability and nuances of algorithm behavior across different conditions, which requires full access to the algorithms and not just their final outputs.

Chapter 2

Theory Framework

2 Theory Framework

Chapter 2 provides a comprehensive exploration of the theoretical foundations supporting this research. It introduces essential concepts in machine learning, including supervised learning and key resampling techniques, such as Bootstrapping, which play a critical role in evaluating model performance. By examining these methodologies, the chapter establishes the groundwork for the experimental analyses and performance comparisons conducted in later sections, ensuring a clear understanding of the tools and approaches central to this study.

2.1 Introduction

This chapter establishes the theoretical framework underlying the methodologies employed in this research, with a particular focus on machine learning and statistical evaluation techniques. By exploring the fundamental concepts of supervised learning, resampling methods, and Bootstrapping, this chapter provides the conceptual tools required for assessing algorithm performance in competitive and research contexts.

The discussion begins with an overview of machine learning, emphasizing the differences between supervised, unsupervised, and reinforcement learning. A deeper examination of supervised learning techniques, such as Classification and regression, is included to contextualize their relevance to real-world applications. This is followed by an exploration of resampling techniques, which are critical for validating machine learning models, ensuring their robustness, and facilitating reliable performance comparisons.

Special attention is given to the bootstrap method, a powerful non-parametric approach for evaluating performance metrics and constructing Confidence Intervals. This chapter also highlights the role of permutation tests and their application in feature importance analysis and statistical significance testing, underscoring their utility in addressing challenges posed by high-dimensional data and complex distributions.

By integrating these concepts, this chapter lays the groundwork for the experimental methodologies and evaluations conducted in subsequent chapters, ensuring a comprehensive understanding of the tools and techniques that underpin this research.

2.2 Machine Learning

Machine Learning (ML) is a subset of artificial intelligence that focuses on developing algorithms that enable computers to learn from and make data-based decisions. Three main machine learning types are supervised, unsupervised, and reinforcement learning [48].

Supervised Learning: In supervised learning, the algorithm is trained on a labeled dataset, which means that each training example is paired with an output label. The objective is to learn a mapping from inputs to outputs that can be used to predict the labels of unseen data. This type of learning is commonly used in applications such as image recognition, speech recognition, and predictive analytics [12].

Unsupervised Learning: Unsupervised learning involves training algorithms on datasets that do not have labeled responses. The goal is to identify patterns or structures within the data. Techniques such as clustering and association are used in unsupervised learning to find hidden patterns or intrinsic structures in the data. Applications include market basket analysis, customer segmentation, and anomaly detection [48].

Reinforcement Learning: Reinforcement learning is a type of learning where an agent learns to make decisions by taking actions in an environment to maximize cumulative reward. It differs from supervised learning in that the correct input/output pairs are never presented, and sub-optimal actions are not explicitly corrected. Instead, the agent learns from the consequences of its actions, adjusting its strategy to achieve better outcomes over time. Common applications include robotics, game playing, and autonomous vehicles [35].

2.3 Supervised Learning

Supervised learning is one of the most widely used types of machine learning. It involves training a model on a labeled dataset, meaning the data includes both input features and the corresponding correct output. The model learns to map inputs to outputs to predict new, unseen data output. Supervised learning is divided into two main categories: Classification and regression [29].

2.3.1 Classification

Classification is a type of supervised learning where the output variable is categorical. A Classification algorithm aims to assign input data to one of a finite set of categories. For example, an email can be classified as “spam” or “not spam” [12].

Mathematically, the Classification problem can be described as follows: Given a training set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where x_i represents the feature vector of the i-th example and y_i is the corresponding class label, the objective is to learn a function $f : X \to Y$ where X is the input space and Y is the set of possible class labels.

One common algorithm for Classification is logistic regression, which models the probability that a given input belongs to a certain class. The logistic function is defined as:

$$P(Y = 1|X = x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{2.1}$$

where $P(Y = 1|X = x)$ is the probability that the output is 1 given the input x, and β_0 and β_1 are parameters to be learned [12].

Another widely used Classification algorithm is the Support Vector Machine (SVM), which finds the hyperplane that best separates the classes in the feature space. The decision boundary in an SVM is defined by:

$$w \cdot x + b = 0 \tag{2.2}$$

where w is the weight vector and b is the bias term. The SVM algorithm aims to maximize the margin between the two classes [64].

2.3.2 Regression

Regression is another type of supervised learning where the output variable is continuous. A regression algorithm aims to predict the output value based on input features. For example, predicting the price of a house based on its features such as size, location, and number of rooms [12].

Mathematically, the regression problem can be described as follows: Given a training set $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ where x_i represents the feature vector of the i-th example and y_i is the corresponding continuous output, the objective is to learn a function $f : X \to \mathbb{R}$ where X is the input space and $\mathbb{R}$ is the set of real numbers.

One common algorithm for regression is linear regression, which models the relationship between the input features and the output as a linear combination of the features. The linear regression model is defined as:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p + \epsilon \tag{2.3}$$

where y is the output, $\beta_0, \beta_1, \ldots, \beta_p$ are the coefficients to be learned, $x_1, x_2, \ldots, x_p$ are the input features, and ϵ is the error term [29].

Another popular regression algorithm is the Decision Tree Regression, which uses a tree-like model of decisions. The decision tree splits the data into subsets based on the value of the input features, and the splits are chosen to minimize the error in predicting the output [29].

2.4 Resampling Techniques

Machine learning models require robust validation techniques to ensure their performance and generalizability. Resampling techniques, such as K-fold Cross-Validation and Bootstrapping, play a crucial role in this process by providing reliable estimates of model performance.

2.4.1 Cross-Validation

Cross-Validation is a widely used resampling technique in which the data is partitioned into subsets, and the model is trained and validated on different combinations of these subsets. The most common forms of Cross-Validation are k-fold Cross-Validation and leave-one-out Cross-Validation (LOOCV).

K-Fold Cross-Validation: In k-fold Cross-Validation, the data is divided into k equally sized folds. The model is trained on $k-1$ folds and tested on the remaining fold. This process is repeated k times, with each fold used exactly once as the test set. The performance metric is averaged over the k iterations to provide a robust estimate of model performance [6, 37].

Leave-One-Out Cross-Validation: LOOCV is a case of k-fold Cross-Validation where k equals the number of data points. This method is computationally intensive but can be useful for small datasets. It involves training the model on all data points except one and repeating this process for each, providing an almost unbiased estimate of model performance [6].

2.4.2 Bootstrapping

Bootstrapping is a powerful statistical method that involves repeatedly sampling from the data with replacement to create multiple training sets. This technique estimates the distribution of a statistic (e.g., mean, variance) by sampling and is particularly useful for assessing the stability and variance of machine learning models [24].

Bootstrapping can be used to estimate the Confidence Intervals of model performance metrics, such as accuracy or AUC. By evaluating the model on multiple bootstrap samples, practitioners can gain insights into the variability and reliability of the model's predictions [29].

2.4.3 Permutation Tests

Permutation Tests, also known as randomization tests, are a non-parametric statistical technique used to assess the significance of an observed relationship between variables. This method is especially useful in scenarios where the assumptions required for traditional parametric tests, such as normality or homoscedasticity, may not hold. Instead of relying on these assumptions, permutation tests generate a null distribution by randomly rearranging or permuting the labels of the data points. This allows for the computation of a test statistic under the null hypothesis, which states no association between the independent and dependent variables [33].

One of the key advantages of permutation tests is their flexibility and applicability to various types of data. Since they do not depend on the underlying distribution of the data, they can be applied to situations where traditional methods might struggle. Permutation tests have been widely used in a variety of fields, including biology, psychology, and, more recently, machine learning. In the latter, these tests are often used to evaluate the statistical significance of model features, as they allow for a more robust assessment of feature importance by circumventing the limitations of parametric tests [34].

In machine learning, permutation tests are commonly employed to assess the contribution of each feature to the model's performance. The process involves randomly shuffling the values of a given feature while leaving the other features unchanged and then measuring how much the model's accuracy or performance metric declines as a result. If a feature is important, its randomization will lead to a noticeable decrease in performance, indicating that the feature provides valuable information to the model. By comparing the model's performance on the original data with that on the permuted data, researchers can quantify the significance of the feature in relation to the overall prediction task [51].

Furthermore, permutation tests are especially useful when working with models that involve a large number of features, such as in high-dimensional datasets. In these cases, identifying which features contribute the most to the model's predictive capability is crucial for improving model efficiency, interpretability, and generalization to new data. Permutation tests offer an intuitive and computationally feasible approach to achieve this, as they do not require retraining the model for each feature assessment, unlike other feature importance techniques such as recursive feature elimination [15].

In summary, permutation tests provide a powerful and flexible tool for assessing the statistical significance of features in a machine-learning context. By comparing the model's performance on shuffled data with its performance on the original dataset, this method offers valuable insights into the importance of individual features, even in the presence of complex or non-standard data distributions. This makes permutation tests an essential component in the toolkit for feature selection and model interpretation [33].

2.4.4 Recent Advancements and Practical Considerations

Recent advancements in resampling techniques focus on improving computational efficiency and adapting methods to complex data structures, such as time series and hierarchical data. Techniques like stratified Cross-Validation and time-series split are tailored for specific data types, ensuring more reliable performance estimates [10].

Practitioners must consider the computational cost when implementing resampling techniques, especially with large datasets and complex models. Efficient implementation and parallel computing can alleviate some of these challenges. Additionally, understanding the assumptions and limitations of each resampling method is crucial for accurate model evaluation [37].

2.5 Bootstrap

The concept of “Bootstrapping” originates from the idea of “pulling oneself up by one's bootstraps,” a phrase that seems to have been first coined in 1786 by Rudolph Erich Raspe in his book *The Singular Travels, Campaigns, and Adventures of Baron Munchausen*. Statistics refers to making inferences about a sampling distribution of statistics by resampling the sample itself with replacement [17, 24]. The accuracy of inferences depends on how well the resampling distribution replicates the original sampling distribution. This accuracy improves with larger original sample sizes, assuming the central limit theorem holds.

This technique is advantageous when our sample is small or it is difficult to obtain a representative sample from the population. Bootstrapping helps overcome the limitations of traditional statistical methods by providing a way to estimate the variability and uncertainty of statistics without making strong assumptions about the population distribution. We can use the Bootstrapping method to obtain Confidence Intervals for our statistics of interest by repeatedly resampling the sample data and calculating the statistic of interest. The term resampling was initially used in 1935 by R. A. Fisher in his famous randomization test and in 1937 and 1938 by E. J. G. Pitman, but in these instances, the sampling was carried out without replacement.

The theory and applications of the bootstrap have exploded in recent years, and the Monte Carlo approximation to the bootstrap has developed into a well-established method for drawing statistical conclusions without making firm parametric assumptions. Bootstrap refers to various methods now included under the broad category of nonparametric statistics known as resampling methods. Brad Efron's publication in the *Annals of Statistics* was published in 1979, making it a crucial year for the bootstrap [23, 24]. Efron developed the bootstrap resampling technique. His initial objective was to extract the bootstrap's features to understand better the jackknife (an earlier resampling technique created by John Tukey). He built it as a straightforward approximation to that technique. However, as a resampling method, the bootstrap frequently performs better than the jackknife.

2.5.1 Applications of Bootstrapping in ML

Bootstrap methods have numerous applications in machine learning, some of which are discussed below.

Bootstrap is widely used to assess the performance of predictive models.

By generating multiple bootstrap samples, one can obtain a distribution of model performance metrics such as accuracy, precision, and recall. This approach provides a more robust evaluation compared to traditional train-test splits, as Hastie (2009) [37] explains.

In feature selection, bootstrap methods help estimate the stability and importance of features. By repeatedly sampling the dataset and evaluating feature selection algorithms, one can determine which features consistently contribute to model performance, as discussed by Friedman (2001) [29].

Bootstrap is a fundamental component of ensemble methods like bagging and random forests. In bagging, multiple models are trained on different bootstrap samples, and their predictions are aggregated to produce a final prediction. This technique reduces variance and improves model robustness, as demonstrated by Breiman (2001) [15].

2.5.2 Case Studies

A study by Breiman (2001) [15] demonstrated the effectiveness of bootstrap methods in improving the accuracy and reliability of predictive models. The study used bootstrap to construct ensemble models, which consistently outperformed single-model approaches.

Research by Meinshausen (2010) [47] highlighted using Bootstrap to assess the stability of feature selection. The findings indicated bootstrap methods provided valuable insights into which features were reliably important across different data samples.

Bootstrap has already been applied in NLP, particularly in the statistical significance analysis of NLP systems. For instance, in the study conducted by Koehn (2004) [41], bootstrap was used to estimate the Type I Error of the BLEU (bilingual evaluation understudy) score in Machine Translation (MT). Similarly, the research by Zhang (2004) [67] was employed to measure the Confidence Intervals for BLEU/NIST scores. Additionally, in the field of Automatic Speech Recognition (ASR), researchers have used Bootstrap to estimate Confidence Intervals in performance evaluation, as illustrated in the work of Bisani (2004) [11].

Although using Bootstrap in machine learning is not a novel technique, it remains highly relevant. Bootstrap is a versatile and powerful method that enhances various aspects of machine learning, from model evaluation to feature selection and ensemble methods. Its ability to provide robust estimates and Confidence Intervals makes it an indispensable tool in the data scientist's toolkit. By resampling data and creating multiple synthetic datasets, researchers can obtain more reliable estimates of model metrics, reducing the impact of variance due to limited sample sizes.

In summary, Bootstrapping continues to be an essential technique in machine learning. It contributes to the rigorous evaluation and validation of models, which is crucial for advancing the state of the art. Future research and applications are likely to uncover even more ways Bootstrapping can contribute to the advancement of machine learning.

2.6 Summary

This chapter has provided an in-depth exploration of the key concepts and methodologies central to evaluating Algorithmic Competitions and machine learning performance. Beginning with an overview of machine learning paradigms, such as supervised and unsupervised learning, the chapter outlined foundational principles and their application contexts.

Key statistical methods for performance assessment, including bootstrap-based inference and hypothesis testing, were examined. Resampling techniques such as Cross-Validation and permutation tests were also discussed, highlighting their utility in ensuring robust and reliable evaluation metrics. The exploration emphasized the flexibility of these methods, particularly Bootstrapping, in addressing challenges such as limited data availability and non-parametric distributions.

The insights gained from this chapter establish the theoretical framework that supports subsequent experimental and practical analyses. This foundation ensures a rigorous approach to evaluating algorithm performance in competitive contexts and broader applications, aligning the methodology with best practices in modern machine learning research.

Chapter 3

Implementation of the Proposed Evaluation Framework

3 Implementation of the Proposed Evaluation Framework

3.1 Introduction

This chapter focuses on the proposed evaluation framework's practical implementation, highlighting its utility in assessing participants' performance in competitive contexts and its potential for broader applications. Building on the earlier theoretical principles, the chapter illustrates how the framework operationalizes statistical methods to ensure fair, reliable, and reproducible evaluations.

By leveraging techniques such as Bootstrapping, hypothesis testing, and multiple comparisons, the framework provides a robust mechanism for quantifying performance variability, identifying statistically significant differences, and drawing reliable conclusions about participants' rankings. These methodologies are applicable to competitions and extend to other scenarios, such as evaluating algorithms in experimental research or system benchmarking.

The chapter addresses practical questions, including how to transition from theoretical design to implementation, which tools and methods are most effective for handling diverse datasets, and how to ensure methodological consistency across varied scenarios. The insights and results derived from this work contribute to advancing fair and transparent evaluation practices, making it a valuable resource for both researchers and challenge organizers.

3.2 Implementation

3.2.1 Bootstrap´s Mathematical Formulation

The fundamental idea behind Bootstrapping is simple yet elegant. Given an original sample of size n, the bootstrap method involves generating a large number of resampled datasets (each of size n) by random sampling with replacement from the original sample. Each resampled dataset, known as a bootstrap sample, is used to calculate the statistic of interest. The collection of these bootstrap statistics forms an empirical distribution, which can be used to make inferences about the population parameter [18].

This concept can be formally expressed as follows: let $X = \{x_1, x_2, \ldots, x_n\}$ be the original sample of size n. The bootstrap procedure can be described as follows:

1. Generate B bootstrap samples, where each bootstrap sample $X_b^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ is obtained by sampling with replacement from X.

2. Compute the statistic of interest $\hat{\theta}$ for each bootstrap sample X_b^*. Let $\hat{\theta}_b^*$ denote the statistic computed from the b-th bootstrap sample.

3. Construct the empirical distribution of $\hat{\theta}$ using the bootstrap statistics $\{\hat{\theta}_1^*, \hat{\theta}_2^*, \ldots, \hat{\theta}_B^*\}$.

In Algorithm 1 the process for obtaining B bootstrap samples to compute the empirical distribution of the statistic T is presented.

The bootstrap procedure is structured systematically to implement this methodology, as outlined in the pseudocode below (Algorithm 1). This algorithm describes the step-by-step process of generating bootstrap samples, calculating the statistics of interest, and constructing the empirical distribution. By detailing these steps, the pseudocode provides a clear foundation for applying the bootstrap method within the evaluation framework.

