Weighted Optimisation in Ordinal Regression for Imbalanced Problems

Pedro J. Torres-González[1], Javier Sánchez-Monedero[1][0000−0001−8649−1709], and David Guijo-Rubio[2][0000−0002−8035−4057]

[1] Department of Computer Science and Numerical Analysis, Universidad de Córdoba, Córdoba, 14041, Spain. z32togop@uco.es, jsanchezm@uco.es

[2] Department of Signal Processing and Communications, Universidad de Alcalá, Alcalá de Henares, 28805, Spain. david.guijo@uah.es

Abstract. Ordinal regression, also known as ordinal classification, is a Machine Learning (ML) task that focuses on predicting the right category of an instance. The key distinction from the standard ML classification paradigm is that the categories in ordinal classification follow a specific, inherent order. This type of problem requires specialised learning methods and performance metrics due to its particular characteristics. A common challenge in ordinal classification is class imbalance, where one or more minority classes are underrepresented compared to the majority class or classes. This imbalance often leads to models that are biased towards the majority classes, resulting in poorer performance for the minority classes and reduced practical applicability. Logistic regression has long been the cornerstone of statistical and ML approaches for classification because of its computational efficiency, interpretability, and probabilistic modelling capabilities. Despite its widespread use, traditional logistic regression models face limitations when applied to ordinal datasets, particularly in the presence of imbalance. The Python package `mord` provides several adaptations of the logistic regression approach to ordinal classification, such as Logistic All Thresholds (LogAT) and Logistic Immediate Thresholds (LogIT). However, `mord` lacks of built-in mechanisms to address imbalance explicitly. The present work aims to implement and assess the integration of class weights into logistic ordinal models extending `mord`. Moreover, this study will explore the impact of imbalance on model performance through a range of experiments including a extensive collection of datasets. The goal is to enhance the predictive accuracy and robustness of ordinal logistic models in imbalanced scenarios by integrating class-specific weight adjustments.

Keywords: Class weights optimisation · Imbalanced problems · Logistic regression · Ordinal classification, `mord`.

1 Introduction

Machine Learning (ML) [2] is a broad field encompassing various tasks aimed at deriving insights from data and making predictions. These tasks include classification, regression, clustering, and more.

Classification involves categorising data into predefined classes, regression focuses on predicting continuous outcomes, and clustering is used for grouping similar data points together. Our research specifically focuses on ordinal classification, also known as ordinal regression, because our problem involves predicting outcomes that fall into distinct categories with an inherent order.

The order of classes in ordinal classification represents varying levels of severity, quality, or other gradations, making it particularly valuable in applications where the relative positioning of categories matters. In such problems, the severity of a misclassification is linked to the distance between classes; for instance, misclassifying an example from class 2 into class 8 is far more severe than misclassifying it into class 3, due to the greater deviation from the correct class. This type of classification is prevalent in many fields, such as medical diagnosis [13] (e.g., stages of cancer progression), customer satisfaction surveys (e.g., rating satisfaction from 'very dissatisfied' to 'very satisfied'), and risk assessment [1] (e.g., categorising risk levels from 'low' to 'high').

A key characteristic of ordinal classification is that datasets often exhibit class imbalance. This imbalance occurs when some classes are significantly underrepresented compared to others, which can adversely affect model performance. This is a well-documented issue in the literature [16], where various methods have been proposed to address it, but solutions for linear models in ordinal classification are still limited.

Meanwhile logistic regression models have long been foundational in both statistical research and industry, thanks to their effectiveness in modelling relationships between input variables and categorical responses. Traditionally, logistic regression has been employed for binary and multiclass classification. However, given the focus of this research on ordinal classification, we aim to refine ordinal logistic regression by addressing the challenges posed by class imbalance.

Our objective is to enhance the framework of ordinal logistic regression by introducing a weight optimisation method that prioritises errors in the minority classes while penalising the majority ones. This approach aims to improve the model's robustness when considering imbalanced aware metrics.

To demonstrate the effectiveness of this enhanced models, we conduct experiments on both real-world ordinal datasets and discretized regression datasets. The latter involves converting continuous variables into discrete ordinal labels using various discretization strategies, providing a controlled environment for testing. Furthermore, we will compare the performance of our model against established methods from previous studies [14]to assess its relative effectiveness in handling imbalanced ordinal classification.

