Harnessing the synergy between the cumulative link model and the SLACE loss in ordinal classification

Francisco Bérchez-Moreno[1⋆][0000−0003−1648−2780], Víctor M. Vargas[2][0000−0002−0700−275X], Miguel Á. Contreras-Córdoba[1][0009−0006−2977−903X], Juan C. Fernández[1][0000−0001−8849−6036], and Pedro A. Gutiérrez[1][0000−0002−2657−776X]

[1] Departamento de Ciencia de la Computación e Inteligencia Artificial, Universidad de Córdoba, Córdoba, España.
{fberchez,macontreras,jfcaballero,pagutierrez}@uco.es

[2] Departamento de Teoría de la Señal y Comunicaciones, Universidad de Alcalá, Madrid, España.
victor.vargas@uah.es

Abstract. Ordinal classification arises in many real-world scenarios where labels follow a natural order, yet most standard classifiers ignore this structure. The Cumulative Link Model (CLM) is a probabilistic approach for modelling ordinal outcomes, where a projection function uncovers an assumed latent variable divided into ordered categories by thresholds. Meanwhile, Soft Labels Accumulating Cross-Entropy (SLACE) is a recent loss function that likely satisfies monotonicity and balance sensitivity, showing strong performance when used with eXtreme Gradient Boosting (XGB-SLACE). This work proposes MLP-CLM-SLACE, an ordinal MultiLayer Perceptron (MLP) that integrates a CLM output layer with the SLACE loss. This combination enables the model to capture both ordinal relationships among classes and class-specific distance penalties during training. We conduct an empirical evaluation across 46 datasets from the TOCUCO repository, comparing our approach with eight state-of-the-art ordinal and nominal methods. MLP-CLM-SLACE consistently outperforms these methods across ordinal metrics, with statistical tests confirming a significant improvement over XGB-SLACE. Overall, MLP-CLM-SLACE provides a robust foundation for ordinal classification.

Keywords: Machine Learning · Ordinal classification · Output layers · CLM · Loss functions · SLACE.

1 Introduction

Many real-world problems require predicting categories that follow a natural progression or severity scale. This setting, commonly referred to as Ordinal Classification (OC) [15], differs from nominal classification because the labels have

⋆ Corresponding author: fberchez@uco.es

an inherent order. A typical example is credit-risk assessment [11], where labels such as "low risk", "medium risk", and "high risk" reflect an increasing level of financial uncertainty. OC applications are present in many other areas such as atmospheric events prediction [14] or time-series forecasting [2].

Despite its relevance, modelling ordinal relationships remains challenging. Nominal classification models treat categories as independent, ignoring their natural order. As a result, all misclassification errors are penalised equally, even though, for example, misclassifying a sample by two levels should be more severe than an error in an adjacent class. These limitations commonly arise from both the loss functions and the output layers typically used in standard classifiers, which do not consider the ordered nature of the labels and, therefore, fail to exploit the ordinal information. For instance, the widely used Categorical Cross-Entropy (CCE) [12] loss function assumes that all classes are equally distant. In the same way, the softmax output layer produces a probability distribution over unordered categories, instead of producing unimodal probability distributions, which are typical in ordinal classification problems. Together, such components prevent neural networks from explicitly learning the underlying ordinal structure, often leading to suboptimal performance in ordinal problems.

Existing approaches attempt to model ordinal structure using specialized losses, such as SORD [10], SLACE [16] or Weighted Kappa [20], which introduce distance-aware penalties or soft labels, or through ordinal output layers like the CLM [20], which impose a monotonic threshold structure. These methods aim to ensure that class order is encoded in the optimisation process, penalising distant errors more severely and better leveraging the inherent order in the labels. Although many of these formulations require modifications to the architecture or training objective, such adaptations are designed to strengthen the model's ability to capture ordinal relationships and ultimately improve predictive performance. Consequently, they represent effective alternatives to nominal classification, offering ways to encode ordinal structure across different types of models.