3.2.2 Pseudocode for Bootstrapping Algorithm

The process begins with the input of the original dataset, denoted as **X**, which consists of n observations and the specification of the number of bootstrap samples, B. An empty list **T** is initialized to store the statistics computed from each bootstrap sample. For each iteration, a bootstrap sample $\mathbf{X}^*$ is generated by sampling with replacement from the original dataset **X**. The statistic of interest, denoted as t^* (e.g., mean, median,

Algorithm 1 Pseudocode for the Bootstrapping Algorithm: A resampling method used to estimate the sampling distribution of a statistic by repeatedly sampling with replacement from the original dataset.

```
Require: $\mathbf{X} = \{x_1, x_2, \ldots, x_n\}$          ▷ Original data sample of size $n$
Require: $B$                                                ▷ Number of bootstrap samples
Initialize $\mathbf{T} = []$                                ▷ List to store bootstrap estimates
for $i = 1$ to $B$ do
    $\mathbf{X}^* \leftarrow$ Sample with replacement from $\mathbf{X}$          ▷ Bootstrap sample
    $t^* \leftarrow$ Compute statistic of interest from $\mathbf{X}^*$
    Append $t^*$ to $\mathbf{T}$
end for
Compute the empirical distribution of $\mathbf{T}$
Compute Confidence Intervals and standard errors from $\mathbf{T}$ if needed
return Empirical distribution of $\mathbf{T}$
```

or standard deviation), is computed based on the values in $\mathbf{X}^*$ and appended to the list $\mathbf{T}$. This iterative procedure ensures that each bootstrap sample captures the variability inherent in the data, enabling robust statistical inference. The steps for generating bootstrap samples and analyzing their resulting statistics are explained in detail following the algorithm.

After generating all B bootstrap samples and computing the corresponding statistics, the empirical distribution of the statistic can be analyzed. This distribution allows for further statistical analyses, such as the calculation of Confidence Intervals and standard errors, based on the variability of the bootstrap estimates.

From the sampling distribution, it is possible to make inferences about the statistic of interest, which, in our case, is the performance measure. Specifically, inferences can be made through Confidence Intervals and hypothesis tests. Below, we detail how Confidence Intervals are constructed.

3.2.3 Confidence Intervals Using Bootstrap

Bootstrapping can also be used to construct Confidence Intervals for the statistic of interest, as shown by Efron [24]. Several methods are available, each with unique characteristics and suitability depending on the data and the underlying assumptions. Below, we describe some of the most commonly used approaches for constructing bootstrap Confidence Intervals, highlighting their strengths and appropriate use cases.

Percentile Bootstrap Interval: This method involves generating multiple bootstrap replicates, ordering the values of the statistic of interest, and using the percentiles of the empirical distribution to define the bounds of the Confidence Interval. For example, for a 95% Confidence Interval, the bounds would be taken from the 2.5th and 97.5th percentiles of the bootstrap statistics distribution. This approach is simple and widely used due to its ease of implementation. In general, it is constructed by taking the $(\alpha/2) \times 100$th and $(1-\alpha/2) \times 100$th percentiles of the bootstrap distribution of the statistic. For a $(1-\alpha) \times 100\%$ Confidence Interval, the percentile interval is given by:

$$[\hat{\theta}_{(\alpha/2)}, \hat{\theta}_{(1-\alpha/2)}]$$

where $\hat{\theta}_{(\alpha/2)}$ and $\hat{\theta}_{(1-\alpha/2)}$ are the $(\alpha/2) \times 100$th and $(1-\alpha/2) \times 100$th percentiles of the bootstrap distribution, respectively [24].

Bootstrap Standard Error Interval: In this method, the standard error of the statistic is estimated from the variability observed in the bootstrap replicates. Then, assuming that the distribution of the statistic is approximately normal, a Confidence Interval is constructed using the standard formula for normal-based Confidence Intervals:

$$\hat{\theta} \pm z_{\alpha/2} \cdot SE_{\text{bootstrap}},$$

where $\hat{\theta}$ is the estimator of the parameter and $SE_{\text{bootstrap}}$ is the standard error estimated from the bootstrap samples.

Bias-Corrected and Accelerated (BCa) Bootstrap Interval: This is a more advanced method that adjusts the Confidence Interval to correct bias and variability in the estimator. The BCa method uses two key parameters: the bias and the acceleration, which measure how far the bootstrap estimator is from being centered and the rate of change of the standard error, respectively. This approach is more robust than the simple percentile method, especially when the statistic of interest is biased or has a highly skewed distribution.

In the *Proposed Evaluation Framework*, the Percentile Bootstrap Interval is employed for constructing Confidence Intervals. This method is chosen for its simplicity, ease of implementation, and effectiveness in capturing the variability of performance metrics across bootstrap samples. By leveraging the percentiles of the empirical distribution, the framework ensures robust and interpretable interval estimates for the evaluation of participant performance.

3.2.4 Hypothesis Testing Using Bootstrap

Bootstrap methods provide a robust and flexible framework for conducting hypothesis tests without relying heavily on traditional parametric assumptions. Bootstrap hypothesis testing can evaluate whether observed data supports or contradicts a null hypothesis by leveraging the empirical distribution of a statistic obtained from resampling as explained by Efron (1994) [24].

The process typically begins by defining the null hypothesis (H_0) and the alternative hypothesis (H_A). For instance, in evaluating algorithm performance, H_0 might assert that there is no difference in the performance metrics of two algorithms, while H_A suggests a significant difference.

Bootstrap hypothesis testing involves the following steps:

1. **Generate Bootstrap Samples Under** H_0: Create bootstrap samples under the assumption that H_0 is true.

2. **Compute the Test Statistic**: For each bootstrap sample, compute the test statistic (e.g., the difference in means or medians between groups).

3. **Construct the Null Distribution**: Build the empirical distribution of the test statistic under H_0 using the bootstrap samples.

4. **Compare the Observed Statistic**: Compare the observed test statistic from the original dataset to the null distribution to calculate the p-value. This measures the proportion of bootstrap samples where the test statistic is as extreme as or more extreme than the observed value.

5. **Draw Conclusions**: Based on the p-value and a predetermined significance level (α), decide whether to reject or fail to reject H_0.

This approach is particularly advantageous when the underlying distribution of the data is unknown or does not meet the assumptions of classical parametric tests. By relying on resampling, bootstrap hypothesis testing offers a data-driven method for robust statistical inference.

3.2.5 Hypothesis Testing for Multiple Comparisons

When we talk about multiple testing, we mean checking several hypotheses simultaneously. This happens often in research. In our case, it's about comparing the performance of different competitors in a challenge to see how they stack up against each other. Considering the ranking generated based on the performance metric chosen by the challenge organizers, our goal is to determine whether the first place is better than the second, third, ..., up to the m-th place, given that the challenge has m participants. If we denote $\theta_i(x)$ as the performance of the i-th place in the ranking, then the multiple hypotheses would be $H_0^j : \theta_1(x) \leq \theta_j(x)$ versus $H_A^j : \theta_1(x) > \theta_j(x)$ for $j = 2, 3, \ldots, m$.

When multiple comparisons or hypothesis tests are performed on a dataset, the probability of making Type I errors (falsely rejecting a true null hypothesis) increases, as explained by Jafari (2019) [40]. This increase occurs because conducting more tests raises the likelihood of finding statistically significant results by pure chance. If we perform m independent tests, each with a significance level α, the probability of at least one Type I Error is called the Familywise Error Rate (FWER) (FWER) and is given by:

$$\text{FWER} = P(\text{at least one Type I error}) = 1 - (1 - \alpha)^m. \tag{3.1}$$

Controlling the FWER involves adjusting the significance levels of individual tests to ensure that the overall probability of making one or more Type I errors does not exceed a specified threshold, typically 0.05.

Several methods have been developed to address the issue of multiple comparisons. Below, we discuss some of the most commonly used techniques [8, 60] (Benjamini and Hochberg, 1995; Søgaard et al., 2014).

Bonferroni Correction

The Bonferroni correction is one of the simplest and most conservative methods. It adjusts the significance level by dividing α by the number of comparisons m:

$$\alpha' = \frac{\alpha}{m}. \tag{3.2}$$

While easy to implement, this method can be overly conservative, especially when the number of comparisons is large, leading to a loss of statistical power [13, 22] (Dunn, 1961; Bonferroni, 1936).

Holm's Procedure

Holm's step-down procedure is a sequentially rejective method less conservative than the Bonferroni correction. It involves sorting the p-values in ascending order and comparing each to a progressively adjusted significance level:

$$\alpha_i = \frac{\alpha}{m - i + 1}, \tag{3.3}$$

where i is the rank of the p-value. This method controls the FWER and is more powerful than the Bonferroni correction [38] (Holm, 1979).

Benjamini-Hochberg procedure

The False Discovery Rate (FDR) approach, also known as the Benjamini-Hochberg (BH) procedure introduced by Benjamini and Hochberg, focuses on controlling the expected proportion of Type I errors among the rejected hypotheses. The Benjamini-Hochberg procedure adjusts the p-values as follows:

$$p_{(i)} \leq \frac{i}{m}\alpha, \tag{3.4}$$

where $p_{(i)}$ is the ith ordered p-value. This method is beneficial in large-scale testing scenarios, such as genomic studies, where controlling the FDR is more appropriate than controlling the FWER [8] (Benjamini and Hochberg, 1995).

3.3 Performance Comparison Using Bootstrap

Comparing the performance of algorithms is a complex and ongoing problem. Performance can be defined in many ways, such as accuracy, speed, etc. Numerous performance measures have been presented in the literature, as discussed by Labatut (2012) [43], Sokolova (2009) [61], and Hastie (2009) [37]; consult appendix A.1.

The main objective of this work is to make inferences on the performance parameter θ of the algorithms developed by the teams participating in the competition. This inference is made on a single dataset of size n, with minimal submissions. The inference concerns the parameter's value (performance) in the population from which the dataset is considered to be randomly drawn.

The traditional method employed in academic competitions involves punctual estimation, which means calculating the performance of each competing system using the test dataset for each metric. The competitions specify which metrics are used and which determine the competitors' rankings. The result is a table similar to the one presented in Table 3.1, corresponding to the **Close Track** of the **VaxxStance 2021** challenge [2], which focused on determining the stance expressed on the highly controversial topic of the anti-vaxxers movement in two languages: Basque and Spanish.

The primary objective of VaxxStance 2021 was to identify whether a given tweet conveyed an *against*, *favor*, or *neutral* (none) stance regarding this predefined topic.[1] The competition introduced specific participation categories for Basque and Spanish, referred to as the **Close Track**. Within this track, participant systems were presented with two evaluation choices: **Textual**, enabling them to work exclusively with the provided tweets in the target language during development, and **Contextual**, which

[1] **Open track** and **Zero-shot track** were not considered because of too limited participation.

permitted the utilization of supplementary Twitter-related data, including user-based features, friend connections, and retweet information.

The Macro-averaged F1 score was utilized for these subtasks and exclusively applied to the *favor* and *against* classes, despite the presence of the *none* class in the dataset. Table 3.1 provides an example of the results obtained using this methodology, highlighting the differences in performance across participant systems.

Table 3.1: Macro-averaged F1 Scores for *favor* and *against* in the VaxxStance Close Track (2021) - Contextual Evaluation.

System	**Basque**
WordUp.01	0.5734
WordUp.02	0.5465
MultiAztertest.01	0.5024
SQYQP.01	0.4256
MultiAztertest.02	0.3428

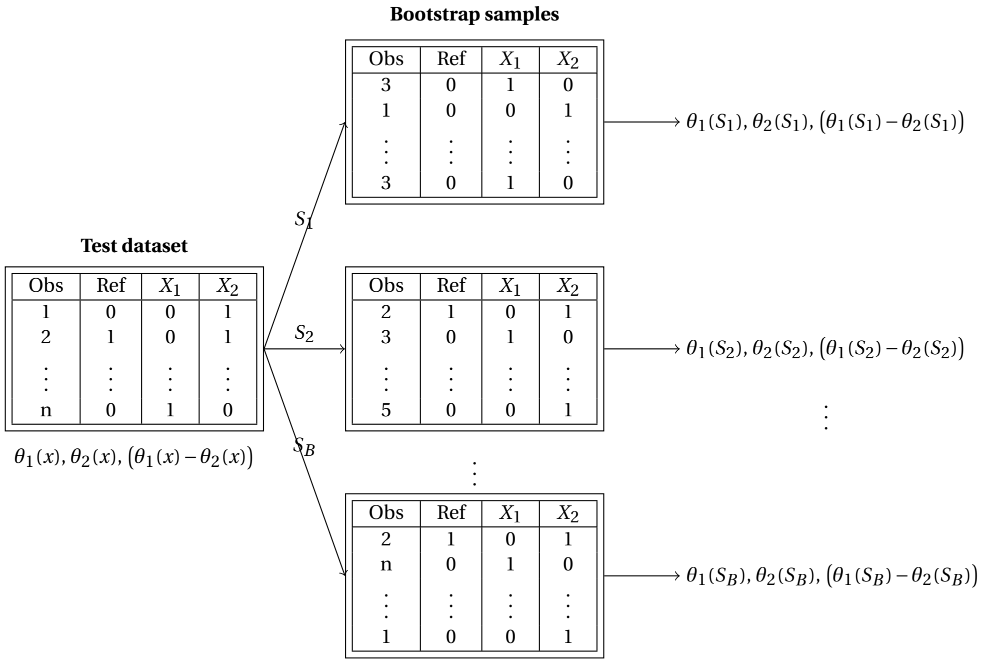
The inference on the statistical parameter θ often involves statistical hypothesis testing and Confidence Interval estimation. These statistical methods help determine whether observed differences are statistically significant or could have occurred randomly. This reasoning is now applied to performance parameters in academic competition schemes. Techniques such as Cross-Validation, bootstrap methods, and permutation tests are commonly used to assess the robustness and reliability of performance estimates, as explained by Efron (1994) [24] and Goodfellow (2016) [35]. However, beyond evaluating performance solely on the test dataset, we aim to infer the potential performance of the algorithms on the population from which the dataset was drawn. This allows us to make broader generalizations about the algorithm's behavior in real-world scenarios, ensuring that the observed performance is not merely a result of specific characteristics of the test data but reflective of its likely performance in unseen data from the same population.

To make inferences about the performance parameter θ through statistical hypothesis testing and Confidence Interval estimation, it is vital to have the sampling distribution of the parameter. Typically, this distribution is obtained by repeatedly sampling from the population and calculating the parameter of interest for each sample, which provides a distribution of the sample statistics. This approach allows a more accurate estimation of the variability and uncertainty around θ. However, in academic competitions, we only have access to a single sample—the testing dataset—making it impossible to generate a traditional sampling distribution. Consequently, we employ *Bootstrapping* to construct B bootstrap samples, each formed by resampling with replacement from the original dataset. This allows us to approximate the sampling distribution of θ using the bootstrap sample distribution. This method compensates for the lack of multiple independent samples, as discussed by Nava-Muñoz (2023) [49] and Nava-Muñoz (2024) [50]. However, it is important to note that the quality of the bootstrap estimates still depends on the original dataset's representativeness and size.

The procedure involves extracting B bootstrap samples (e.g., $10,000$, with replacement, each size n) from the dataset containing the n Gold Standard examples and their corresponding predictions for each team. For each bootstrap sample S_j, with $j = 1, 2, \ldots, B$, the performance metric $\theta_i(S_j)$ is calculated for each team i, resulting in a sampling distribution of performance metrics. These distributions provide insights into the variability and reliability of the performance metrics for each participant.

Figure 3.1 illustrates this paired bootstrap method applied to a hypothetical binary Classification competition with two participants. The figure demonstrates how the bootstrap samples are generated and how each sample's performance metrics are computed. Algorithm 1 complements this explanation by providing a detailed step-by-step pseudocode of the implementation. This approach accounts for dataset and prediction variability, ensuring a robust and reliable evaluation of participant performances across multiple iterations.

We will consider the **VaxxStance 2021** challenge for a practical understanding of how the bootstrap method works. This challenge focuses on determining the stance expressed on the highly controversial topic of the anti-vaxxers movement in two languages: Basque and Spanish. The primary objective is identifying whether a given tweet conveys an *against*, *favor*, or *neutral* (none) stance regarding this predefined topic. The dataset for this competition provides labeled examples indicating the true


Bootstrap samples
$\theta_1(S_1), \theta_2(S_1), \big(\theta_1(S_1) - \theta_2(S_1)\big)$
S_1
Test dataset
S_2
$\theta_1(S_2), \theta_2(S_2), \big(\theta_1(S_2) - \theta_2(S_2)\big)$
$\theta_1(x), \theta_2(x), \big(\theta_1(x) - \theta_2(x)\big)$
S_B
$\theta_1(S_B), \theta_2(S_B), \big(\theta_1(S_B) - \theta_2(S_B)\big)$

Obs	Ref	X_1	X_2
3	0	1	0
1	0	0	1
⋮	⋮	⋮	⋮
3	0	1	0

Obs	Ref	X_1	X_2
1	0	0	1
2	1	0	1
⋮	⋮	⋮	⋮
n	0	1	0

Obs	Ref	X_1	X_2
2	1	0	1
3	0	1	0
⋮	⋮	⋮	⋮
5	0	0	1

Obs	Ref	X_1	X_2
2	1	0	1
n	0	1	0
⋮	⋮	⋮	⋮
1	0	0	1

Figure 3.1: Illustration of the Paired Bootstrap Sampling Scheme. This figure depicts the process of generating bootstrap samples by resampling the dataset with replacement, preserving the pairing of gold standard labels and predictions. The scheme highlights how performance metrics are calculated for each sample to construct empirical distributions for robust statistical analysis.

stance for each tweet, serving as the basis for applying the bootstrap methodology to evaluate performance metrics across participants.