2 Ordinal Classification

2.1 General Description

Classification is a fundamental task in ML where the goal is to assign a label to an input instance from a predefined set of categories. Among various classification techniques, logistic regression [17] stands out due to its simplicity and efficiency. Traditionally used for binary classification and ported to multiclass through ensemble strategies [9], logistic regression models the probability that a given instance belongs to a particular class using a logistic function. In binary logistic regression, the model predicts the probability of an instance belonging to one of two classes, typically labelled as 0 and 1. This probability is modelled as a function of the input features using the logistic function.

2.2 Mathematical Definition of the Classification Problem

The classification problem can be formally defined as follows [17]: given a dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, where $\mathbf{x}_i \in \mathbb{R}^d$ represents the d-dimensional feature vector of the i-th instance and $y_i \in \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ is its corresponding class label. Here, K represents the number of classes, while N denotes the number of instances. The goal is to learn a function $f : \mathbb{R}^d \to \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ that maps each feature vector $\mathbf{x}_i$ to a class label y_i.

2.3 Mathematical Definition of the Ordinal Classification Problem

Ordinal classification, also known as ordinal classification, is a type of classification where the class labels have a natural order. However, unlike in traditional classification, the distances between the classes are not necessarily equal or known.

Formally, in ordinal classification, the task is to learn a function

$$f : \mathbb{R}^d \to \{\mathcal{C}_1, \ldots, \mathcal{C}_k\}, \tag{1}$$

where the labels have an inherent order such that $\mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_K$. Here, K represents the number of classes, while N denotes the number of instances. The challenge is to correctly predict the ordered class label while considering this ordinal nature of the classes.

2.4 Mathematical Model of Logistic Regression (Binary and Multinomial)

Binary Logistic Regression. In binary logistic regression, the probability that an instance $\mathbf{x}$ belongs to the class $y = \mathcal{C}_1$ is modelled as:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + \exp(-\mathbf{w}^\top \mathbf{x} - b)} \tag{2}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ is the bias term. The predicted class label $\hat{y}$ is then given by:

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1 \mid \mathbf{x}) \geq \sigma \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where σ is the threshold set, it is normally 0.5.

Multinomial Logistic Regression. Multinomial logistic regression extends the binary case with the one-vs-all strategy [9] to handle multiple classes ($K > 2$). The probability of an instance $\mathbf{x}$ belonging to class k is given by:

$$P(y = \mathcal{C}_k \mid \mathbf{x}) = \frac{\exp(\mathbf{w}_k^\top \mathbf{x} + b_k)}{\sum_{j=1}^{K} \exp(\mathbf{w}_j^\top \mathbf{x} + b_j)} \tag{4}$$

where $\mathbf{w}_k$ and b_k are the weight vector and bias term for class k. The predicted class label $\hat{y}$ is the one with the highest probability:

$$\hat{y} = \arg \max_{k \in \{1,\ldots,K\}} P(y = \mathcal{C}_k \mid \mathbf{x}) \tag{5}$$

3 Background

This study builds upon the comprehensive review of ordinal classification methods by Gutiérrez *et al.* [15]. Their review encompasses a wide range of models categorised as Naïve approaches [18] (Support Vector Classifier, Cost-Sensitive Support Vector Classifier...), Ordinal Binary decompositions (Neural Network[3], Extreme Learning Machine[6]...) and Threshold models (Proportional Odds Model[21], Support Vector Machines for ordinal classification with different kernels[4], etc.). They assess their performance across various datasets, which are detailed later in the study. Notably, this review does not address the issue of class imbalance although it is present in many of the datasets in the study so that our article focus on this evaluation.

3.1 Class imbalance

The bias toward the majority class is a well-known problem in ML and it can be reduced either by adjusting the training data to minimise imbalance or by modifying the model's learning or decision process to increase the sensitivity to the minority class. Improvements not only depends on the mitigation technique but also on data characteristics such as dimensionality or class noise amongst others [7].