Among the different ordinal losses proposed in the literature, SLACE [16] provides a way to incorporate distance-sensitive penalties by assigning smoothly decaying weights to neighbouring classes, ensuring that errors grow progressively with ordinal distance. This loss is shown to satisfy two important desirable properties: monotonicity (it penalises misclassifications that are further away from the true class more severely) and balance sensitivity (minority classes are not overshadowed by majority ones).

In the original paper [16], the loss is applied to train a nominal model, namely XGBoost. We hypothesise that using an ordinal model with an output function that ensures a unimodal probability distribution can further boost performance and better exploit the properties of SLACE. The present work follows this principle by integrating an ordinal loss function (SLACE) with an ordinal output layer that enforces unimodal predictions (CLM) within a standard MLP. This results in a fully ordinal model that leverages the inherent order of the problem both in the predicted probability distribution and in the objective function.

The remainder of this paper is structured as follows. Section 2 provides the necessary background on ordinal classification, ordinal loss functions and output layers. Section 3 introduces the proposed MLP-CLM-SLACE model. Section 4 details the experimental setup, describing the datasets, training configuration, and evaluation metrics. Results and statistical analysis are presented in Section 5, followed by conclusions in Section 6.

2 Background

This section provides the necessary background to contextualise the paper. It first defines ordinal classification and its key characteristics, then outlines the role of loss functions in nominal and ordinal settings, and finally introduces ordinal output layers that exploit label ordering.

2.1 Ordinal Classification

In machine learning, a classification problem with J classes refers to the task of assigning the correct label $y_i \in \mathcal{Y} = \{\mathcal{C}_1, \mathcal{C}_2, ..., \mathcal{C}_J\}$ to a given pattern defined by an input vector $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$, where d is the number of input variables. Thus, a classification dataset is given by $\mathcal{D} = \langle \mathbf{X}, \mathbf{y} \rangle = \langle (\mathbf{x_i}, y_i);\ i = 1, 2, \ldots, N \rangle$, where N is the number of patterns in the dataset, $\mathbf{X} = \langle \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N \rangle$ is the input tensor, and $\mathbf{y} = \langle y_1, y_2, \ldots, y_N \rangle$ is the vector of labels. In OC [15] the output categories exhibit a natural ordering determined by the real-world problem, unlike in nominal classification where classes are unordered. This order between categories is often represented as $\mathcal{C}_1 \prec \mathcal{C}_2 \prec ... \prec \mathcal{C}_J$, where $\prec$ expresses the order relationship between the classes without defining a specific distance between them, as it is usually unknown. When a classifier is applied to this dataset, it produces a set of predictions $\{\hat{y}_1, \ldots, \hat{y}_N\}$.

To facilitate the use of ordinal evaluation metrics, each class is mapped to its corresponding position on the ordinal scale using the function $\mathcal{O}(\hat{y}_i)$. Specifically, the mapping assigns $\mathcal{O}(\mathcal{C}_j) = j$ to the class $\mathcal{C}_j$, so that if $y_i = \mathcal{C}_j$ then $O(y_i) = j$, for $1 \leq j \leq J$. This numerical encoding provides a straightforward way to quantify the severity of prediction errors according to the natural order of the classes. For simplicity, throughout this work, we will refer to each ordinal category interchangeably as either $\mathcal{C}_j$ or its numeric counterpart $\mathcal{O}(\mathcal{C}_j) = j$, depending on the context.

2.2 Loss functions

In classification problems where the categories have no underlying ordering, the Categorical Cross-Entropy (CCE) loss [12] is the standard objective function used to train different models such as neural networks. CCE is computed as:

$$\mathscr{L}_{\text{CCE}}(\mathbf{X}, \mathbf{y}) = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{J} p_{y_i,j} \left[-\log \mathrm{P}(y = \mathcal{C}_j \mid \mathbf{x}_i) \right], \tag{1}$$

where y_i is the target category of the i-th instance, and $p_{y_i,j} = \mathbb{1}\{y_i = \mathcal{C}_j\}$ is the i-th element of a one-hot encoding of the target label, y_i. The term $\mathrm{P}(y = \mathcal{C}_j \mid \mathbf{x}_i)$ corresponds to the predicted probability for class $\mathcal{C}_j$, estimated by the model and typically obtained after applying a softmax to the output of the model. Because CCE only penalises the probability estimated for the correct class, incorrect prediction receives the same treatment, regardless of its relative position in the ordinal scale.