Table 3.2 presents an excerpt of the data used for evaluating the challenge. It includes the Gold Standard labels, which represent the true stance, along with predictions from various submissions made by participating teams. For instance, `WordUp.01` corresponds to the first submission from the WordUp team, while `WordUp.02` represents their second submission. The competition utilized all submissions from each participating team, enabling a comprehensive system performance analysis across multiple iterations. Each row corresponds to a test example, showcasing the comparison between the ground truth and the predicted labels for the different submissions under evaluation.

Table 3.3 demonstrates an example of a bootstrap sample generated from the original dataset shown in Table 3.2. In this sample, rows are selected with replacement, allowing some rows to appear multiple times while others may not appear at all. This example represents one of the B bootstrap samples that can be generated during the evaluation process.

Table 3.2: Results for VaxxStance Close Track - Contextual (2021), showcasing predictions from multiple teams compared against the gold standard labels.

Obs	Gold Standard	WordUp.01	...	SQYQP.01
1	favor	favor	...	favor
2	favor	favor	...	none
3	against	none	...	against
4	none	none	...	none
...	...	...	...	...
n_{test}	none	favor	...	against

Table 3.3: Top 5 entries from a bootstrap sample from the VaxxStance Close track results. The rows are sampled with replacements from the original dataset.

Obs	Gold Standard	WordUp.01	...	SQYQP.01
3	against	none	...	against
1	favor	favor	...	favor
2	favor	favor	...	none
n_{test}	none	favor	...	against
3	against	none	...	against

For each of the B bootstrap samples generated in the form of Table 3.3, the performance measure is calculated for every submission. These computations result in a table similar to Table 3.4, where the performance metrics, such as the Macro-averaged F1 Score for the classes *favor* and *against*, are presented for each team and their corresponding submissions. This process produces a comprehensive representation of the variability and reliability of the performance metrics across all bootstrap samples, enabling robust statistical analysis and inference about the participants' results.

Table 3.4: Top 5 entries from the 'Results for VaxxStance Close Track - Contextual (2021)' using Macro-averaged F1 Score for 'favor' and 'against'.

Sample	MultiAztertest.01	MultiAztertest.02	SQYQP.01	WordUp.01	WordUp.02
1	0.49025	0.308788	0.482413	0.617927	0.550909
2	0.524976	0.263587	0.473899	0.581218	0.622033
3	0.535814	0.366176	0.50666	0.593348	0.592989
4	0.508311	0.313008	0.508253	0.565969	0.53524
5	0.552128	0.353765	0.431905	0.59568	0.557417

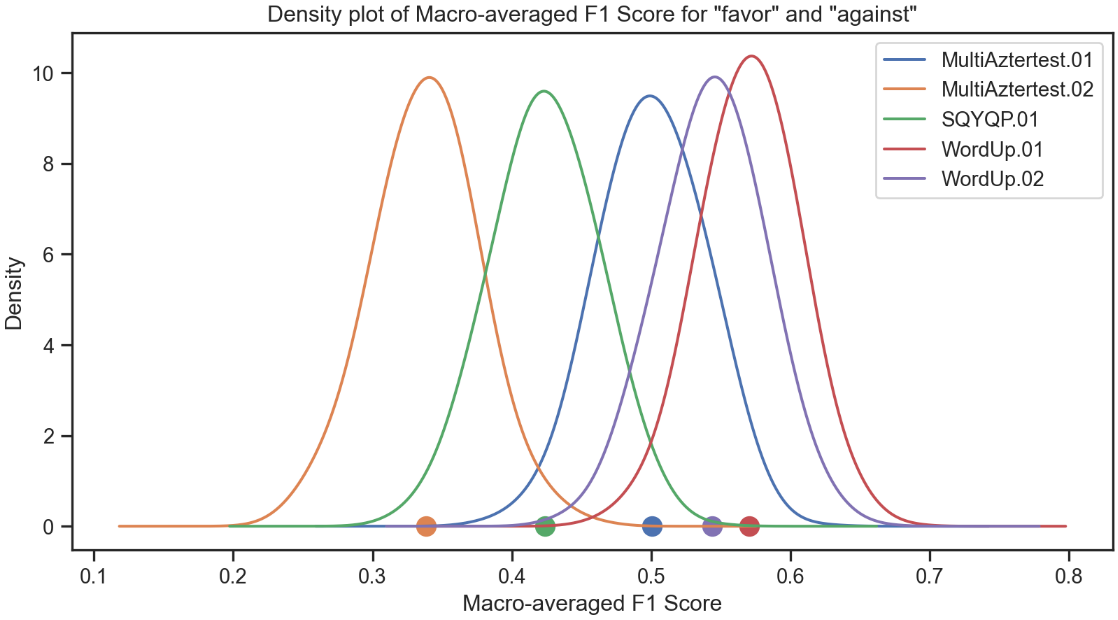


Figure 3.2: Bootstrap sampling distributions for performance metrics obtained from the VaxxStance 2021 challenge. The lines represent the empirical distributions, while the points indicate the performance of each system, providing a comprehensive view of variability and central tendency.

Finally, the sampling distribution of the performance measure is obtained, providing insights into the evaluated metric's variability and stability. This distribution, illustrated in Figure 3.2, is the foundation for constructing Confidence Intervals and conducting hypothesis tests, enabling a thorough and statistically sound comparison of participants' results.

3.3.1 Comparison through Independents Samples

We can compare the estimated bootstrap Confidence Intervals either by analyzing their numerical values or through visual inspection using graphs, as indicated in Section 3.3. Graphical representations serve as intuitive tools that facilitate comparison and support decision-making.

We follow a standard bootstrap procedure to compute these Confidence Intervals, as explained in Sections 3.2.2 and 3.2.3. First, we resample the original dataset with replacement to generate multiple bootstrap samples. For each sample, the metric of interest is calculated, resulting in a distribution of the metric values. The

Confidence Interval is then derived using the *Percentile Bootstrap Interval* from this empirical distribution by selecting the appropriate percentiles, such as the 2.5th and 97.5th percentiles for a 95% confidence level. This methodology not only quantifies the variability inherent in the performance metrics but also provides a robust framework for statistical inference.

It is important to note that while overlapping Confidence Intervals may suggest no significant difference between performances, this is not always a definitive conclusion. The degree of overlap does not perfectly correspond to statistical significance, and a more formal hypothesis test may still be required to draw conclusions. On the other hand, if the intervals do not overlap, there is a stronger indication that the difference in performance might be statistically significant.

In this context, we are considering the samples as independent, which is essential for interpreting the intervals and the results of hypothesis testing correctly. A more formal approach would involve hypothesis testing, where we set the null hypothesis H_0, that $\theta_i = \theta_j$, against the alternative hypothesis H_A, that $\theta_i \neq \theta_j$, for $i \neq j$. This approach, using the appropriate significance level α, would provide a more rigorous determination of whether the observed difference in performance is statistically significant.

3.3.2 Comparison through Paired Samples

However, since each bootstrap sample contains the Gold Standard and the predictions made by each algorithm, it is possible to calculate both the performance of each algorithm for every bootstrap sample and the performance differences between pairs of algorithms. This approach, known as the *paired bootstrap method*, is used here, as discussed by Chernick (2011) [17] and Efron (1994) [24]. Confidence Intervals at the 95% level for the difference in performance between paired samples are constructed using the same method as before. Specifically, these intervals compare the performance of the top algorithm against the second place, the top algorithm against the third place, and so on.

If the Confidence Interval for the difference includes zero, it suggests that the performance of the two algorithms is indistinguishable in the population from which the dataset is drawn, meaning H_0 cannot be rejected.

3.3.3 Statistical Hypothesis Testing

In Section 3.3.1, we introduced the conceptual ideas for constructing Confidence Intervals for comparison. Meanwhile, Section 3.3.2 outlined the theoretical basis for comparisons with the best-performing competitor. These discussions raise the question of whether it is necessary to formally evaluate the hypothesis of equality versus difference, given that the test dataset clearly shows one competitor outperforming the others. This question can be addressed by comparing the performance of two competitors, A and B, to determine whether A is superior to B in a larger data population, i.e., $\theta_A > \theta_B$. Given the test dataset $x = x_1, \ldots, x_n$, assume that A outperforms B by a magnitude $\delta(x) = \theta_A(x) - \theta_B(x)$. The null hypothesis, H_0, is that A is not superior to B in the overall population, while the alternative hypothesis, H_1, is that it is. Therefore, the goal is to determine the likelihood of a similar victory for A occurring in a new independent test dataset, denoted as y, assuming that H_0 is true.

Hypothesis testing aims to calculate the probability $p(\delta(X) > \delta(x) \mid H_0, x)$, where X represents a random variable considering the possible test sets of size n that could have been selected, while $\delta(x)$ refers to the observed difference, which is a constant. The probability $p(\delta(X) > \delta(x) \mid H_0, x)$ is known as the p-value(x). Traditionally, if the p-value$(x) < 0.05$, the observed value $\delta(x)$ is considered sufficiently unlikely to reject H_0, indicating that the evidence suggests A is superior to B, as discussed by Berg-Kirkpatrick (2012) [9].

In most cases, the p-value(x) is not easily calculated and must be approximated. This work uses the paired bootstrap method, not only because it is widely used, as discussed by Berg-Kirkpatrick (2012) [9], Bisani (2004) [11], Zhang (2004) [67], and Koehn (2004) [41], but also because it can be easily applied to any performance metric.

As shown in Berg-Kirkpatrick (2012) [9], the p-value(x) can be estimated by computing the fraction of times that this difference is greater than $2\delta(x)$. It is crucial to remember that this distribution is centered around $\delta(x)$, given that X is drawn from x, where it is observed that A outperforms B by $\delta(x)$.

3.4 Summary

This chapter presented the successful implementation of the proposed evaluation framework, showcasing its ability to provide rigorous and reproducible performance assessments in competitive settings. By employing statistical methods, such as Bootstrapping and hypothesis testing, the framework enables precise estimation of performance metrics, robust participant comparisons, and statistically sound conclusions regarding their rankings.

The results demonstrated the effectiveness of Confidence Intervals and hypothesis testing in capturing variability and ensuring the reliability of performance metrics. Additionally, methods for addressing multiple comparisons, such as the Bonferroni correction, were implemented, further strengthening the framework's analytical capabilities.

In conclusion, this chapter bridges the gap between theoretical design and practical implementation, offering a detailed guide for evaluating systems in competitive or research environments. The framework's flexibility and rigor make it a powerful tool for advancing fair and transparent evaluations in diverse contexts, paving the way for more robust decision-making processes in competitive analysis and algorithm benchmarking.

Chapter 4

Performance Comparison in Challenge Scheme

4 Performance Comparison in Challenge Scheme

This chapter focuses on comparing algorithmic performance within the context of a competitive challenge framework. Performance evaluation is critical in algorithm development and comparison, particularly in competitive environments where participants submit models or solutions to predefined tasks. This chapter delves into the various methods and metrics used to assess the performance of competing systems, highlighting the complexity of drawing inferences about algorithm efficiency and effectiveness. By examining traditional approaches alongside the nuances of specific challenges, this chapter lays the foundation for understanding how performance metrics shape competition outcomes and contribute to developing robust algorithms.

4.1 Introduction

Chapter 4 delves into the comparative analysis of Algorithmic Competitions, examining their frameworks, methodologies, and implications for advancing machine learning research. Competitions have become pivotal in benchmarking algorithms, fostering innovation, and driving collaborative problem-solving. This chapter evaluates key aspects of competitions, such as design principles, evaluation metrics, and participant dynamics, to understand how they contribute to progress in the field.

The focus is on identifying commonalities and differences across competitions, exploring their impact on participant engagement, and assessing the effectiveness of their evaluation schemes. This chapter aims to provide insights into competitions' role in shaping research directions and improving algorithmic performance by analyzing these elements. The findings are expected to guide future competition designs, ensuring they continue to foster innovation and produce meaningful outcomes for the broader research community.

4.2 Performance Comparison

Evaluating algorithm performance is multifaceted, encompassing metrics such as accuracy and speed. This study aims to infer the performance parameter θ of algorithms submitted to competitions using a single dataset of size n. The inference pertains to the broader population from which the dataset is assumed to be drawn.

Traditional approaches in academic competitions rely on punctual estimation, calculating each system's performance on a predefined test dataset using specified metrics. These metrics determine rankings and are often summarized in tables like Table 4.1, corresponding to Subtask 3 of MeOffendES 2021, organized at IberLEF 2021 and co-located with the 37th International Conference of the Spanish Society for Natural Language Processing (SEPLN 2021). The main goal of MeOffendES is to advance research in recognizing offensive language in Spanish-language variants, and Subtask 3 involves *Mexican Spanish non-contextual binary classification.* Participants must classify tweets in the OffendMEX corpus as offensive or non-offensive [54].

OffendMEX is a dataset comprising samples of offensive, aggressive, and vulgar text in Mexican Spanish, primarily collected from Twitter. Table 4.1 summarizes the Precision, Recall, and F_1 scores results for Subtask 3 of MeOffendES 2021.

Table 4.1: Results for the Non-contextual binary classification for Mexican Spanish, highlighting precision, recall, and F_1 score for each team.

Team	precision	recall	F_1
NLPCIC	0.7208	0.7100	0.7154
CIMATMTYGTO	0.6533	0.7600	0.7026
DCCDINFOTEC	0.6966	0.6733	0.6847
CIMATGTO	0.6958	0.6633	0.6792
UMUTeam	0.6763	0.6650	0.6706
Timen	0.6081	0.6000	0.6040
CICIPN	0.6874	0.5350	0.6017
xjywing	0.3419	0.8883	0.4937
aomar	0.3241	0.8750	0.4730
CENAmrita	0.3145	0.9183	0.4685

4.3 Performance Comparison through Independents samples

As indicated in Section 3.2.3, a bootstrap procedure is applied, generating $B = 10,000$ samples, with replacement and size n, from the original dataset containing the n Gold Standard examples and their predictions. Performance parameters are calculated for each team within these samples, constructing a sampling distribution of the performance metrics.

From this distribution, 95% percentile Confidence Intervals for the performance parameters are generated. Table 4.2 presents the Confidence Intervals obtained, ordered according to the estimated performance, facilitating interpretation and comparison among participants.

Table 4.2: Ordered Bootstrap Confidence Intervals for the MeOffendES Challenge. This table presents the confidence intervals for the performance of competing systems, calculated using the bootstrap method. The intervals are arranged in order to highlight the relative differences among participants and provide insights into their comparative performance.

Precision		Recall		F_1	
Team	CI	Team	CI	Team	CI
NLPCIC	(0.6844,0.7572)	CENAmrita	(0.8962,0.9402)	NLPCIC	(0.6864,0.7438)
DCCDINFOTEC	(0.6585,0.7345)	xjywing	(0.8632,0.9134)	CIMATMTYGTO	(0.6739,0.7306)
CIMATGTO	(0.6578,0.7338)	aomar	(0.8485,0.9015)	DCCDINFOTEC	(0.6536,0.7152)
CICIPN	(0.6458,0.7290)	CIMATMTYGTO	(0.7260,0.7935)	CIMATGTO	(0.6481,0.7098)
UMUTeam	(0.6381,0.7143)	NLPCIC	(0.6739,0.7458)	UMUTeam	(0.6393,0.7011)
CIMATMTYGTO	(0.6175,0.6888)	DCCDINFOTEC	(0.6351,0.7112)	Timen	(0.5713,0.6365)
Timen	(0.5691,0.6474)	UMUTeam	(0.6269,0.7025)	CICIPN	(0.5665,0.6363)
xjywing	(0.3182,0.3656)	CIMATGTO	(0.6255,0.7011)	xjywing	(0.4676,0.5196)
aomar	(0.3011,0.3470)	Timen	(0.5608,0.6392)	aomar	(0.4470,0.4987)
CENAmrita	(0.2926,0.3364)	CICIPN	(0.4946,0.5751)	CENAmrita	(0.4433,0.4935)

In this way, besides evaluating the performance of a competitor's algorithm using the testing data, we can estimate an interval that is likely to contain the performance for the population from which the testing data were drawn, with a probability of 0.95 (i.e., $\alpha = 0.05$).

However, it is important to note that while overlapping Confidence Intervals may suggest no significant difference between performances, this is not always a definitive conclusion. The degree of overlap does not perfectly correspond to statistical significance, and a more formal hypothesis test may still be required to conclude. On the other hand, if the intervals do not overlap, there is a stronger indication that the difference in performance might be statistically significant.

In this context, we consider the samples as independent, which is essential for correctly interpreting the intervals and the results of hypothesis testing. A more formal approach would involve hypothesis testing, where we set the null hypothesis H_0, that $\theta_i = \theta_j$, against the alternative hypothesis H_1, that $\theta_i \neq \theta_j$, for $i \neq j$. Using the appropriate significance level α, this approach would provide a more rigorous determination of whether the observed difference in performance is statistically significant.