Thus, strategies for addressing class imbalance are categorised into data-level techniques, algorithm-level methods, and hybrid approaches [19]:

- **Data-level methods** [22]:
  - **Resampling Techniques**: these methods involve either oversampling the minority class or undersampling the majority class to produce a more balanced dataset.
  - **Synthetic Data Generation**: beyond simple oversampling, synthetic data generation methods create entirely new samples of the minority class based on its existing characteristics, helping to balance the class distribution more naturally. Techniques such as Synthetic Minority Over-sampling Technique (SMOTE) and random undersampling are frequently used although they can downgrade performance in many real scenarios [8].
- **Algorithm-level methods:** [27]
  - **Cost-sensitive Learning**: this approach assigns different costs to misclassifications, typically by penalising errors in the minority class more heavily. This helps the model to focus more on correctly predicting the minority class. This correction can be made often at the cost of degrading global accuracy [10].
  - **Ensemble Methods**: algorithms like Random Forest and Boosting can be adapted to handle imbalance by combining multiple models weighting minority class predictions more heavily. Depending on the ensemble technique, this strategy can be considered an hybrid strategy ([26]) that combine sampling with cost-sensitive learning.

Before delving into the modifications we have made, it is essential to contextualise our work within the broader landscape of ML strategies. Specifically, we focus on weighted optimisation during the learning stage applied at Logistic Regression.

3.2 LogAT and LogIT

In the context of ordinal logistic regression, two primary approaches are widely recognised [23,25,24]: Logistic All-Threshold (LogAT) and Logistic Immediate-Threshold (LogIT). Both methods are integral components of the `mord` package, a library implementing ordinal logistic regression methods, compatible with `scikit-learn`.

- **Logistic All Thresholds** (LogAT). This model assumes that there is a single set of coefficients for all classes, but different thresholds (cut-points) for each class. The idea is that the same underlying process generates the outcomes, but the thresholds at which these outcomes change differ across classes. The loss function used in this model in `mord` is the Absolute Error (AE) also known as L1 loss, which is defined as the absolute difference between the predicted value and the actual value.

$$\mathrm{AE} = |y_{\mathrm{true}} - y_{\mathrm{pred}}| \tag{6}$$

- **Logistic Immediate Thresholds** (LogIT). Similar to LogAT, this model also uses a single set of coefficients, but with the additional constraint that the thresholds must increase monotonically. This means that as the model move from one class to the next, the threshold must always increase, preserving the natural order of the categories. 0-1 Loss function is used in this model by `mord` package. This function assigns a loss of 0 for a correct prediction and a loss of 1 for an incorrect prediction, regardless of how wrong the prediction is.

$$\mathrm{L}_{0-1}(y_{\mathrm{true}}, y_{\mathrm{pred}}) = \begin{cases} 0 & \text{if } y_{\mathrm{true}} = y_{\mathrm{pred}} \\ 1 & \text{if } y_{\mathrm{true}} \neq y_{\mathrm{pred}} \end{cases} \tag{7}$$

In the current landscape of ordinal classification, various methods have been proposed to handle class imbalance, yet many linear models still struggle with this challenge. The review previously mentioned [15] reveals that while advanced techniques like support vector classifier and ensemble methods have shown promise, linear models have not been effectively adapted to address imbalanced datasets. Even when not designed to deal with imbalance, support vector machines and ensemble models can better deal with non-linear data, class overlap, amongst other data set characteristics that influence performance in an imbalanced context [7].

4 Weighted optimisation for ordinal logistic regression

In this study, we enhance the baseline ordinal classification models by incorporating class-specific weights into the `mord` package. This adjustment addresses the common challenge of class imbalance, where certain categories in ordinal data are underrepresented. By assigning higher weights to minority classes, the modified models are expected to improve their performance in imbalanced datasets, making them more robust in real-world scenarios where ordinal imbalance is prevalent.

The proposed method method extends ordinal logistic models described in Section 3.2, by incorporating a weighting mechanism to address imbalance. This mechanism calculates weights using the formula derived from the literature [20]:

$$w_i = \frac{N}{K \cdot n_i} \tag{8}$$

where w_i represents the weight for class i, N is the total number of instances, K is the number of classes, and n_i is the number of instances in class i within the training set.

To incorporate these class-specific weights into the model, we extend the traditional loss function to a weighted version. The weight vector, $\mathbf{w} = [w_1, w_2, \ldots, w_k]$, is applied to the loss function to adjust the contribution of each class in proportion to its imbalance. Specifically, for the logistic regression model, the weighted loss function L_w is given by:

$$L_w = \sum_{i=1}^{N} w_{y_i} \cdot \mathcal{L}(y_i, \hat{y}_i), \tag{9}$$

where w_{y_i} is the weight corresponding to the true class label y_i of the i-th sample, and $\mathcal{L}(y_i, \hat{y}_i)$is the logistic loss for the prediction $\hat{y}_i$. This adjustment ensures that the model gives more importance to minority classes, reducing their misclassification rates. In essence, the higher the imbalance of a class, the more weight it carries during training.