For OC problems, the literature includes various loss formulations designed to account for the ordinal relationships between classes. Examples include the Weighted Kappa (WK) function [20], SORD loss [10], and SLACE loss [16]. These approaches incorporate information about the relative distances between categories, typically through distance-aware penalties or soft labels, so that errors involving classes farther from the true label are penalised more severely than errors in adjacent classes. Specifically, the SLACE loss function is considered for the methodology proposed in this work. Thus, it is further described in Section 3.

2.3 Output functions

A common choice for the output layer of classification models, that disregards the ordinal nature of the labels, is the softmax function, which normalises a vector of J logits into a probability distribution over the J classes.

In ordinal classification, it is common to assume that ordinal class labels arise from a continuous latent variable obtained by projecting the input onto a one-dimensional space [1]. Threshold-based models are typically used to capture this relationship by discretising the latent variable into ordered categories. The idea of using a CLM layer as the output layer of a neural network has been previously proposed, showing strong empirical performance while ensuring unimodal predictive distributions [20]. This unimodality is particularly well suited to ordinal classification problems, as it reflects the assumption that probability mass should concentrate around a single most plausible class and decrease monotonically as predictions move further away from it, thereby respecting the inherent order of the labels. To do that, these models learn a projection from the data along with a set of $J-1$ thresholds that divide the real-space where the data is projected into J intervals. Thus, the output probability is determined by the interval where the instance is projected.

When this framework is implemented within a neural network, the latent projection corresponds naturally to the output of a single neuron in the final layer, which provides a one-dimensional representation of the input. The thresholds operate on this projection to produce the class prediction.

Given that the CLM is part of the approach proposed in this work, it is described in depth in Section 3.

3 Proposed Methodology

In this section, the ordinal neural network methodology proposed in this work, namely MLP-CLM-SLACE, is described. The proposed approach consist of an MLP with a CLM output layer. The neural network model is trained using gradient-descent and the SLACE loss function as objective.

Traditionally, in nominal classification problems, an MLP employs a softmax output layer and the CCE loss function, which discards the order information, potentially resulting in a loss of performance in terms of OC metrics. To determine the output probabilities taking into account the order information, we replace the softmax layer by a CLM output layer. When using the CLM, the probability of the i-th instance ($\mathbf{x}_i$) belonging to each of the classes is obtained as a function of cumulative probability:

$$\mathrm{P}(\mathrm{y} = \mathcal{C}_1 \mid \mathbf{x}_i) = \mathrm{P}(y \preceq \mathcal{C}_1 \mid \mathbf{x}_i), \tag{2}$$

$$\mathrm{P}(\mathrm{y} = \mathcal{C}_j \mid \mathbf{x}_i) = \mathrm{P}(y \preceq \mathcal{C}_j \mid \mathbf{x}_i) - \mathrm{P}(y \preceq \mathcal{C}_{j-1} \mid \mathbf{x}_i), \tag{3}$$

$$\mathrm{P}(\mathrm{y} = \mathcal{C}_J \mid \mathbf{x}_i) = 1. \tag{4}$$

The cumulative probabilities are computed as follows:

$$\mathrm{P}(\mathrm{y} \preceq \mathcal{C}_j \mid \mathbf{x}_i) = g(b_j - f(\mathbf{x}_i)), \quad j = 1, \ldots, J, \tag{5}$$

where $g(\cdot)$ is the link function, and $\mathbf{b} = \langle b_0, b_1, b_2, \ldots, b_J \rangle$ is a non-decreasing thresholds vector. Thus, the CLM layer uses the output of the preceding layer (with one neuron), $f(\mathbf{x}_i)$, as the projection that represents the latent variable.