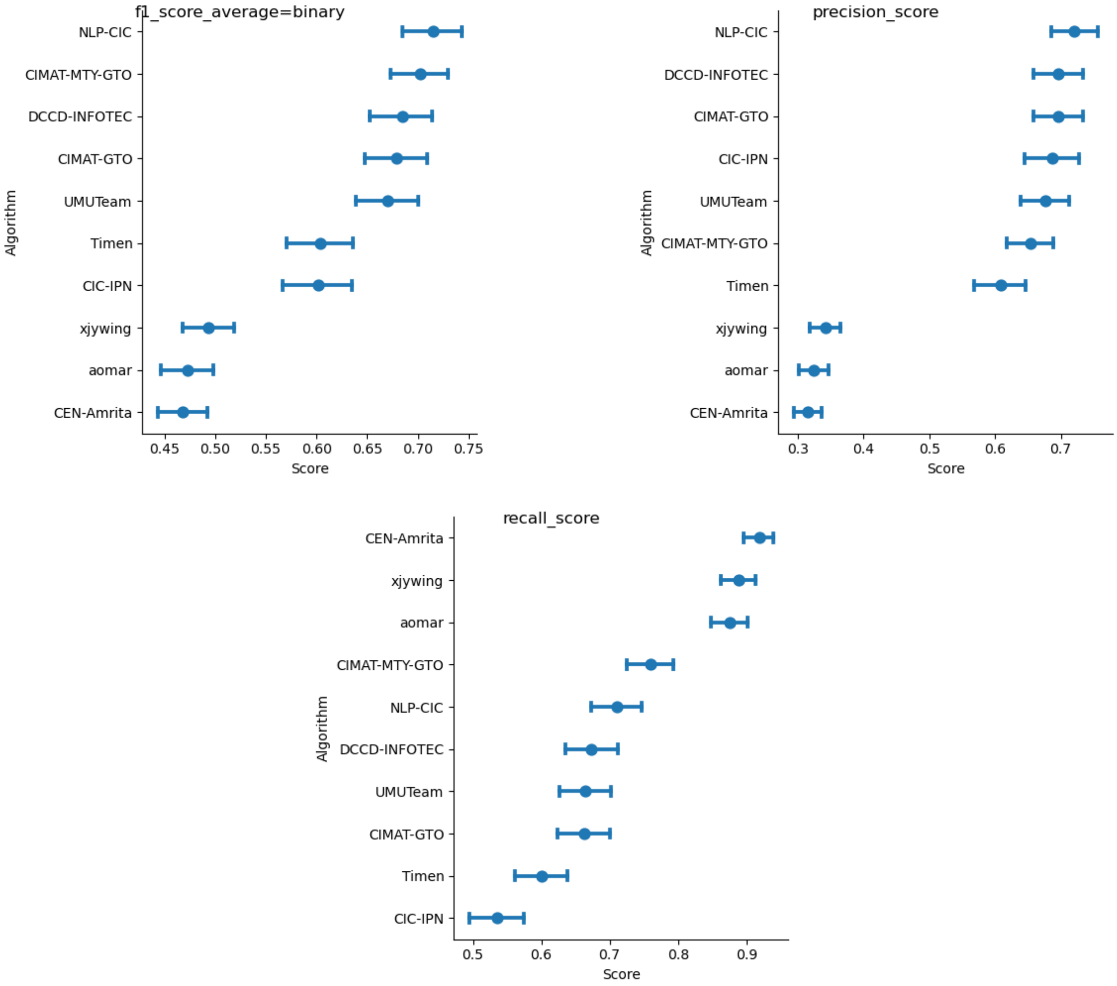


Figure 4.1: Bootstrap confidence intervals for multi-metric evaluations from the MeOffendES challenge, illustrating system performances across multiple metrics.

For the performance metrics of the participant´s systems of *OffendMEX* Subtask 3, the 95% Confidence Intervals can be seen in Table 4.2 and Figure 4.1. These intervals have been ordered to make interpretation easier. As shown, the team with the highest F_1 score is *NLPCIC*, with a 95% Confidence Interval of $(0.6864, 0.7438)$. The second place is *CIMATMTYGTO* with an interval of $(0.6739, 0.7306)$. Since the first two intervals overlap, it suggests that the F_1 scores of both teams are likely similar in the population from which the dataset was sampled. In contrast, a significant difference exists between *NLPCIC* and *Timen*.

4.4 Performance Comparison through paired samples

Given that the bootstrap samples include both the Gold Standard and each team's predictions, it is feasible to calculate the performance metrics and performance differences between teams for each sample. This paired bootstrap approach, was used to construct 95% Confidence Intervals for performance differences. These intervals specifically compare the top-performing team with the second place, the third place, and so forth. Table 4.3 and Figure 4.2 present these Confidence Intervals, enabling a clear evaluation of relative performance.

Recall that if the Confidence Interval for the difference includes zero, it suggests that the performance of the two algorithms is indistinguishable in the population from which the dataset was drawn, meaning H_0 cannot be rejected. For the F_1 score, the top-performing team is *NLPCIC*. The intervals indicate that its performance is statistically similar to that of *CIMATMTYGTO* and *DCCDINFOTEC*. However, significant differences in F_1 scores are observed when compared to the other teams.

Regarding *recall*, the best-performing team was *CENAmrita*, with no other team matching its performance. In terms of precision, *NLPCIC* led, but *DCCDINFOTEC*, *CIMATGTO*, and *CICIPN* also achieved comparable results.

Table 4.3: Bootstrap Confidence Intervals for Differences from the Best System. This table displays the confidence intervals for performance differences between competing systems and the best-performing system in the MeOffendES Challenge. The bootstrap method was employed to calculate these intervals, emphasizing the magnitude and significance of the performance gaps.

	Precision				Recall				F_1		
	NLPCIC				CENAmrita				NLPCIC		
Team	ICI	Mean	SCI	Team	ICI	Mean	SCI	Team	ICI	Mean	SCI
DCCDINFOTEC	-0.0110	0.0243	0.0596	xjywing	0.0060	0.0299	0.0539	CIMATMTYGTO	-0.0128	0.0128	0.0385
CIMATGTO	-0.0063	0.0250	0.0563	aomar	0.0221	0.0432	0.0643	DCCDINFOTEC	-0.0008	0.0307	0.0621
CICIPN	-0.0065	0.0334	0.0733	CIMATMTYGTO	0.1211	0.1585	0.1958	CIMATGTO	0.0087	0.0361	0.0635
UMUTeam	0.0116	0.0446	0.0776	NLPCIC	0.1683	0.2084	0.2485	UMUTeam	0.0161	0.0449	0.0736
CIMATMTYGTO	0.0380	0.0677	0.0974	DCCDINFOTEC	0.2058	0.2451	0.2844	Timen	0.0784	0.1112	0.1440
Timen	0.0763	0.1126	0.1488	UMUTeam	0.2122	0.2535	0.2948	CICIPN	0.0788	0.1137	0.1486
xjywing	0.3471	0.3789	0.4108	CIMATGTO	0.2150	0.2549	0.2949	xjywing	0.1896	0.2215	0.2534
aomar	0.3651	0.3967	0.4284	Timen	0.2782	0.3182	0.3582	aomar	0.2105	0.2422	0.2740
CENAmrita	0.3753	0.4063	0.4373	CICIPN	0.3401	0.3833	0.4266	CENAmrita	0.2155	0.2467	0.2779

4.5 Statistical hypothesis Testing

To further illustrate the statistical hypothesis testing process, we delve into the evaluation of p-values as a measure of significance for performance differences between competing teams. By leveraging the bootstrap distribution of performance differences, we gain insights into whether observed differences are likely to occur by random chance or are statistically meaningful. This approach builds upon the paired bootstrap methodology described earlier, allowing for precise comparisons of F_1 scores among top-performing teams.

The analysis focuses on key comparisons, such as between *NLPCIC* and *CIMATMTYGTO*, and *NLPCIC* and *DCCDINFOTEC*. The p-value is derived by evaluating the proportion of bootstrap samples that show a difference as extreme as, or more extreme than, the observed difference. These comparisons reveal whether the null hypothesis (H_0) of no significant difference in performance can be rejected in favor of the alternative hypothesis (H_1).

Figure 4.3 illustrates the p-value(x) process by showing the bootstrap distribution of the F_1 score differences between *NLPCIC* and *CIMATMTYGTO* (a), and *NLPCIC* and *DCCDINFOTEC* (b). The values zero, $\delta(x)$, and $2\delta(x)$ are highlighted for better understanding.

When comparing *NLPCIC* and *CIMATMTYGTO* in the test dataset x, the

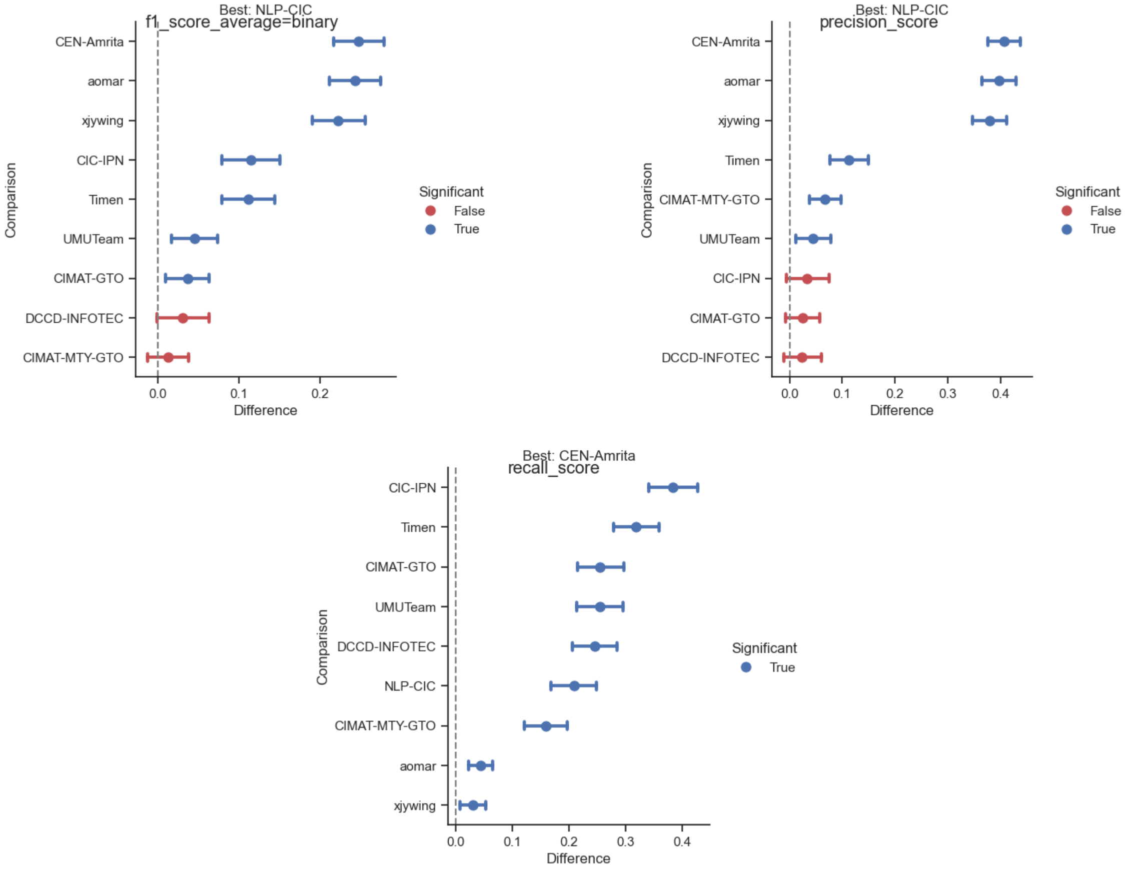


Figure 4.2: Ordered bootstrap confidence intervals for performance differences in multi-metric evaluations from the MeOffendES challenge.

difference $\delta(x) = 0.7154 - 0.7026 = 0.0128$ is not significant at the 5% level because the p-value(x) is 0.1730. On the other hand, when comparing *NLPCIC* and *DCCDINFOTEC*, $\delta(x) = 0.7154 - 0.6847 = 0.0307$, which is significant at the 5% level with a p-value(x) of 0.0292. In other words, *NLPCIC* is not significantly better than *CIMATMTYGTO* but is better than *DCCDINFOTEC*. In Section 4.4, it was shown through Confidence Intervals that the evidence supports H_0 (same performance) instead of H_1 (difference in performance). If we estimate the p-value(x), it would be approximately $2 \times 0.0292 = 0.0584$, which is not statistically significant at the 5% level.

Table 4.4 summarizes the differences in the F_1 scores between teams and their corresponding significance levels. The table is presented as a lower triangular matrix, where each entry represents the difference calculated as the score of the team in the column minus the score of the team in the row. For instance, *NLPCIC* outperforms

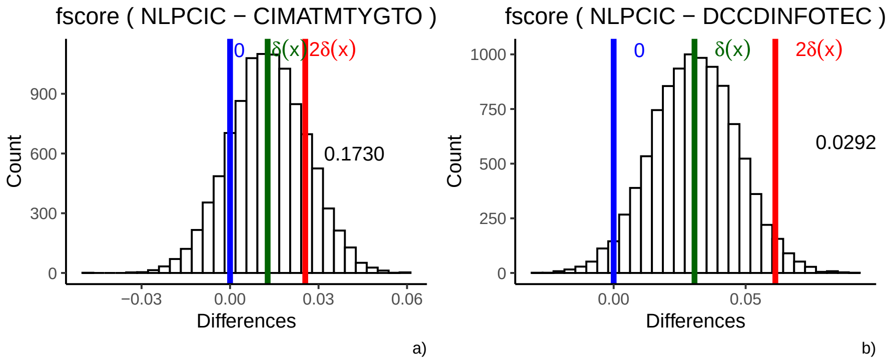


Figure 4.3: Bootstrap distribution of the F1 score differences between NLPCIC and CIMATMTYGTO (a), and NLPCIC and DCCDINFOTEC (b).

CIMATGTO by 0.036, with this difference being statistically significant at the 1% level.

Table 4.4: Pairwise Differences in F_1 Scores with Statistical Significance. This table shows the differences in F_1 scores calculated as (column)-(row) for competing systems in the MeOffendES Challenge. Statistical significance is indicated using the following notation: † for $p < .1$, * for $p < .05$, ** for $p < .01$, and *** for $p < .001$. The table highlights both the magnitude and significance of the performance differences among systems.

	NLPCIC	CIMATMTYGTO	DCCDINFOTEC	CIMATGTO	UMUTeam	Timen	CICIPN	xjywing	aomar
CIMATMTYGTO	0.013								
DCCDINFOTEC	0.031 *	0.018							
CIMATGTO	0.036 **	0.023 *	0.006						
UMUTeam	0.045 **	0.032 **	0.014	0.009					
Timen	0.111 ***	0.099 ***	0.081 ***	0.075 ***	0.067 ***				
CICIPN	0.114 ***	0.101 ***	0.083 ***	0.077 ***	0.069 ***	0.002			
xjywing	0.222 ***	0.209 ***	0.191 ***	0.185 ***	0.177 ***	0.110 ***	0.108 ***		
aomar	0.242 ***	0.230 ***	0.212 ***	0.206 ***	0.198 ***	0.131 ***	0.129 ***	0.021 ***	
CENAmrita	0.247 ***	0.234 ***	0.216 ***	0.211 ***	0.202 ***	0.135 ***	0.133 ***	0.025 ***	0.004

4.6 Multiple Comparisons

As previously mentioned in Section 3.2.5, multiple testing involves simultaneously evaluating multiple hypotheses, a common scenario in empirical research. In this work, we focus on comparing competitors' performance in a challenge, aiming to determine whether the first-place performer is statistically better than the second, third, and so on, up to the m-th participant.

When multiple hypothesis tests are conducted, the likelihood of committing Type I errors (false positives) increases. This is quantified by the Familywise Error Rate (FWER) (FWER), which measures the probability of at least one Type I error occurring among the tests. For m independent tests at significance level α, to mitigate this risk, adjustments to the individual significance levels are applied, ensuring the overall probability of making one or more Type I errors does not exceed a predetermined threshold, typically set at 0.05.

Various methods have been proposed to address the challenges posed by multiple comparisons. Here, we review some of the most frequently employed techniques, as outlined by Benjamini and Hochberg (1995) and Søgaard et al. (2014) [8,60].

The choice of method for multiple comparisons depends on the context of the study and the relative importance of Type I and Type II Errors. In clinical trials, controlling the FWER is often critical, whereas controlling the FDR may be more appropriate in exploratory research. Each method has trade-offs between conservativeness and power, and researchers must carefully consider these when designing their studies.

To provide a clearer understanding of the impact of different methods for correcting multiple comparisons, Table 4.5 illustrates the results after applying these corrections to the data presented in Table 4.4. As observed, out of 45 comparisons, 38 are statistically significant at the $\alpha = 0.05$ level without considering corrections for multiple comparisons. After applying the Bonferroni and Holm corrections, this number is reduced to 34, while using the Benjamini and Hochberg correction results in 38 significant differences once again.

These methods are essential for controlling the likelihood of false positives, which can arise when conducting multiple statistical tests simultaneously. Comparing the outcomes of various correction techniques allows us to better appreciate each approach's strengths and limitations.

In the next chapter, these correction methods will be further explored and applied to competition metrics for a more rigorous comparison. This will allow us to evaluate the performance of different algorithms under competition constraints, providing deeper insights into the reliability of the results obtained through these metrics.

Table 4.5: Estimated p-values for F_1 Score Differences. The table provides p-values for the observed differences in F_1 scores, both unadjusted and adjusted using Bonferroni, Holm, and False Discovery Rate (FDR) corrections. These adjustments account for multiple comparisons, ensuring the reliability of statistical inferences.