Specifically, let us examine how these adjustments impact the previously presented methods, LogAT and LogIT. Firstly, note that the new weighted version are known as LogAT_c and LogIT_c, respectively.

The LogAT_c model uses the AE as the loss function. Therefore, the weighted loss can be expressed as:

$$\text{LogAT}_\text{c} = \sum_{i=1}^{N} w_{y_i} \cdot |y_i - \hat{y}_i|, \tag{10}$$

where w_{y_i} is the weight for the class of the i-th sample, y_i is the true label, and $\hat{y}_i$ is the predicted label.

For the LogIT_c model, which uses the 0-1 loss function, the weighted loss is:

$$\text{LogIT}_\text{c} = \sum_{i=1}^{N} w_{y_i} \cdot \text{L}_{0-1}((y_i, \hat{y}_i), \tag{11}$$

where $y_i = y_{true}$ and $\hat{y}_i = y_{pred}$.

In both cases, the class weights w_{y_i} ensure that more importance is given to samples from underrepresented classes, effectively addressing class imbalance during model training.

5 Experimental Settings.

In this section, we outline the experimental setup used to assess the performance of the enhanced models. This includes a detailed description of the datasets and the evaluation metrics chosen for the comparison. We also present the baseline models against which our improved versions are tested.

5.1 Datasets

In this study, we select a diverse set of datasets from the review [14,15], to train and evaluate our ordinal classification models. There are two different groups of datasets:

- **Discrete Ordinal Datasets** characterised by a discrete ordinal structure and notable imbalance. This group provides a direct test of the method's ability to handle real-world imbalances.
- **Continuous Datasets** consists of continuous datasets that have been discretised into 5 bins.

Tables 1 and 2 provide further details on the distribution of these data and the imbalance ratio, the ratio between minority and majority classes, as defined in [12].

By applying discretisation with a specified number of bins, continuous features are transformed into ordered categories. This process divides the features into a fixed number of bins and results in datasets that are both ordinal and imbalanced, providing a suitable testing ground for the methods implemented in our study.

Below is a brief description of each dataset:

- **Automobile**. This dataset contains information about various car models, including attributes like price, engine size, horsepower, and fuel efficiency. The target variable is the car's overall evaluation, which is an ordinal variable.
- **Balance-Scale**. The balance-scale dataset simulates the behaviour of a balance scale based on the weights placed on its two arms. The target variable is the tilt direction (left, right, balanced), which is ordered.
- **Car**. This dataset evaluates cars based on several attributes such as buying price, maintenance cost, and safety features. The target variable is an ordinal evaluation of the car's overall acceptability.
- **Contact-Lenses**. This dataset involves the recommendation of contact lenses based on factors like age, prescription, astigmatism, and tear production rate. The target variable is the suitability of contact lenses, which is an ordered category.
- **ERA**. This is a dataset from a well-known ordinal classification benchmark collection. It typically involves economic and financial variables, with the target variable representing an ordered classification.
- **ESL**. Similar to ERA, the ESL dataset is another benchmark used in ordinal classification studies, with variables often relating to educational or economic status.
- **Eucaplyptus**. This dataset contains information on eucalyptus species, with attributes such as rainfall, temperature, and altitude influencing the suitability of various species. The target variable is the suitability rating, which is ordinal.
- **LEV**. This is another ordinal classification benchmark dataset. It includes variables related to labor and employment. The target variable is usually an ordered category related to employment status.
- **Newthyroid**. This dataset contains medical information used to classify thyroid function. The target variable is the thyroid diagnosis, which is ordinal (e.g., hypothyroid, normal, hyperthyroid).
- **Pasture**. This dataset involves the classification of pasture conditions based on environmental factors. The target variable is an ordinal assessment of pasture quality.
- **Squash-Stored**. This dataset pertains to the quality of stored squash over time, with various attributes affecting the quality. The target variable is the quality rating, which is ordinal.
- **Squash-Unstored**. Similar to Squash-Stored, this dataset evaluates the quality of unstored squash. The same set of attributes is used, with the target variable being an ordinal quality rating.