On the other hand, the loss function used to train the model is the SLACE loss [16]. SLACE is specifically designed for ordinal regression tasks: it accumulates the predicted probabilities of the true class and neighbouring categories while weighting them according to their proximity to the correct label. This ensures that misclassifications to classes further away are penalised more heavily than those to nearby classes. It is defined as follows:

$$\mathscr{L}_{\mathrm{SLACE}}(\mathbf{X}, \mathbf{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{J} q(\mathbf{x}_i)_j \cdot \log\left(\sum_{k=1}^{J} P_{y_i}(\mathcal{C}_j, \mathcal{C}_k) \cdot \mathrm{P}(y = \mathcal{C}_k \mid \mathbf{x}_i) \right), \tag{6}$$

where $\mathrm{P}(y = \mathcal{C}_k \mid \mathbf{x}_i)$ is the predicted probability and:

$$q(\mathbf{x}_i)_j = \frac{\exp\left(-\alpha \cdot |j - \mathcal{O}(y_i)|\right)}{\sum_{k=1}^{J} \exp\left(-\alpha \cdot |j - \mathcal{O}(y_i)|\right)}, \tag{7}$$

and:

$$P_{y_i}(\mathcal{C}_j, \mathcal{C}_k) = \begin{cases} 1, & |j - \mathcal{O}(y_i)| \geq |k - \mathcal{O}(y_i)|, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

By combining the CLM output layer with the SLACE loss, the proposed MLP becomes a model that is fully aware of the ordinal nature of the target labels, both in the generation of class probabilities and in the optimisation process. This integration provides a coherent learning signal and enables the network to exploit ordinal relationships between classes more effectively than standard nominal MLPs.

4 Experimental settings

This section details the experiments carried out to compare the proposed MLP-CLM-SLACE methodology using a set of performance metrics. First, the 46 datasets employed in the study are introduced. Next, the model training procedure is detailed, including the hyperparameter tuning process. Finally, the evaluation metrics used to assess model performance are outlined. All experiments are conducted using the `dlordinal` Python library [4], which is openly available on GitHub[3].

4.1 Datasets

This work employs a total of 46 datasets, collectively referred to as TOCUCO [3], which are publicly available at the AYRNA website[4]. The TOCUCO collection comprises 24 discretised regression datasets as well as 22 datasets containing ordered categorical variables.

4.2 Evaluated Methodologies

The proposed MLP-CLM-SLACE methodology is compared against a set of well-established ordinal and nominal classification techniques. Table 1 summarises the main characteristics of each method, highlighting the model type and whether it incorporates ordinal information through the loss function or output layer.

4.3 Models training

The experimental procedure involves training every model configuration on the 46 datasets included in TOCUCO, using 30 independent seeds to capture stochastic variation and strengthen the reliability of the results. We used the already generated fixed stratified holdout partition provided in TOCUCO [3] (70% for training and 30% for testing).

For each seed, the training portion is further resampled in a stratified manner, and a randomised search is used to explore the hyperparameter space, evaluating a maximum of 20 randomly sampled configurations. The ranges of values considered for each hyperparameter during the tuning process are detailed in Table 2 guided by the Average Mean Absolute Error (AMAE) metric [13].

[3] `https://github.com/ayrna/dlordinal`
[4] `https://www.uco.es/grupos/ayrna/tocuco`

Table 1. Summary of evaluated methodologies.

Method	Model Type	Ordinal
Ridge [18]	Multi-output regression	No
RF [7]	Ensemble of decision trees	No
XGB [8]	Gradient boosting trees	No
XGB-SLACE [16]	XGB with SLACE loss function	Yes
LogAT-cwBal [17]	Logistic All-Threshold classifier (CLM)	Yes
MLP [6]	Multi-Layer Perceptron (softmax)	No
MLP-CLM [20]	MLP with CLM output layer	Yes
MLP-SLACE	MLP with SLACE loss function	Yes
MLP-CLM-SLACE	MLP with CLM output layer and SLACE loss function	Yes

Table 2. Hyperparameter values used for cross-validation of each methodology.