A	B	$A-B$	$p-value$	bonferroni	holm	fdr-hg
NLP-CIC	CIMAT-MTY-GTO	0.013	0.162	1.000	0.972	0.182
NLP-CIC	DCCD-INFOTEC	0.031	0.030	1.000	0.238	0.035
NLP-CIC	CIMAT-GTO	0.036	0.004	0.194	0.047	0.006
NLP-CIC	UMUTeam	0.045	0.002	0.081	0.022	0.002
NLP-CIC	Timen	0.111	0.000	0.000	0.000	0.000
NLP-CIC	CIC-IPN	0.114	0.000	0.000	0.000	0.000
NLP-CIC	xjywing	0.222	0.000	0.000	0.000	0.000
NLP-CIC	aomar	0.242	0.000	0.000	0.000	0.000
NLP-CIC	CEN-Amrita	0.247	0.000	0.000	0.000	0.000
CIMAT-MTY-GTO	DCCD-INFOTEC	0.018	0.107	1.000	0.745	0.123
CIMAT-MTY-GTO	CIMAT-GTO	0.023	0.015	0.657	0.131	0.018
CIMAT-MTY-GTO	UMUTeam	0.032	0.006	0.261	0.058	0.007
CIMAT-MTY-GTO	Timen	0.099	0.000	0.000	0.000	0.000
CIMAT-MTY-GTO	CIC-IPN	0.101	0.000	0.000	0.000	0.000
CIMAT-MTY-GTO	xjywing	0.209	0.000	0.000	0.000	0.000
CIMAT-MTY-GTO	aomar	0.230	0.000	0.000	0.000	0.000
CIMAT-MTY-GTO	CEN-Amrita	0.234	0.000	0.000	0.000	0.000
DCCD-INFOTEC	CIMAT-GTO	0.006	0.357	1.000	0.972	0.365
DCCD-INFOTEC	UMUTeam	0.014	0.173	1.000	0.972	0.190
DCCD-INFOTEC	Timen	0.081	0.000	0.000	0.000	0.000
DCCD-INFOTEC	CIC-IPN	0.083	0.000	0.000	0.000	0.000
DCCD-INFOTEC	xjywing	0.191	0.000	0.000	0.000	0.000
DCCD-INFOTEC	aomar	0.212	0.000	0.000	0.000	0.000
DCCD-INFOTEC	CEN-Amrita	0.216	0.000	0.000	0.000	0.000
CIMAT-GTO	UMUTeam	0.009	0.238	1.000	0.972	0.249
CIMAT-GTO	Timen	0.075	0.000	0.000	0.000	0.000
CIMAT-GTO	CIC-IPN	0.077	0.000	0.000	0.000	0.000
CIMAT-GTO	xjywing	0.185	0.000	0.000	0.000	0.000
CIMAT-GTO	aomar	0.206	0.000	0.000	0.000	0.000
CIMAT-GTO	CEN-Amrita	0.211	0.000	0.000	0.000	0.000
UMUTeam	Timen	0.067	0.000	0.009	0.003	0.000
UMUTeam	CIC-IPN	0.069	0.000	0.000	0.000	0.000
UMUTeam	xjywing	0.177	0.000	0.000	0.000	0.000
UMUTeam	aomar	0.198	0.000	0.000	0.000	0.000
UMUTeam	CEN-Amrita	0.202	0.000	0.000	0.000	0.000
Timen	CIC-IPN	0.002	0.451	1.000	0.972	0.451
Timen	xjywing	0.110	0.000	0.000	0.000	0.000
Timen	aomar	0.131	0.000	0.000	0.000	0.000
Timen	CEN-Amrita	0.135	0.000	0.000	0.000	0.000
CIC-IPN	xjywing	0.108	0.000	0.000	0.000	0.000
CIC-IPN	aomar	0.129	0.000	0.000	0.000	0.000
CIC-IPN	CEN-Amrita	0.133	0.000	0.000	0.000	0.000
xjywing	aomar	0.021	0.000	0.005	0.001	0.000
xjywing	CEN-Amrita	0.025	0.000	0.000	0.000	0.000
aomar	CEN-Amrita	0.004	0.198	1.000	0.972	0.212

4.7 Summary

This chapter explored the intricacies of performance comparison in competitive academic settings, focusing on methods and metrics used to evaluate algorithms in competitions such as *MeOffendES* Subtask 3. We discussed the challenges of drawing inferences from limited data using the test dataset from *OffendMEX*. We examined how statistical techniques, such as the bootstrap method, can help improve the reliability of performance estimates.

A significant portion of this chapter was dedicated to the paired bootstrap method, allowing for a robust performance comparison between competitors. We explained how Confidence Intervals are constructed to assess whether observed differences in performance metrics are significant, providing deeper insight into the likely generalization of results to larger populations. Additionally, we addressed the issue of multiple comparisons, emphasizing its importance in competition scenarios where the performance of several teams is compared simultaneously. Multiple comparisons require careful statistical handling to avoid inflated error rates, and we discussed methods such as adjusting confidence levels to maintain the reliability of conclusions across numerous comparisons.

The chapter also underscored the role of hypothesis testing in academic competitions, with examples illustrating how observed differences between top-performing teams can be tested for statistical significance. We demonstrated how techniques like the paired bootstrap method help assess whether the differences observed on the test dataset will likely hold in a broader population, ensuring that the results are not due to chance or specific dataset characteristics.

Overall, this chapter laid the groundwork for understanding how rigorous statistical methods, including adjustments for multiple comparisons, enhance the fairness and reliability of algorithm evaluations in competitive contexts. These methods drive innovation and help identify truly superior solutions.

Chapter 5

Comparison of Competitions

5 Comparison of Competitions

This chapter centers on the comparative analysis of competitive challenges in various fields. Whether in academic, professional, or community contexts, they serve as vital platforms and are crucial to innovation and identifying top performers. The objective is to explore their structure and impact, focusing on how they drive excellence and creativity among participants. By examining different competitive frameworks, this chapter provides insights into the role of competition in advancing methodologies, enhancing participant skills, and promoting the development of novel solutions. Comparison of competitions is essential for understanding their significance and improving future competitive frameworks.

5.1 Introduction

This Chapter focuses on the practical applications of the methodologies and frameworks developed throughout this research. This chapter integrates the theoretical insights and evaluation techniques presented in previous chapters into concrete implementations and case studies. By examining real-world applications, the chapter demonstrates the utility and effectiveness of the proposed methods in addressing challenges in competitive and research-driven environments.

The chapter begins by outlining the datasets and experimental setups used to validate the proposed frameworks. Key use cases are presented, showcasing how the methodologies enhance the evaluation and comparison of algorithms in machine learning competitions. The focus is on illustrating the adaptability of the frameworks to diverse scenarios, ensuring their relevance across different contexts and challenges.

Additionally, this Chapter discusses the implications of the results obtained from these applications, emphasizing their significance for advancing the state of the art. The findings highlight how robust evaluation techniques can foster innovation and drive meaningful contributions to the field, providing a foundation for future research and development.

Through a combination of detailed case studies and critical analysis, this chapter underscores the practical value of the methodologies developed in this work and their potential for shaping future directions in algorithmic evaluation and competition design.

5.2 Comparison of Competitions

Competitions are widespread across diverse domains, ranging from academia to professional and community settings. These events aim to recognize excellence and inspire participants to achieve their best. As noted by Escalante (2023), they encourage individuals to enhance their abilities and pursue higher standards [26]. Similarly, Egele (2024) highlights their role in fostering personal and professional growth, pushing participants to expand their boundaries and continually improve [25].

One of the primary benefits of competitiveness in these challenges is the motivational boost it provides. Driven by the desire to succeed, participants develop a deeper commitment to the task. This heightened motivation often leads to improved performance and greater accomplishment.

Furthermore, competitiveness promotes innovation and creativity. In environments where participants strive to outperform each other, there is a constant push towards developing novel solutions and approaches. This is particularly evident in fields such as technology and business, where competitive challenges often lead to breakthroughs and advancements.

5.3 Competitions Analyzed in the Comparison

Our proposed evaluation methodology is adaptable and designed to be applied universally to any challenge. We selected several NLP challenges as case studies to demonstrate their effectiveness and applicability. These competitions, encompassing various tasks and evaluation metrics, provide diverse scenarios that showcase the robustness of our approach. Below, we briefly describe each competition, highlighting the specific tasks and metrics for ranking participant systems.

MEX-A3T 2019 [5] consists of two tracks. The first track, **Author Profiling**, aims to determine the gender, occupation, and place of residence of Twitter users in Mexico based on their tweets. It incorporates text and images as information sources to assess their relevance and complementarity in user profiling. Evaluation for this track is conducted using the macro-averaged F1 score. The second track, **Aggressiveness Detection**, focuses on identifying aggressive tweets in Mexican Spanish. The evaluation is based on the F1 score in the *aggressiveness* class.

TASS 2020 [31] consists of two tracks: **General Polarity at Three Levels** and **Emotion Detection**; however, in this analysis, we focused solely on the former. The objective is to evaluate polarity Classification systems for tweets written in Spanish and their different variants. Participant systems in this competition were ranked based on the macro-averaged F1 score.

The **VaxxStance 2021** challenge [2] aims to determine the stance expressed on the highly controversial topic of the anti-vaxxers movement in two languages: Basque and Spanish. The primary objective is identifying whether a given tweet conveys an *against*, *favor*, or *neutral* (none) stance regarding this predefined topic.[1] The competition introduced specific participation categories for Basque and Spanish, referred to as the **Close Track**. Within this track, participant systems are presented with two evaluation choices: **Textual**, enabling them to work exclusively with the provided tweets in the target language during development, and **Contextual**, which permits the utilization of supplementary Twitter-related data, including user-based features, friend connections, and retweet information. The Macro-averaged F1 score was also utilized for these subtasks. Nevertheless, it was exclusively applied to two classes, *favor* and *against*, despite the presence of the *none* class in the dataset.

EXIST 2021 [58]: Sexism Identification in Social Networks. According to the following two tasks, participant systems classify tweets and posts from alternative social media platforms (in English and Spanish). The **Sexism Identification** task aims

[1] **Open track** and **Zero-shot track** were not considered because of too limited participation

to determine whether a given text is sexist. Evaluation for this track is done using the accuracy. The **Sexism Categorization** task uses only sexist texts; it categorizes the message based on the type of sexism. The macro-averaged F1 score ranks the participant systems.

DETOXIS 2021 [63] (DEtection of TOxicity in comments In Spanish) primarily aims to identify toxicity in Spanish comments posted in response to online news articles related to immigration. Specifically focusing on the **Toxicity Detection** task, it involves classifying comment content as toxic or non-toxic, with participant systems' performance ranked based on F1 scores.

MeOffendEs 2021 [54] contributes to the progress of research in identifying offensive language across various Spanish-language variations.[2] The subtask analyzed involves **Mexican Spanish non-contextual binary classification**, where participant systems categorize tweets from the OffendMEX corpus as offensive or non-offensive. The evaluation is based on the F1 score of the offensive class.

REST-MEX 2021 [3] encompasses two objectives: a **Recommendation System** and **Sentiment Analysis** utilizing text data from Mexican tourist destinations. The Recommendation System task involves forecasting the level of satisfaction a tourist might experience when suggesting a destination in Nayarit, Mexico, based on the places they visited and their feedback. Conversely, the Sentiment Analysis task determines the sentiment expressed in a review provided by a tourist who visited the most iconic locations in Guanajuato, Mexico. This competition ranked the participant systems using the metric *mean square error* (MAE).

REST-MEX 2022 [4] has three tasks: **Recommendation System** (not analyzed), **Sentiment Analysis**, and **Epidemiological Semaphore**. The Sentiment Analysis one involves classifying sentiments in tourist reviews about Mexican destinations, ranging from 1 (most negative) to 5 (most positive), with attractiveness assessment classes: Attractive, Hotel, and Restaurant, evaluated using the $measure_S$ metric [4]. Based on COVID news, the Epidemiological Semaphore task predicts the Mexican Epidemiological Semaphore. It employs a four-color system (red, orange, yellow,

[2] This challenge consists of four subtasks, but only subtask three was used.

green) with varying restrictions across Mexican states. It is assessed using the $measure_C$ metric [4].

PAR-MEX 2022 [7] (**Paraphrase Identification** In Mexican Spanish) consists in determining whether a pair represents a paraphrase relationship, i.e., classifying them as either paraphrases or non-paraphrases; the competition utilized the F1 score as the ranking metric for participant systems.

In all the analyzed competitions, the datasets include all participant systems, except in EXIST, where only the top 10 for individual languages (English and Spanish) were included. Additionally, only the best runs from each participant system were considered in EXIST, TASS, DETOXIS, PAR-MEX, and MeOffendEs. In contrast, the other competitions included all submitted runs. Another consideration is that REST-MEX 2021 and EXIST included a *baseline,* while MeOffendEs included two. REST-MEX 2022 also included the majority class. As such, when we refer to competitors, we may be referring to different runs by the same competitor or even to baselines or majority class representations. All the metrics used in these competitions are designed so that higher values indicate better performance, except for MAE (Mean Absolute Error), where lower values represent superior results.

Table 5.1 provides a summary of these competitions, detailing the subtasks, languages, ranking metrics, and the participants or runs evaluated. Competitions like MEX-A3T, TASS, VaxxStance, and REST-MEX addressed tasks such as sentiment analysis, stance detection, and sexism identification. The table also highlights the primary ranking metrics used, including F1 Score, Accuracy, and Mean Absolute Error (MAE), specifying whether all participants or only the top performers were considered in the final evaluation.

5.4 Measuring Competitiveness in Challenges

Various metrics can be employed to assess the competitiveness level in a challenge, providing valuable insights into how closely matched the participants are in terms of performance. These metrics aim to quantify the degree of similarity or disparity

Table 5.1: Overview of evaluated competitions, detailing subtasks, languages, ranking metrics, and the participants or runs considered. These provide a comprehensive summary of the competitions analyzed in this chapter.

Competition	**Subtask / Language**	**Ranking metric**	**Participants/Runs Considered**
MEX-A3T 2019	Author Profiling (Spanish, text and images) Aggressiveness Detection (Spanish)	Macro-averaged F1 Score F1 Score	All participants
TASS 2020	General Polarity (Spanish)	Macro-averaged F1 Score	All participants (Best Runs)
VaxxStance 2021	Stance Detection (Basque, Spanish)	Macro-averaged F1 Score for "favor" and "against"	All participants
EXIST 2021	Sexism Identification (English, Spanish)	Accuracy	Top 10 Runs for each language
	Sexism Categorization (English, Spanish)	Macro-averaged F1 Score	
DETOXIS 2021	Toxicity Detection (Spanish)	F1 Score	All participants (Best Runs)
MeOffendEs 2021	Offensive Language Identification (Mexican Spanish)	F1 Score	All participants (Best Runs)
REST-MEX 2021	Sentiment Analysis (Mexican Spanish)	MAE	All participants (baseline)
	Recommendation System (Mexican Spanish)	MAE	
REST-MEX 2022	Sentiment Analysis (Mexican Spanish)	$measure_S$	All participants (majority class)
	Epidemiological Semaphore (Mexican Spanish)	$measure_C$	
PAR-MEX 2022	Paraphrase Identification (Mexican Spanish)	F1 Score	All participants (Best Runs)

among competitors, offering a deeper understanding of the challenge dynamics. By analyzing factors such as score variability, ties, and performance gaps, these measures can highlight whether the competition is tightly contested, with participants achieving comparable results, or if there are significant disparities that indicate varying skill levels or algorithmic effectiveness. Such insights are essential for organizers to evaluate the challenge's effectiveness and for participants to understand their relative standing.

Scores Variability

The variability of scores is a fundamental measure of competitiveness. A lower standard deviation (σ) indicates that scores are closely clustered around the mean (μ), suggesting a highly competitive challenge.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

Coefficient of Variation (CV)

The coefficient of variation (CV) provides a normalized measure of dispersion relative to the mean score. A smaller CV implies that performance scores are tightly clustered, highlighting a competitive environment.

$$CV = \frac{\sigma}{\mu} \times 100\%$$

Number of Ties

The number of ties reflects the level of competitiveness, with more ties suggesting that multiple participants achieved similar performance levels, indicative of a tightly contested challenge.

Performance Difference

The performance difference between the winner and the median competitor measures the gap between the top and average performances. A smaller value suggests a more competitive environment with less disparity among participants.

$$PD = |win. - med.|$$

5.5 Results

In Tables 5.3 to 5.7, we use eight competitions as case studies to demonstrate the applicability of our methodology. Each table shows the information described in Table 5.2.

In the subsequent tables, we present the results of various NLP competitions to illustrate the practical application of these metrics. These tables include detailed performance metrics for each competition, providing insights into their competitiveness and potential for improvement.

The competitiveness of a Challenge can be evaluated using three primary aspects: the number of ties in relation to the possible comparisons, the coefficient of variation (CV) of participants' performance, and the performance difference between the winner and the median competitor ($|win. - med.|$).

Table 5.2: Metrics used to compare NLP competitions. The table includes descriptions of the key metrics used to evaluate competitiveness across different challenges.