Method	Hyperparameter	Values
Ridge, LogAT-cwBal	Regularisation strength	$\{10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{1}, 10^{2}, 10^{3}\}$
	Maximum iterations	$\{1000, 1500, 3000, 5000\}$
RF	Max depth	$\{3, 5, 8\}$
	Estimators	$\{100, 250, 500, 1000\}$
	Minimal Pruning	$\{0.0, 0.05, 0.1\}$
	Feature subsample	{No, squared root}
	Bootstrap	{No, Yes}
XGB, XGB-SLACE	Max depth	$\{3, 5, 8\}$
	Maximum iterations	$\{100, 250, 500, 1000\}$
	Learning rate	$\{0.01, 0.05, 0.10\}$
	Pattern subsample (%)	$\{0.75, 0.95, 1.0\}$
	Feature subsample (%)	$\{0.75, 0.95, 1.0\}$
MLP, MLP-CLM-SLACE, MLP-SLACE, MLP-CLM	Hidden units	$\{5, 8, 10, 15, 20, 50, 100\}$
	Maximum iterations	$\{1000, 1500, 3000, 5000\}$
	Learning rate	$\{10^{-5}, 10^{-4}, 10^{-3}\}$
MLP-CLM-SLACE, MLP-SLACE, XGB-SLACE	Alpha	$\{0.3, 0.5, 0.8, 1, 2, 3\}$ $4, 7, 10, 15, 20, 25\}$
MLP-CLM	Minimum distance	$\{0.0, 0.1, 0.2\}$

4.4 Model evaluation

Four evaluation metrics are considered to assess the performance of the methodologies compared in this work. It is important to use performance measures that account for both the ordinal nature of the classes and the imbalance in their distributions. The metrics considered in this study are:

1. Quadratic Weighted Kappa (QWK) [5] is an ordinal evaluation metric derived from the Kappa coefficient (κ). It penalises misclassifications proportionally to the distance between the predicted and true classes, using a weight matrix that emphasises more severe errors.
2. Average Mean Absolute Error (AMAE) [13] is an ordinal metric that averages the mean absolute error of each class, making it suitable for imbalanced datasets.
3. Maximum Mean Absolute Error (MMAE) [9] computes the MAE separately for each class and then reports the maximum value. It is particularly useful

in imbalanced settings where poor performance on minority classes should not be overlooked.
4. Balanced Accuracy (BA) [19] is a nominal evaluation measure designed to mitigate inflated performance on imbalanced datasets. It is calculated as the product of the sensitivities per class.

5 Results

In this section, we present the experimental results of the evaluated methodologies across the metrics introduced in Section 4.4. Table 3 shows the mean value and the standard deviation of the results obtained for the test set using each of the methodologies across all the seeds and datasets. Therefore, each mean value represents the mean of 1380 values (30 runs $\times$46 datasets).

Table 3. Ranking results of each method and evaluation metric (Mean and Standard Deviation, Mean_{SD}). The best result is shown in bold and the second best one in italics.

Methodology	QWK	AMAE	MMAE	BA
Ridge	$8.022_{1.612}$	$8.522_{1.278}$	$8.674_{0.990}$	$8.413_{1.392}$
RF	$6.554_{2.151}$	$6.554_{2.191}$	$6.467_{1.979}$	$5.815_{2.341}$
XGB	$6.011_{1.893}$	$5.728_{1.960}$	$5.598_{2.070}$	$4.533_{2.358}$
XGB-SLACE	$4.783_{2.666}$	$3.935_{2.255}$	$4.022_{2.092}$	$4.391_{2.155}$
LogAT-cwBal	$\mathit{4.022_{2.206}}$	$5.217_{2.190}$	$\mathit{3.957_{2.241}}$	$5.174_{2.434}$
MLP-softmax	$4.326_{2.582}$	$4.500_{2.420}$	$4.957_{2.366}$	$\mathbf{3.500_{2.465}}$
MLP-CLM	$4.457_{2.137}$	$4.217_{1.943}$	$4.543_{2.297}$	$5.478_{2.116}$
MLP-SLACE	$4.109_{1.792}$	$\mathit{3.848_{1.763}}$	$4.152_{1.850}$	$\mathit{3.609_{1.949}}$
MLP-CLM-SLACE	$\mathbf{2.717_{1.822}}$	$\mathbf{2.478_{1.670}}$	$\mathbf{2.630_{1.818}}$	$4.087_{2.199}$

The ranking results presented in Table 3 indicate that the proposed MLP-CLM-SLACE model achieves the best average rank in the ordinal-aware metrics, namely QWK, AMAE, and MMAE. These results demonstrate its effectiveness at capturing the ordinal structure of the target labels. Although BA is not the metric in which MLP-CLM-SLACE attains the best rank, its average ranking remains competitive with respect to the best-performing methodologies.

These findings highlight the strength of our approach in optimising metrics that explicitly account for class ordering, which are more relevant in ordinal classification tasks than nominal accuracy measures. Furthermore, the proposed model proves to be robust and highly effective across the 46 datasets from TOCUCO, ranging from problems with as few as 3 classes to those with up to 10 classes, some exhibiting moderate class imbalance. MLP-CLM-SLACE consistently achieves the best or near-best rankings, demonstrating not only its adaptability to datasets of varying complexity and size but also its robustness to class imbalance. Overall, these results confirm that the integration of a CLM output layer with the SLACE loss effectively leverages ordinal information to improve predictive performance.

To further support these observations, a direct comparison was conducted between the proposed and best-performing ordinal methodology, MLP-CLM-SLACE, and the previous methodology from the literature, XGB-SLACE [16]. This analysis, illustrated in Figure 1, presents a scatter plot for the QWK metric, where each point corresponds to a different dataset, totalling 46 points. The number of wins, draws, and losses between the two approaches is also reported, alongside Wilcoxon and Student-t tests results.

The results clearly show that MLP-CLM-SLACE outperformed XGB-SLACE, achieving 34 wins versus 12 and a higher mean score (0.747 vs. 0.716). As observed in Figure 1, both the Wilcoxon test on medians and the Student test on means indicate significant differences between the two methodologies in the QWK metric.

This comparison highlights that our approach improves upon the original SLACE proposal across most evaluated metrics, particularly in capturing the ordinal relationships among classes, while maintaining competitive performance in standard classification measures. To reinforce these conclusions, a statistical analysis is presented in the following subsection.