Name	Description
n	Test data size
m	Number of participants or runs
Ties w/ win	Possible ties with the winner (corrections: none/Bonferroni/Holm/BH)
Poss. compars.	Total possible comparisons ($m \times (m-1)/2$)
none/Bonf./Holm/BH	Ties between competitors with corrections (none/Bonferroni/Holm/BH)
$\|win. - med\|$	Performance difference between the winner and the competitor in the middle of the table
CV	Coefficient of variation of competitors' performance. ($CV = 100 \times s_x/\overline{x}$, where $\overline{x}$ is the mean of x and s_x is the standard deviation of x)
PPI	Possible Percentage Improvement, e.g., for F1 score, it's calculated as $100 \times (1 - F_1^{winner})$

We also introduce the **Possible Percentage Improvement (PPI)** metric to assess the potential for improvement in a competition. This is particularly useful for understanding the room for growth in participants' performance. Higher PPI values indicate greater potential for improvement, suggesting that the top performance is still far from the ideal or maximum possible score.

For example, in the TASS2020 challenge (Table 5.3), the macro-averaged F1 scores across different countries show varying levels of competition. The CV values range from 12.910 in Uruguay to 31.234 in Costa Rica, indicating different levels of variability in performance. The PPI values, ranging from 32.98 to 36.647, show significant room for improvement.

Similarly, in the VaxxStance challenge (Table 5.4), the CV values for the Close Track-Contextual in Basque (64.766) highlight substantial variability, indicating a less competitive task compared to others.

When comparing different NLP competitions, different aspects must be considered. The first aspect is whether the winner is better than the competitors or other runs. In this regard, we can see that in all competitions, there is at least one tie with the winner, except in the Close Track-Contextual in Basque in the VaxxStance

Table 5.3: Results for the TASS 2020 challenge across various countries, illustrating metrics and performance variability. Metrics include macro-averaged F1 score for polarity classification tasks in Spanish tweets.

Task	General polarity at three levels				
Language	Spain	Peru	Costa Rica	Uruguay	Mexico
Metric	macro-averaged F1 score				
n	1706	1464	1166	1428	1500
m	3	3	3	3	3
Ties w/ win. (None)	1	1	1	1	1
Ties w/ win. (Bonf.)	1	1	1	1	1
Ties w/ win. (Holm)	1	1	1	1	1
Ties w/ win. (BH)	1	1	1	1	1
Poss. compars.	3	3	3	3	3
None	1	1	1	1	1
Bonf.	1	1	1	1	1
Holm	1	1	1	1	1
BH	1	1	1	1	1
$\|win. - med\|$	0.010	0.008	0.001	0.016	0.002
CV	24.010	27.310	31.234	12.910	24.625
PPI	32.98	36.647	35.365	33.669	36.599

competition, in the Sentiment task of REST-MEX 2021, and even in DETOXIS.

Overall, these indicators provide a comprehensive view of each competition's competitive landscape, helping to identify areas of strength and opportunities for improvement.

It can be observed from the tables that the most competitive task is Sexism Identification for the English language, as it has the smallest CV with a value of 0.78%. It also shows that almost all comparisons result in ties compared to the winner. Additionally, it has one of the smallest $|win. - med.|$ values, which is 0.78. It is worth noting that this task analyzed only the top 10 participants. Something similar happens with the other EXIST subtasks involving only one language (English, Spanish). If we do not consider these cases, one of the most competitive competitions is PAR-MEX 2022, where slightly less than a quarter of the total comparisons result in ties. It has a CV of 4.72% and a $|win. - med.|$ of 0.061. At the other extreme, we can find tasks like Close Track - Contextual in Basque from VaxxStance with a CV of 64.76% and a $|win. - med.|$ of 0.410. Furthermore, one out of every ten comparisons resulted in a tie. Due to the

Table 5.4: Results for the VaxxStance challenge, comparing textual and contextual tracks in Basque and Spanish. The metrics focus on macro-averaged F1 scores for stance detection subtasks.

Task	Close Track-Textual		Close Track-Contextual	
Language	Spanish	Basque	Spanish	Basque
Metric	macro-averaged F1 score(FAVOR, AGAINST).			
n	694	312	694	312
m	5	5	5	5
Ties W/ Win. (None)	1	2	1	0
Ties W/ Win. (Bonf.)	1	2	1	0
Ties W/ Win. (Holm)	1	2	1	0
Ties W/ Win. (BH)	1	2	1	0
Poss. Compars.	10	10	10	10
None	2	3	2	1
Bonf.	2	4	2	1
Holm	2	3	2	1
BH	2	3	2	1
$\|win.-med\|$	0.068	0.071	0.098	0.410
CV	9.970	19.680	10.463	64.766
PPI	19.084	42.660	10.871	22.291

nature of the MAE metric, this aspect does not include the analysis of its CV.

The calculation of Possible Percentage Improvement (PPI) is proposed to assess the potential of a task considering its metric. This indicator will be higher when the gap between the performance of the so-called winner and the ideal value of the performance metric is large. The indicator works for both metrics where the highest value is the best and for those where the lowest value is optimal, like MAE in REST-MEX 2021. In the latter case, the order of the difference is reversed. Achieving substantial improvements will be more challenging when the competition has a low PPI value. The competitions that were found to have the highest potential for improvement are MEX-A3T 2019, REST-MEX 2022 in the Epidemiological Semaphore task (although this task was a particular case due to the pandemic), EXIST in the Sexism Categorization task, and VaxxStance in the Close Track - Textual task in Basque. All of these tasks and competitions had values exceeding 39%.

Table 5.5: Results for the EXIST challenge, detailing performance in Sexism Identification and Categorization tasks across English and Spanish datasets. Metrics include accuracy and macro-averaged F1 score.

Task	Sexism Identification			Sexism Categorization		
Language	All	English	Spanish	All	English	Spanish
Metric	accuracy			macro-averaged F1 score		
n	4368	2208	2160	4368	2208	2160
m	31	10	10	28	10	10
None	4	5	3	2	4	2
Bonf.	9	9	7	6	7	5
Holm	7	9	7	2	7	4
BH	5	9	4	2	4	2
Poss. compars.	465	45	45	378	45	45
None	81	41	28	62	35	31
Bonf.	133	45	41	101	43	40
Holm	118	45	41	82	43	39
BH	89	45	37	68	36	33
$\|win. - med\|$	0.029	0.011	0.016	0.053	0.021	0.029
CV	10.920	0.780	1.190	24.120	2.140	2.260
PPI	21.95	22.28	20.55	42.13	43.96	39.27

Table 5.6: Results for the REST-MEX challenge, focusing on recommendation systems, sentiment analysis, and epidemiological semaphore tasks. Metrics include Mean Absolute Error (MAE), $measure_S$, and $measure_C$.

Challenge	REST-MEX 2021		REST-MEX 2022	
Task	Recommendation	Sentiment	Sentiment	Epi Semaphore
Metric	MAE		$measure_S$	$measure_C$
n	681	2216	12938	744
m	4	15	27	15
None	1	0	2	1
Bonf.	1	0	4	1
Holm	1	0	2	1
HB	1	0	2	1
Poss. compars.	6	105	351	105
none	1	8	16	8
Bonf.	1	12	31	14
Holm	1	9	18	8
BH	1	8	16	8
$\|win. - med\|$	0.212	0.193	0.023	0.161
CV	84.283	28.740	14.370	38.557
PPI	0.310	0.475	10.761	51.001

Table 5.7: Results for various challenges including DETOXIS, PAR-MEX, MeOffendEs, and MEX-A3T. The analysis highlights competitiveness using F1 scores and macro-averaged F1 scores for diverse NLP tasks.

Challenge	DETOXIS 2021	PAR-MEX 2022	MeOffendEs 2021	MEX-A3T 2019	
Task	Toxicity detection	Paraphrase Identification	Non contextual	Agg	author profiling
Metric	F1 score				macro-averaged F1 score
n	891	2821	2182	3156	1500
m	31	8	10	25	4
None	0	1	1	3	1
Bonf.	3	1	2	7	1
Holm	0	1	2	4	1
BH	0	1	1	3	1
Poss. compars.	465	28	45	300	6
none	80	6	7	70	2
Bonf.	135	6	9	91	2
Holm	112	6	8	80	2
BH	85	6	7	63	2
$\|win. - med\|$	0.223	0.061	0.078	0.098	0.164
CV	42.600	4.722	16.070	19.620	46.491
PPI	35.390	5.758	28.46	52.038	42.581

5.6 Summary

In this chapter, we comprehensively compared several academic competitions, focusing on how they foster innovation, skill development, and participant growth. Through a detailed analysis of various competitive frameworks, we demonstrated the critical role competitions play in pushing the boundaries of research and development, especially in areas such as natural language processing (NLP). Competitions like MEX-A3T, TASS, VaxxStance, and EXIST were examined, highlighting their unique tasks, evaluation metrics, and the diverse approaches employed by participants.

One of the key takeaways from this comparison is the importance of carefully designed tasks and evaluation criteria to ensure fair and meaningful comparisons across systems. We explored how different metrics, such as F1 score, accuracy, and Mean Absolute Error (MAE), are employed to assess performance and how they shape the strategies of competing teams. Additionally, we addressed the role of multiple evaluation tracks, like contextual and non-contextual tasks, which allow participants to explore different aspects of each challenge.

The comparative analysis also highlighted potential areas for improvement, both in the design of competitions and in the development of participant systems. By identifying competitions with the highest potential for performance gains, such as MEX-A3T and EXIST, we provided insights into how future competitions can be structured to maximize innovation and challenge participants to push the limits of their capabilities.

In conclusion, comparing competitions has proven valuable as a crucial tool for advancing methodologies, improving skills, and fostering creativity across different areas. This analysis reveals the essential function of competitions in driving research, encouraging innovation, and promoting the continuous evolution of state-of-the-art methodologies.

Conclusions

Conclusions

This chapter presents a detailed synthesis of the research findings, reflections on the theoretical and practical contributions of the work, and an analysis of the limitations and avenues for future research. The chapter is organized as follows:

Summary of Research

This thesis set out to achieve several research objectives focused on understanding and improving the evaluation of algorithms in competitive contexts. The main objectives were:

- To analyze existing methodologies for comparing algorithms in competitive scenarios.
- To develop a robust framework for evaluating algorithmic performance, focusing on statistical tests and comparison metrics.
- To apply the proposed framework to real-world Algorithmic Competitions to validate its effectiveness.

The research adopted a mixed-methods approach, combining literature review, methodological development, and empirical case studies. The key findings from this research include:

- **Effectiveness of Statistical Tests**: Through various statistical tests, the study demonstrated that specific tests (e.g., Wilcoxon Signed-Rank Test, Friedman Test) are more suitable for evaluating Classification algorithms. These tests provided robust results even when dealing with heterogeneous datasets.
- **Performance Evaluation in Competitions**: A significant finding focuses on how algorithm performance is evaluated in competitive frameworks. This includes using statistical methods and performance metrics. The thesis introduces tools for comparing results among competitors and draws statistical inferences about

their performance.

- **Evaluation Frameworks for Competitions**: A significant contribution of this thesis is developing a structured framework to compare and assess algorithm performance across different competitions. The framework facilitates the fair comparison of results, considering both performance metrics and the competitive context (e.g., task complexity, participant diversity).

These findings contribute to a deeper understanding of how to fairly and accurately assess algorithmic performance in competitive settings.

Contribution to Knowledge

The contributions of this thesis to the existing body of knowledge are both theoretical and practical:

Theoretical Contributions

The theoretical contributions of this thesis lie primarily in the formalization of competition-based algorithm evaluation. By building on existing literature, this work extends the current understanding of assessing algorithm performance in a competitive environment. Specifically, the thesis:

- Expands on statistical tests to offer a more in-depth understanding of algorithm differences.
- Proposes a new methodology for competition comparison, which can be adapted to various types of competitions (e.g., Classification, regression).

These theoretical advancements contribute to the broader field of machine learning, particularly in competitive frameworks, where objective performance evaluation is crucial.

Practical Contributions

This work's practical algorithm differences researchers and practitioners involved in Algorithmic Competitions. The proposed evaluation framework, tools, and methodologies can be directly applied to ongoing and future competitions, enhancing the accuracy and fairness of algorithm assessments. Key practical outcomes include:

- A toolkit for statistical evaluation, enabling competition organizers to compare and rank participants more effectively.
- Provide participants with actionable feedback on improving their algorithms, focusing on leveraging statistical tests.

These contributions can help improve the quality and outcomes of future Algorithmic Competitions, fostering innovation and collaboration among participants.

Versatile Framework for Systematic Evaluation in Competitions and Beyond

This work introduces a comprehensive methodology designed to compare competitors systematically and equitably in challenge-based contexts. The proposed framework is built on transparency, fairness, and reproducibility principles, ensuring a robust basis for evaluating performance and facilitating decision-making. While its primary focus is on competitive environments, the methodology's versatility extends its applicability to various scenarios, such as comparing algorithms or models in research and development projects.

The framework emphasizes critical steps to ensure unbiased evaluations. First, data partitioning involves dividing datasets into training, validation, and testing subsets, allowing for optimized model development while preventing overfitting. This process ensures that the evaluation reflects genuine performance rather than artifacts of the specific dataset. Second, predefined performance metrics are employed to assess competitors or algorithms consistently and comparably, enabling a standardized approach to evaluation across different contexts.

Although initially designed for challenges, this methodology can easily be adapted to broader use cases. By treating algorithms or models as "competitors," researchers and developers can systematically explore multiple approaches within a project, leveraging the framework's structure to assess performance rigorously. This adaptability highlights the framework's value in facilitating robust evaluations that extend beyond competitive rankings to more generalized algorithmic comparisons.

The proposed methodology is a structured approach that fosters rigor and replicability, empowering challenge organizers, researchers, and practitioners to conduct fair and consistent evaluations. By addressing diverse needs in competitive and research contexts alike, this framework represents a valuable tool for advancing transparency and informed decision-making.

Limitations of the Study

While this thesis has made significant contributions, it is important to acknowledge its limitations:

- **Data Availability**: The study utilized datasets provided by the organizers of existing Algorithmic Competitions, whose interest and support were instrumental in the success of this research. While these datasets offer valuable insights, they may not fully capture the diversity of challenges encountered in other competitive contexts, which could limit the generalizability of the findings.
- **Narrow Focus on Classification Competitions**: Although Classification tasks are prevalent in Algorithmic Competitions, the thesis primarily focuses on them, potentially limiting the applicability of the proposed methods to other types of competitions, such as regression or clustering tasks.
- **External Validity**: The research is based on a limited number of competitions, which may affect the external validity of the results. Future studies should consider a more comprehensive array of competitions across different fields to enhance the robustness of the findings.

Addressing these limitations presents opportunities for further research, which is discussed in the next section.

Future Work

This thesis opens up several avenues for future research. Suggestions for future work include:

- **Expansion to Other Competition Types**: Future research could apply the methodologies developed in this thesis to other types of Algorithmic Competitions (e.g., regression, forecasting). This would provide a more comprehensive understanding of how competition structures influence algorithm performance across different problem domains.
- **Improvement of Evaluation Metrics**: As Algorithmic Competitions grow in complexity, there is a need for more refined evaluation metrics that can capture not only performance but also aspects such as efficiency, scalability, and ethical considerations. Research in this area could lead to more holistic evaluation frameworks.
- **Consistent Integration of Statistical Tools**: A key area for future work is the consistent integration of statistical tools in problem-solving across machine learning, Algorithmic Competitions, and related fields. By embedding robust statistical analyses into the evaluation processes, researchers and competition organizers can ensure that performance differences between algorithms are understood more granularly. This could improve the accuracy of competition results, enable more effective feedback to participants, and foster the development of better-performing algorithms that are more adaptable across diverse problem domains.

These areas for future work offer exciting possibilities for advancing both the theoretical and practical aspects of Algorithmic Competition research.

Final Remarks

In conclusion, this research offers valuable insights into the field of Algorithmic Competitions and proposes a new framework for evaluating algorithm performance in these contexts. The significance of this work lies not only in its contribution to the theoretical understanding of algorithm comparison but also in its practical implications for how competitions are structured and evaluated.

The impact of this research extends beyond Algorithmic Competitions to broader fields that rely on performance-based evaluations. The tools and methodologies developed here can be adapted to other competitive domains, enhancing the fairness and accuracy of performance assessments. Moreover, by fostering more effective competition frameworks, this research encourages innovation and excellence, ultimately advancing algorithm development's state-of-the-art.

Bibliography

[1] Kaggle: Your machine learning and data science community, 2010.

[2] AGERRI, R., CENTENO, R., ESPINOSA, M., DE LANDA, J. F., AND RODRIGO, Á. VaxxStance@IberLEF 2021: Overview of the Task on Going Beyond Text in Cross-Lingual Stance Detection. *Procesamiento del Lenguaje Natural 67*, 0 (sep 2021), 173–181.

[3] ÁLVAREZ-CARMONA, M. Á., ARANDA, R., ARCE-CARDENAS, S., FAJARDO-DELGADO, D., GUERRERO-RODRÍGUEZ, R., LÓPEZ-MONROY, A. P., MARTÍNEZ-MIRANDA, J., PÉREZ-ESPINOSA, H., AND RODRÍGUEZ-GONZÁLEZ, A. Y. Overview of Rest-Mex at IberLEF 2021: Recommendation System for Text Mexican Tourism. *Procesamiento del Lenguaje Natural 67*, 0 (sep 2021), 163–172.