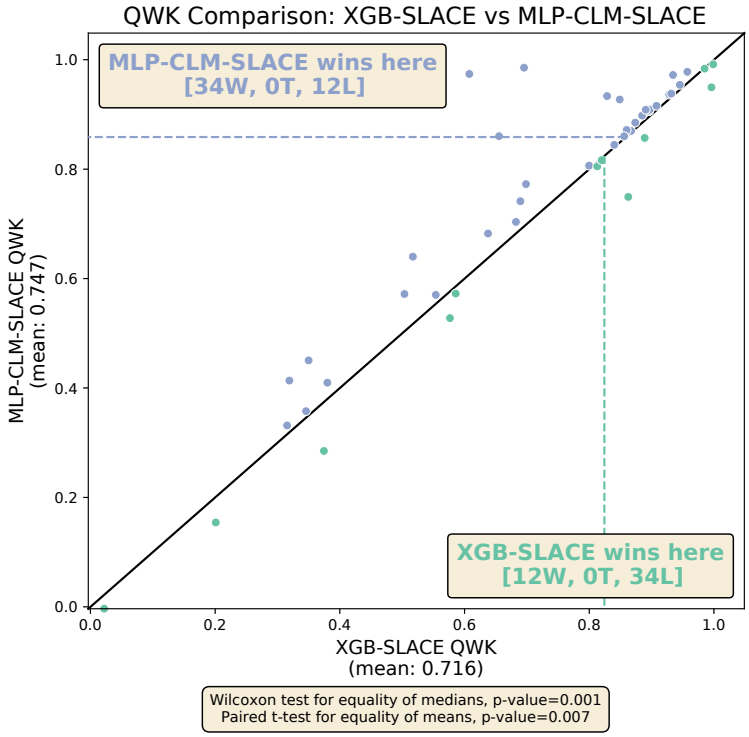


Fig. 1. QWK comparison between XGB-SLACE and MLP-CLM-SLACE estimators

5.1 Statistical Analysis

To assess whether the differences in the AMAE metric are statistically significant across the considered factors, we first perform a non-parametric Kolmogorov–Smirnov (K–S) test ($\alpha = 0.050$) to evaluate the normality of the results from different random seeds. The test indicates that normality holds in 86% of the cases. Based on this, we conduct a two-way ANOVA with Estimators and Datasets as factors. Results show that both factors significantly affect AMAE ($p < 0.001$), indicating notable variation across datasets and a strong influence of the estimator choice. A significant Datasets × Estimators interaction ($p < 0.001$) further indicates that relative performance is not consistent across datasets. Overall, both main effects and their interaction substantially impact AMAE.

Following the significant effects detected by the two-way ANOVA, a post-hoc Tukey test is conducted to determine which estimators differ significantly in AMAE. As reported in Table 4, the methods are grouped into eight overlapping subsets (S1–S8), where estimators appearing within the same subset do not exhibit statistically significant differences in performance. The proposed MLP-CLM-SLACE methodology constitutes the sole member of the first subset, indicating that it achieves the best AMAE performance among all evaluated approaches. This is followed by the MLP-SLACE methodology, while XGB-SLACE appears in both the second and third subsets, reflecting an intermediate position between these groups. Overall, ordinal classification methodologies tend to appear in better-ranked subsets than nominal classification approaches. In particular, the performance of MLP-CLM improves substantially when combined with the SLACE loss function, as implemented in the proposed methodology. Finally, the eighth subset (S8) contains the Ridge model, which is expected given its simplicity.

Table 4. Tukey test results for the AMAE metric in terms of the two factors considered.

	Subsets							
Group	S1	S2	S3	S4	S5	S6	S7	S8
MLP-CLM-SLACE	0.678							
MLP-SLACE		0.714						
XGB-SLACE		0.723	0.723					
MLP-CLM			0.726					
MLP				0.750				
LogAT-cwBal					0.783			
XGB						0.813		
RF							0.863	
Ridge								1.267

6 Conclusions

In this work, we introduced a new ordinal classification methodology, MLP-CLM-SLACE, which combined an ordinal output layer based on a threshold

model (CLM) with an ordinal loss function based on soft labels (SLACE loss) to better exploit the inherent ordering of the target variable. The proposed approach was evaluated against eight state-of-the-art estimators on a large and diverse benchmark comprising 46 datasets from TOCUCO, enabling a rigorous and comprehensive assessment of its performance.

Across this extensive experimental study, the results showed that MLP-CLM-SLACE consistently achieved the best overall performance. The method outperformed all other alternatives on ordinal metrics, and statistical tests further supported these findings. In particular, the Tukey test reported significant differences compared to all other methodologies across all datasets regarding the AMAE metric, while the Wilcoxon and Student tests indicated that MLP-CLM-SLACE significantly improved the QWK metric when compared to the original XGB-SLACE model. These results confirmed not only its competitiveness but also its superiority over the methodology from which it was derived. Moreover, the statistical analysis showed that the proposed approach, along with the other OC alternatives, generally outperformed nominal classification methodologies.

Overall, these results demonstrate the effectiveness of MLP-CLM-SLACE in modelling ordinal relationships, outperforming existing methods across multiple metrics. They highlight the benefit of combining the CLM output layer with the SLACE loss function, and suggest that further improvements could be achieved by extending this approach to other architectures or application domains.

Acknowledgments

This study has been supported by the "Agencia Estatal de Investigación (España)" (grant ref.: PID2023-150663NB-C22 / AEI / 10.13039 / 501100011033), and by the University of Córdoba and Junta de Andalucía (grant ref.: PP2F_L1_15). F. Bérchez-Moreno has been supported by "Plan Propio de Investigación Submodalidad 2.2 Contratos predoctorales" of the University of Córdoba. Víctor Manuel Vargas aws supported by the Ministerio de Ciencia, Innovación y Universidades, the Agencia Estatal de Investigación and the European Social Fund Plus (grant ref.: MICIU/AEI/10.13039/501100011033, JDC2024-054787-I).