[4] ÁLVAREZ-CARMONA, M. Á., DÍAZ-PACHECO, Á., ARANDA, R., RODRÍGUEZ-GONZÁLEZ, A. Y., FAJARDO-DELGADO, D., GUERRERO-RODRÍGUEZ, R., AND BUSTIO-MARTÍNEZ, L. Overview of Rest-Mex at IberLEF 2022: Recommendation System, Sentiment Analysis and Covid Semaphore Prediction for Mexican Tourist Texts. *Procesamiento del Lenguaje Natural 69*, 0 (sep 2022), 289–299.

[5] ARAGÓN, M. E., ÁLVAREZ-CARMONA, M., MONTES-Y-GÓMEZ, M., ESCALANTE, H. J., VILLASEÑOR-PINEDA, L., AND MOCTEZUMA, D. Overview of MEX-A3T at IberLEF 2019: Authorship and aggressiveness analysis in Mexican Spanish tweets. *CEUR Workshop Proceedings 2421* (2019), 478–494.

[6] ARLOT, S., AND CELISSE, A. A survey of cross-validation procedures for model selection. *Statistics Surveys 4* (2010), 40–79.

[7] BEL-ENGUIX, G., SIERRA, G., GÓMEZ-ADORNO, H., TORRES-MORENO, J.-M., ORTIZ-BARAJAS, J.-G., AND VÁSQUEZ, J. Overview of PAR-MEX at Iberlef 2022: Paraphrase Detection in Spanish Shared Task. *Procesamiento del Lenguaje Natural 69*, 0 (sep 2022), 255–263.

[8] BENJAMINI, Y., AND HOCHBERG, Y. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological) 57* (1 1995), 289–300.

[9] BERG-KIRKPATRICK, T., BURKETT, D., AND KLEIN, D. An empirical investigation of statistical significance in NLP. In *EMNLP-CoNLL 2012 - 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Proceedings of the Conference* (2012).

[10] BERGMEIR, C., AND BENITEZ, J. M. A note on the validity of cross-validation for evaluating time series prediction. *Computational Statistics & Data Analysis 120* (2018), 70–83.

[11] BISANI, M., AND NEY, H. Bootstrap estimates for confidence intervals in ASR performance evaluation. *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing 1* (2004).

[12] BISHOP, C. M. *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1 ed. Springer, 2007.

[13] BONFERRONI, C. E. *Teoria statistica delle classi e calcolo delle probabilità.* Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commerciali di Firenze, 1936.

[14] BRABHAM, D. C. *Crowdsourcing.* MIT Press, 2013.

[15] BREIMAN, L. Random forests. 5–32.

[16] CHAI, T., AND DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)? -arguments against avoiding rmse in the literature. *Geoscientific Model Development 7* (6 2014), 1247–1250.

[17] CHERNICK, M. R., AND LABUDDE, R. A. *An introduction to bootstrap methods with applications to R.* Wiley, 2011.

[18] DAVISON, A. C., AND HINKLEY, D. V. *Bootstrap Methods and their Application.* Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge

University Press, 1997.

[19] De Myttenaere, A., Golden, B., Le Grand, B., and Rossi, F. Mean absolute percentage error for regression models. *Neurocomputing 192* (2016), 38–48.

[20] Demšar, J. Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research 7*, 1 (2006), 1–30.

[21] Dietterich, T. G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Computation 10*, 7 (oct 1998), 1895–1923.

[22] Dunn, O. J. Multiple comparisons among means. *Journal of the American Statistical Association 56*, 293 (1961), 52–64.

[23] Efron, B. Bootstrap Methods: Another Look at the Jackknife. *https://doi.org/10.1214/aos/1176344552 7*, 1 (jan 1979), 1–26.

[24] Efron, B., and Tibshirani, R. *An Introduction to the Bootstrap.* Chapman and Hall/CRC, may 1994.

[25] Egele, R., Junior, J. C. S. J., van Rijn, J. N., Guyon, I., Baró, X., Clapés, A., Balaprakash, P., Escalera, S., Moeslund, T., and Wan, J. Ai competitions and benchmarks: Dataset development, 2024.

[26] Escalante, H. J., and Kruchinina, A. Academic competitions, 2023.

[27] Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., and Zisserman, A. The pascal visual object classes (VOC) challenge. *Int. J. Comput. Vis. 88*, 2 (2010), 303–338.

[28] Fawcett, T. An introduction to roc analysis. *Pattern recognition letters 27*, 8 (2006), 861–874.

[29] Friedman, J. H., Hastie, T., and Tibshirani, R. The elements of statistical learning. *Springer series in statistics 1*, 10 (2001), 10.

[30] García, S., and Herrera, F. An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research 9* (2008), 2677–2694.

[31] García-Vega, M., Díaz-Galiano, M. C., García-Cumbreras, M., Del Arco, F. M. P., Montejo-Ráez, A., Jiménez-Zafra, S. M., Cámara, E. M., Aguilar, C. A., Cabezudo, M. A. S., Chiruzzo, L., and Moctezuma, D. Overview of TASS 2020: Introducing Emotion Detection. *CEUR Workshop Proceedings 2664* (2020), 163–170.

[32] Geiger, D., Seedorf, S., Schulze, T., Nickerson, R. C., and Schader, M. Managing the crowd: Towards a taxonomy of crowdsourcing processes. *Proceedings of the Seventeenth Americas Conference on Information Systems 2011* (2011), 1–11.

[33] Good, P. *Permutation Tests: A Practical Guide to Resampling Methods for Testing Hypotheses.* Springer Science & Business Media, 2013.

[34] Good, P. I. *Introduction to statistics through resampling methods and R/S-Plus.* John Wiley & Sons, 2005.

[35] Goodfellow, I., Bengio, Y., and Courville, A. *Deep learning.* MIT press, 2016.

[36] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. A. Result analysis of the nips 2003 feature selection challenge. *Advances in Neural Information Processing Systems 17* (2004), 545–552.

[37] Hastie, T., Tibshirani, R., and Friedman, J. *The elements of statistical learning: data mining, inference, and prediction.* Springer, 2009.

[38] Holm, S. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics 6* (1979), 65–70.

[39] Howe, J. The rise of crowdsourcing. *Wired magazine 14*, 6 (2006), 1–4.

[40] Jafari, M., and Ansari-Pour, N. Why, When and How to Adjust Your P Values? *Cell Journal (Yakhteh) 20*, 4 (2019), 604.

[41] KOEHN, P. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, EMNLP 2004 - A meeting of SIGDAT, a Special Interest Group of the ACL held in conjunction with ACL 2004* (2004), pp. 388–395.

[42] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM 60*, 6 (2012), 84–90.

[43] LABATUT, V., AND CHERIFI, H. Accuracy measures for the comparison of classifiers, 2012.

[44] LACOSTE, A., LAVIOLETTE, F., AND MARCHAND, M. Bayesian comparison of machine learning algorithms on single and multiple datasets, 2012.

[45] LAVESSON, N., AND DAVIDSSON, P. Evaluating learning algorithms and classifiers. *Intelligent Information Systems 2*, 4 (2006), 37–52.

[46] LINTOTT, C. J., ET AL. Galaxy zoo: morphologies derived from visual inspection of galaxies from the sloan digital sky survey. *Monthly Notices of the Royal Astronomical Society 389*, 3 (2008), 1179–1189.

[47] MEINSHAUSEN, N., AND BÜHLMANN, P. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 72* (9 2010), 417–473.

[48] MURPHY, K. P. *Machine learning: a probabilistic perspective.* MIT press, 2012.

[49] NAVA-MUÑOZ, S., GRAFF GUERRERO, M., AND ESCALANTE, H. J. Comparison of classifiers in challenge scheme. In *Pattern Recognition* (Cham, 2023), A. Y. Rodríguez-González, H. Pérez-Espinosa, J. F. Martínez-Trinidad, J. A. Carrasco-Ochoa, and J. A. Olvera-López, Eds., Springer Nature Switzerland, pp. 89–98.

[50] NAVA-MUÑOZ, S., GRAFF, M., AND ESCALANTE, H. J. Analysis of systems' performance in natural language processing competitions. *Pattern Recognition Letters* (3 2024).

[51] Ojala, M., and Garriga, G. C. Permutation tests for studying classifier performance. *Journal of Machine Learning Research 11* (2010), 1833–1863.

[52] Olson, R. S., Bartley, N., Urbanowicz, R. J., and Moore, J. H. Pmlb: a large benchmark suite for machine learning evaluation and comparison. *BioData mining 10,* 1 (2017), 1–13.

[53] Pavao, A., Guyon, I., Letournel, A.-C., Tran, D.-T., Baro, X., Escalante, H. J., Escalera, S., Thomas, T., and Xu, Z. Codalab competitions: An open source platform to organize scientific challenges. *Journal of Machine Learning Research 24,* 198 (2023), 1–6.

[54] Plaza-del-Arco, F. M., Casavantes, M., Escalante, H. J., Martín-Valdivia, M. T., Montejo-Ráez, A., Montes-y Gómez, M., Jarquín-Vásquez, H., and Villaseñor-Pineda, L. Overview of MeOffendEs at IberLEF 2021: Offensive Language Detection in Spanish Variants. *Procesamiento del Lenguaje Natural 67,* 0 (sep 2021), 183–194.

[55] Plevris, V., Solorzano, G., Bakas, N. P., and Seghier, M. E. A. B. Investigation of performance metrics in regression analysis and machine learning-based prediction models. *ECCOMAS Congress 2022 - 8th European Congress on Computational Methods in Applied Sciences and Engineering* (11 2022).

[56] Powers, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061* (2011).

[57] Raschka, S. Model evaluation, model selection, and algorithm selection in machine learning, 2020.

[58] Rodríguez-Sánchez, F., Carrillo-de Albornoz, J., Plaza, L., Gonzalo, J., Rosso, P., Comet, M., and Donoso, T. Overview of EXIST 2021: sEXism Identification in Social neTworks. *Procesamiento del Lenguaje Natural 67,* 0 (sep 2021), 195–207.

[59] RUSSAKOVSKY, O., DENG, J., SU, H., KRAUSE, J., SATHEESH, S., MA, S., HUANG, Z., KARPATHY, A., KHOSLA, A., BERNSTEIN, M. S., BERG, A. C., AND FEI-FEI, L. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis. 115*, 3 (2015), 211–252.

[60] SØGAARD, A., JOHANNSEN, A., PLANK, B., HOVY, D., AND MARTINEZ, H. What's in a p-value in NLP? *CoNLL 2014 - 18th Conference on Computational Natural Language Learning, Proceedings* (2014), 1–10.

[61] SOKOLOVA, M., AND LAPALME, G. A survey of performance evaluation measures for classification systems. *Information Processing & Management 45*, 4 (2009), 427–437.

[62] SUROWIECKI, J. *The wisdom of crowds.* Anchor Books, 2005.

[63] TAULÉ, M., ARIZA, A., NOFRE, M., AMIGÓ, E., AND ROSSO, P. Overview of DETOXIS at IberLEF 2021: DEtection of TOXicity in comments In Spanish. *Procesamiento del Lenguaje Natural 67*, 0 (sep 2021), 209–221.

[64] VAPNIK, V., AND CORTES, C. Support-vector networks. *Machine learning 20*, 3 (1995), 273–297.

[65] WAINER, J. A bayesian bradley-terry model to compare multiple ml algorithms on multiple data sets. *arXiv preprint arXiv:2208.04935* (2016).

[66] WILLMOTT, C. J., AND MATSUURA, K. Advantages of the mean absolute error (mae) over the root mean square error (rmse) in assessing average model performance. *Climate Research 30*, 1 (2005), 79–82.

[67] ZHANG, Y., VOGEL, S., AND WAIBEL, A. Interpreting BLEU/NIST Scores: How Much Improvement do We Need to Have a Better System? In *Proceedings of the 4th International Conference on Language Resources and Evaluation, LREC 2004* (2004), pp. 2051–2054.

Appendix

A Appendix

A.1 Performance Metrics

Machine learning competitions drive innovation and improve existing techniques and models. These competitions focus on various tasks, from classification to prediction, and use specific performance metrics to evaluate the efficacy of submitted models [55, 61] (Plevris et al., 2022; Sokolova and Lapalme, 2009). Below are some of the most common performance metrics in these competitions, categorized by classification and regression and further distinguished by their application to binary and multiclass classification.

A.1.1 Classification Metrics

1. **Accuracy** [28] (Binary and Multiclass):
   - **Description**: The proportion of correct predictions out of the total predictions.
   - **Usage**: Mainly in binary and multiclass classification problems.
   - **Formula**:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

2. **Precision** [56] (Binary and Multiclass):
   - **Description**: The proportion of true positives out of all predicted positives.
   - **Usage**: Important in scenarios where the cost of false positives is high.
   - **Formula**:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

3. **Recall (Sensitivity or True Positive Rate)** [56] (Binary and Multiclass):

- **Description**: The proportion of true positives out of all actual positives.
- **Usage**: Important in scenarios where it is crucial to capture all positive cases.
- **Formula**:
$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

4. **F1 Score** [56] (Binary and Multiclass):
   - **Description**: The harmonic mean of precision and recall.
   - **Usage**: Used when a balance between precision and recall is needed.
   - **Formula**:
$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

5. **F1 Score Averaged** [56] (Multiclass):
   - **Description**: The average of the F1 scores across different classes.
   - **Usage**: Utilized to measure model performance in multiclass classification problems.
   - **Formula**:
$$\text{F1}_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} F1_i$$

   where N is the number of classes and $F1_i$ is the F1 score for class i.

6. **Micro F1** [56]:
   - **Description**: The F1 score is calculated by aggregating the contributions of all classes to compute the average metric.
   - **Usage**: Used to evaluate the performance of a model across multiple classes with equal importance.
   - **Formula**:
$$\text{Micro F1} = \frac{2 \cdot \sum_{i=1}^{N} (\text{TP}_i)}{2 \cdot \sum_{i=1}^{N} (\text{TP}_i) + \sum_{i=1}^{N} (\text{FP}_i + \text{FN}_i)}$$

where TP_i, FP_i, and FN_i are the true positives, false positives, and false negatives for class i, respectively.

7. **Area Under the ROC Curve (AUC-ROC)** [28] (Binary):
   - **Description**: Measures the ability of a model to distinguish between classes.
   - **Usage**: Common in binary classification problems.
   - **Interpretation**: An AUC value close to 1 indicates good performance, while a value close to 0.5 indicates random performance.

8. **Logarithmic Loss (Log Loss)** [48] (Binary and Multiclass):
   - **Description**: Measures the uncertainty of the probabilities assigned to classes.
   - **Usage**: In probabilistic classification, predicting probabilities rather than specific classes is important.
   - **Formula**:

     $$\text{Log Loss} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(p_i) + (1-y_i)\log(1-p_i)\right]$$

     where y_i is the true value and p_i is the predicted probability of the class.

9. **measure$_\mathbf{S}$** [4]:
   - **Description**: A metric for evaluating sentiment analysis tasks.
   - **Usage**: Applied to measure the effectiveness of sentiment classification models.
   - **Formula**:

     $$measure_S = \frac{\frac{1}{1+MAE_p} + F1_A}{2},$$

     where $F1_A$ is the average among the micro F1 for each class (hotel, restaurant, and attractive), and MAE_p is the Mean Absolute Error applied to the polarity.

10. **measure$_\mathbf{C}$** [4]:

    - **Description**: A metric for evaluating epidemiological semaphore predictions.

    - **Usage**: Used to assess the accuracy of predictions for the epidemiological semaphore, indicating COVID-19 risk levels.

    - **Formula**:

    $$measure_C = \frac{F1_{w0} + 2 \times F1_{w2} + 4 \times F1_{w4} + 8 \times F1_{w8}}{15}$$

    where $F1_{wf}$ is the $F1$ score to the prediction of the COVID semaphore after f *weeks* in the future.

A.1.2 Regression Metrics

1. **Mean Squared Error (MSE)** [66]:

    - **Description**: The average of the squared differences between predicted and actual values.

    - **Usage**: Primarily in regression problems.

    - **Formula**:

    $$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

2. **Root Mean Squared Error (RMSE)** [16]:

    - **Description**: The square root of MSE, providing a measure in the same units as the output values.

    - **Usage**: Similar to MSE but more interpretable.

    - **Formula**:

    $$\text{RMSE} = \sqrt{\text{MSE}}$$

3. **Mean Absolute Error (MAE)** [66]:

- **Description**: The average of the absolute differences between predicted and actual values.
- **Usage**: In regression problems where the interpretability of the average error is important.
- **Formula**:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

4. **Mean Absolute Percentage Error (MAPE)** [19]:
   - **Description**: The average of the absolute percentage errors.
   - **Usage**: In regression problems, the data scale varies.
   - **Formula**:

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right|$$

These metrics evaluate different aspects of model performance, from overall accuracy to capturing positive cases and the uncertainty in predictions. The choice of the appropriate metric depends on the specific problem and the model's objectives.

A.2 CompStats Package

This appendix provides a comprehensive guide to using the *CompStats* library, which encapsulates the methodologies proposed in this thesis. The examples demonstrate how to evaluate performance metrics, assess differences, and analyze results across multiple metrics.

The *CompStats* library is designed to facilitate robust statistical analyses in competitive settings. It includes modules for performance evaluation, statistical testing, and visualizing results, offering a user-friendly interface for single and multi-metric assessments.

A.2.1 Installation

The library can be installed using *pip* or *Anaconda*, depending on user preference. For detailed documentation, refer to `http://compstats.readthedocs.org`.

```
pip install CompStats
```

```
conda install -c conda-forge compstats
```

Once *CompStats* is installed, one must load a few libraries.

```
from CompStats import performance, difference, plot_difference
from statsmodels.stats.multitest import multipletests
from sklearn.metrics import f1_score
import pandas as pd
```

To illustrate CompStats, we will use the PAR-MEX 2022 dataset. Let us assume *PARMEX_2022.csv* is a CSV file where the column y has the ground truth, and the other columns are the systems' outputs.

```
DATA = "PARMEX_2022.csv"
df = pd.read_csv(DATA)
```

A.2.2 Single-Metric Performance Assessment

The performance metric used is the F1 score.

```
score = lambda y, hy:f1_score(y, hy)
```

The following instructions calculate each system's performance and the performance difference relative to the best-performing system. They then compare the analyzed algorithms using confidence intervals.

```
perf = performance(df, score=score)
ins = plot_performance(perf)
```

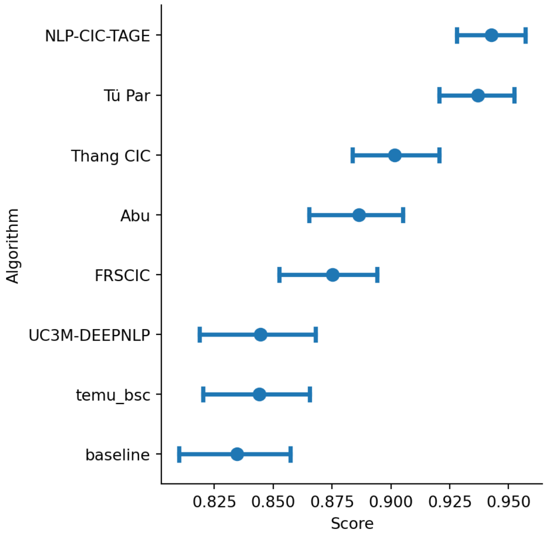


Figure A.1: Bootstrap Confidence Intervals for performance. The plot visualizes the confidence intervals for each system's performance, showing variability across bootstrap samples.

```
diff = difference(perf)
ins = plot_difference(diff)
```

Figure A.1 shows the bootstrap confidence intervals for each system's performance. Figure A.2 visualizes the ordered confidence intervals for performance differences relative to the top-performing system.

A p-value is calculated to determine if performance differences are statistically significant, offering a concise way to assess the reliability of the observed differences.

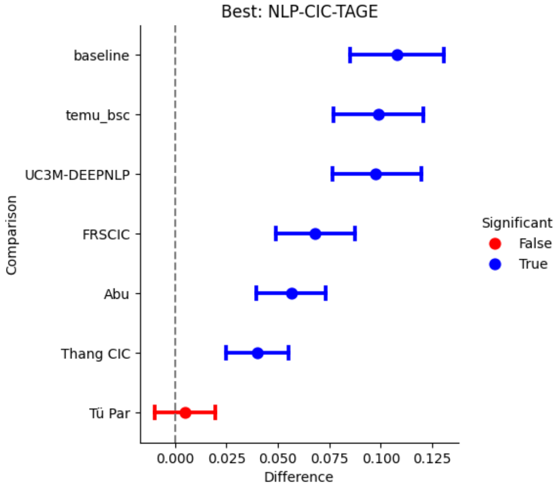


Figure A.2: Ordered Bootstrap Confidence Intervals for performance differences. Each bar represents the interval for the performance difference relative to the top-performing system.

```
p_values = difference_p_value(diff)
p_values
```

```
{'baseline': 0.0, 'temu_bsc': 0.0,  'UC3M-DEEPNLP': 0.0, 'FRSCIC': 0.0,
 'Abu': 0.0, 'Thang CIC': 0.0,  'Tü Par': 0.254}
```

The p-values indicate the likelihood of observing performance differences as extreme as those in the data, assuming no real difference exists. A p-value below the significance threshold (e.g., 0.05) suggests a statistically significant difference.

Finally, the p-values are corrected using a Bonferroni correction, which exemplifies this method of rigorously evaluating significance across multiple tests.

```
result = multipletests(list(p_values.values()),
        method='bonferroni')
p_valuesC = dict(zip(p_values.keys(),result[1]))
p_valuesC
```

```
{'baseline': 0.0, 'temu_bsc': 0.0, 'UC3M-DEEPNLP': 0.0, 'FRSCIC': 0.0,
 'Abu': 0.0, 'Thang CIC': 0.0, 'Tü Par': 1.0}
```

A.2.3 Multi-Metric Performance Assessment

For multi-metric evaluations, the library supports simultaneous analysis of multiple metrics, streamlining the comparison process. Each metric is assessed independently to compute system performance, assess differences, and derive confidence intervals.

```
from CompStats import performance_multiple_metrics,
plot_performance_multiple, difference_multiple,
plot_difference_multiple
from sklearn.metrics import f1_score, accuracy_score,
precision_score, recall_score
import seaborn as sns
```

The performance metrics used are macro-averaged F1 score, accuracy, precision and recall.

```
metrics = [
  {"func": f1_score, "args": {"average": "macro"}, ’BiB’: True},
  {"func": accuracy_score, ’BiB’: True},
  {"func": precision_score, ’BiB’: True},
  {"func": recall_score, ’BiB’: True}
]
```

The following instructions outline the procedure for evaluating the individual performance of each system, calculating the difference in their performance relative to the best-performing system, and comparing the analyzed algorithms through confidence intervals. For the multi-metric case, this approach is extended to consider multiple evaluation metrics simultaneously. Each metric is independently analyzed to compute the performance of the systems, assess their differences relative to the top-performing system, and derive confidence intervals for each metric.

```
perf = performance_multiple_metrics(df, "y", metrics)
face_grid = plot_performance_multiple(perf)
```

```
diff = difference_multiple(perf)
```

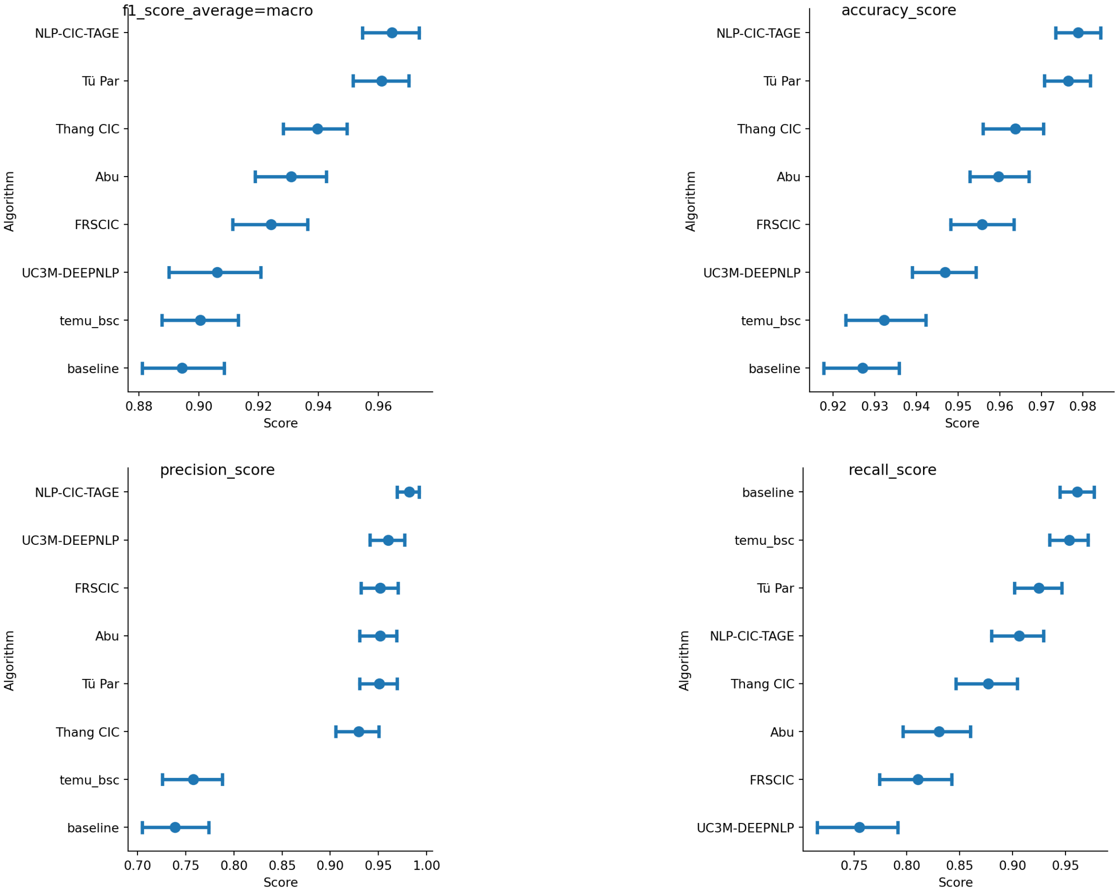


Figure A.3: Bootstrap confidence intervals for multi-metric evaluations. These plots showcase the performance metrics' reliability and variability across different criteria.

```
face_grid_diff = plot_difference_multiple(diff)
```

For the multi-metric case, Figure A.3 presents the bootstrap confidence intervals for each system's performance across multiple evaluation metrics, providing a detailed view of their reliability and variability. Similarly, Figure A.4 illustrates the ordered confidence intervals for performance differences relative to the top-performing system for each metric, enabling a comprehensive comparison of the systems under multiple criteria.

A p-value is calculated for each metric to assess whether the performance differences are statistically significant. This approach provides a detailed evaluation across multiple dimensions, allowing for a more comprehensive understanding of the algorithms' reliability and consistency in various aspects. By analyzing p-values

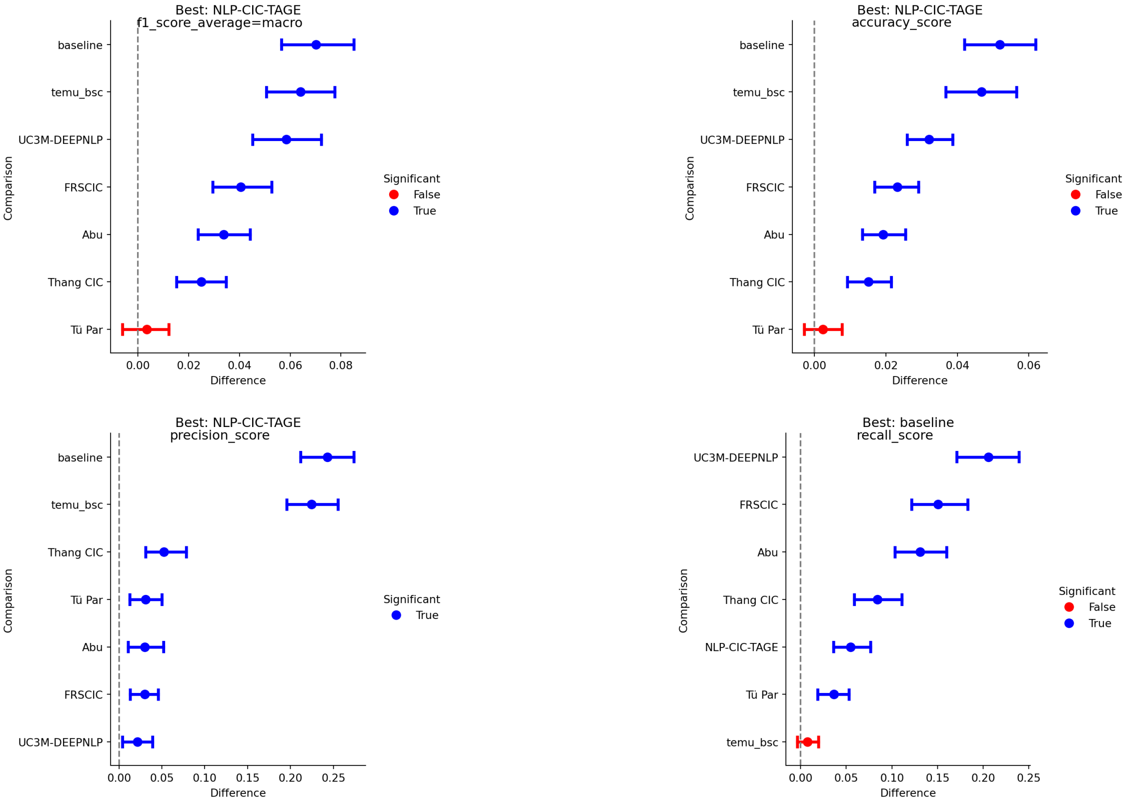


Figure A.4: Ordered bootstrap confidence intervals for performance differences in multi-metric evaluations. The visualization highlights performance competitiveness relative to the leading system.

for each metric, the methodology ensures that significant differences are identified nuancedly, capturing the strengths and weaknesses of the compared systems under different evaluation criteria.

```
for metric, diffs in diff['winner'].items():
    print(f"\nFor␣the␣metric␣{metric}␣the␣best␣is␣{diffs['best']}")
    for key, value in diffs['p_value'].items():
        print(f"p-value␣for␣the␣difference␣with␣{key}␣{value}")
```

```
For the metric f1_score_average=macro the best is NLP-CIC-TAGE
p-value for the difference with UC3M-DEEPNLP 0.0
p-value for the difference with Abu 0.0
p-value for the difference with baseline 0.0
p-value for the difference with FRSCIC 0.0
p-value for the difference with Tü Par 0.216
p-value for the difference with Thang CIC 0.0
```

```
p-value for the difference with temu_bsc 0.0

For the metric accuracy_score the best is NLP-CIC-TAGE
p-value for the difference with UC3M-DEEPNLP 0.0
p-value for the difference with Abu 0.0
p-value for the difference with baseline 0.0
p-value for the difference with FRSCIC 0.0
p-value for the difference with Tü Par 0.208
p-value for the difference with Thang CIC 0.0
p-value for the difference with temu_bsc 0.0

For the metric precision_score the best is NLP-CIC-TAGE
p-value for the difference with UC3M-DEEPNLP 0.008
p-value for the difference with Abu 0.002
p-value for the difference with baseline 0.0
p-value for the difference with FRSCIC 0.0
p-value for the difference with Tü Par 0.0
p-value for the difference with Thang CIC 0.0
p-value for the difference with temu_bsc 0.0

For the metric recall_score the best is baseline
p-value for the difference with UC3M-DEEPNLP 0.0
p-value for the difference with Abu 0.0
p-value for the difference with NLP-CIC-TAGE 0.0
p-value for the difference with FRSCIC 0.0
p-value for the difference with Tü Par 0.0
p-value for the difference with Thang CIC 0.0
p-value for the difference with temu_bsc 0.106
```

Finally, the p-values for each metric are corrected using a Bonferroni correction, demonstrating a rigorous approach to evaluating significance across multiple tests. This adjustment minimizes the likelihood of false positives and provides a robust framework for assessing the statistical reliability of performance differences across various evaluation criteria.

```python
correction = 'bonferroni'
for metric, diffs in diff['winner'].items():
    print(f"\nFor␣the␣metric␣{metric}␣the␣best␣is␣{diffs['best']}")
    result = multipletests(list(diffs['p_value'].values()),
    method=correction)
    p_valuesC = dict(zip(diffs['p_value'].keys(),result[1]))
    for key, value in p_valuesC.items():
        print(f'{key},␣p-value␣corrected␣by␣{correction}␣=␣{value}')
```

```
For the metric f1_score_average=macro the best is NLP-CIC-TAGE
UC3M-DEEPNLP, p-value corrected by bonferroni = 0.0
Abu, p-value corrected by bonferroni = 0.0
baseline, p-value corrected by bonferroni = 0.0
FRSCIC, p-value corrected by bonferroni = 0.0
Tü Par, p-value corrected by bonferroni = 1.0
Thang CIC, p-value corrected by bonferroni = 0.0
temu_bsc, p-value corrected by bonferroni = 0.0

For the metric accuracy_score the best is NLP-CIC-TAGE
UC3M-DEEPNLP, p-value corrected by bonferroni = 0.0
Abu, p-value corrected by bonferroni = 0.0
baseline, p-value corrected by bonferroni = 0.0
FRSCIC, p-value corrected by bonferroni = 0.0
Tü Par, p-value corrected by bonferroni = 1.0
Thang CIC, p-value corrected by bonferroni = 0.0
temu_bsc, p-value corrected by bonferroni = 0.0

For the metric precision_score the best is NLP-CIC-TAGE
UC3M-DEEPNLP, p-value corrected by bonferroni = 0.056
Abu, p-value corrected by bonferroni = 0.014
baseline, p-value corrected by bonferroni = 0.0
FRSCIC, p-value corrected by bonferroni = 0.0
Tü Par, p-value corrected by bonferroni = 0.0
Thang CIC, p-value corrected by bonferroni = 0.0
temu_bsc, p-value corrected by bonferroni = 0.0

For the metric recall_score the best is baseline
UC3M-DEEPNLP, p-value corrected by bonferroni = 0.0
```

```
Abu, p-value corrected by bonferroni = 0.0
NLP-CIC-TAGE, p-value corrected by bonferroni = 0.0
FRSCIC, p-value corrected by bonferroni = 0.0
Tü Par, p-value corrected by bonferroni = 0.0
Thang CIC, p-value corrected by bonferroni = 0.0
temu_bsc, p-value corrected by bonferroni = 0.742
```

These tools ensure rigorous statistical evaluation across various metrics and systems.