References

1. Agresti, A.: Analysis of ordinal categorical data. John Wiley & Sons (2010)
2. Ayllón-Gavilán, R., Guijo-Rubio, D., Gutiérrez, P.A., Bagnall, A., Hervás-Martínez, C.: Convolutional and deep learning based techniques for time series ordinal classification (2023)
3. Ayllón-Gavilán, R., Guijo-Rubio, D., Gómez-Orellana, A.M., Guijo-Rubio, D., Bérchez-Moreno, F., Vargas-Yun, V.M., Gutiérrez, P.A.: Toc-uco: a comprehensive repository of tabular ordinal classification datasets (2025), https://arxiv.org/abs/2507.17348
4. Bérchez-Moreno, F., Ayllón-Gavilán, R., Vargas, V.M., Guijo-Rubio, D., Hervás-Martínez, C., Fernández, J.C., Gutiérrez, P.A.: dlordinal: A python package for

deep ordinal classification. Neurocomputing p. 129305 (2025). https://doi.org/10.1016/j.neucom.2024.129305
5. Berchez-Moreno, F., Fernandez, J.C., Hervas-Martinez, C., Gutierrez, P.A.: Fusion of standard and ordinal dropout techniques to regularise deep models. Information Fusion **106**, 102299 (2024). https://doi.org/10.1016/j.inffus.2024.102299
6. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)
7. Breiman, L.: Random forests. Machine learning **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
8. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. p. 785–794. KDD '16, Association for Computing Machinery, New York, NY, USA (2016). https://doi.org/10.1145/2939672.2939785
9. Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.A.: Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing **135**, 21–31 (2014)
10. Diaz, R., Marathe, A.: Soft labels for ordinal regression. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4738–4747 (2019)
11. Fernandez-Navarro, F., Campoy-Munoz, P., Hervas-Martinez, C., Yao, X., et al.: Addressing the eu sovereign ratings using an ordinal regression approach. IEEE transactions on cybernetics **43**(6), 2228–2240 (2013)
12. Ghosh, J., Gupta, S.: Adam optimizer and categorical crossentropy loss function-based cnn method for diagnosing colorectal cancer. In: 2023 international conference on computational intelligence and sustainable engineering solutions (CISES). pp. 470–474. IEEE (2023)
13. Gómez-Orellana, A., Guijo-Rubio, D., Gutiérrez, P., Hervás-Martínez, C., Vargas, V.: ORFEO: Ordinal classifier and regressor fusion for estimating an ordinal categorical target. Eng. Applications of Artificial Intelligence **133**, 108462 (2024)
14. Guijo-Rubio, D., Casanova-Mateo, C., Sanz-Justo, J., Gutierrez, P., Cornejo-Bueno, S., Hervás, C., Salcedo-Sanz, S.: Ordinal regression algorithms for the analysis of convective situations over madrid-barajas airport. Atmospheric Research **236**, 104798 (2020). https://doi.org/10.1016/j.atmosres.2019.104798
15. Gutiérrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervas-Martinez, C.: Ordinal regression methods: survey and experimental study. IEEE Transactions on Knowledge and Data Engineering **28**(1), 127–146 (2015)
16. Nachmani, I., Genossar, B., Scharf, C., Shraga, R., Gal, A.: Slace: A monotone and balance-sensitive loss function for ordinal regression. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 19598–19606 (2025)
17. Pedregosa, F., Bach, F., Gramfort, A.: On the consistency of ordinal regression methods. Journal of Machine Learning Research **18**(55), 1–35 (2017)
18. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. the Journal of machine Learning research **12**, 2825–2830 (2011)
19. Vargas, V.M., Gómez-Orellana, A.M., Gutiérrez, P.A., Hervás-Martínez, C., Guijo-Rubio, D.: Ebano: A novel ensemble based on unimodal ordinal classifiers for the prediction of significant wave height. Knowledge-Based Systems **300**, 112223 (2024)
20. Vargas, V.M., Gutierrez, P.A., Hervas-Martinez, C.: Cumulative link models for deep ordinal classification. Neurocomputing **401**, 48–58 (2020)