- **SWD**. This dataset is another ordinal classification benchmark dataset. It involves social welfare data, with the target variable representing an ordered category.
- **TAE**. Teaching Assistant Evaluation (TAE) dataset includes evaluations of teaching assistants based on several criteria. The target variable is an ordinal evaluation score.
- **Toy**. This is a synthetic dataset. It contains simple, controlled data with an ordinal target variable.
- **Wine-Quality**. This dataset contains chemical attributes of red wine, with the target variable being the wine quality score, an ordinal variable ranging from low to high quality.

Table 1. Summary of real ordinal problem datasets used in this study. The table includes the number of patterns, attributes, classes, the distribution of classes for each dataset and imbalance ratio (IR).

Dataset	# Pat.	# Attr.	# Classes	Class Distribution	IR
Contact-Lenses	24	6	3	(15,5,4)	3.75
Pasture	36	25	3	(12,12,12)	1
Squash-stored	52	51	3	(23,21,8)	2.875
Squash-Unstored	52	52	3	(24, 24,4)	6
Tae	151	54	3	(49, 50, 52)	1.06
Newthyroid	215	5	3	(30,150,35)	5
Balance-Scale	625	4	3	(288,499,288)	1.733
SWD	1000	10	4	(32,352,399,217)	12.469
Car	1728	21	4	(1210,384,69,65)	18.615
Toy	300	2	5	(35,87,79,68,31)	2.806
Eucalyptus	736	91	5	(180,107,130,214,105)	2.038
LEV	1000	4	5	(3,280,403,197,27)	134.33
Automobile	205	71	6	(10,53,681,638,199,18)	6.81
ESL	488	4	9	(2,12,38,100,116,135,62,199,4)	99.5
ERA	1000	4	9	(92,142,181,172,158,118,88,31,18)	10.056

These following datasets were discretised as explained before:

- **Pyrim**: this dataset contains information related to the biological activity of pyrimidines, which are compounds commonly used in pharmaceuticals. The target variable represents a continuous measure of activity that has been discretised for ordinal classification.
- **Auto**: the Auto dataset includes various features of automobiles, such as engine size, horsepower, and weight. The original target variable, typically the miles per gallon (MPG), was discretised to categorise fuel efficiency.
- **Triazines**: this dataset involves the chemical properties of triazine compounds and their associated biological activity. The continuous target variable indicating activity levels has been transformed into ordinal categories.
- **Abalone**: the Abalone dataset contains physical measurements of abalones, a type of shellfish, with the original goal of predicting their age. The continuous target variable, age, was discretised to create ordered age groups.
- **Housing**: the Housing dataset consists of various attributes related to housing prices in different areas. The target variable, which is the median value of owner-occupied homes, has been discretised into ordinal categories to represent different price ranges.
- **WPBC**: the Wisconsin Prognostic Breast Cancer (WPBC) dataset contains medical data used to predict the outcome of breast cancer. The continuous target variable, typically a recurrence score, was discretised to create ordinal risk categories.
- **Diabetes**: this dataset includes diagnostic measures of diabetes patients. The original target variable, which is a continuous measure related to diabetes progression, was discretised to form ordinal categories representing different progression levels.
- **Stock**: the Stock dataset contains financial data and stock prices. The continuous target variable, usually the return on stock or price change, was discretised to classify performance into ordinal categories.

– **Machine**: the Machine dataset includes attributes related to computer hardware specifications. The continuous target variable, typically the computer's performance score, was discretised into ordinal categories reflecting different performance levels.

Table 2. Summary of regression datasets used in this study. All original datasets were discretised into 5 classes. The table includes the number of patterns, attributes, classes, and the distribution of classes for each dataset. Class imbalance is generally presented in the extreme bins.

Dataset	# Pat.	# Attr.	Class Distribution	IR
Pyrim	74	27	(27, 18, 15, 10, 4)	6.75
Auto	392	7	(131, 101, 91, 59, 10)	13.1
Triazines	186	60	(85, 58, 26, 10, 7)	12.143
Abalone	4177	10	(3036, 557, 448, 129, 7)	433.71
Housing	506	13	(236, 125, 76, 38, 31)	7.613
WPBC	194	32	(67, 43, 41, 24, 19)	3.526
Diabetes	43	2	(22, 8, 6, 5, 2)	11
Stock	950	9	(272, 227, 207, 158, 86)	3.163
Machine	209	6	(152, 27, 13, 10, 7)	21.714

As observed in Table 2, discretising the continuous datasets with $n\_bins = 5$ results in an unbalanced problem where the minority class consistently becomes class 5 across all datasets. This occurs because, in the process of binning[12], the final bin often ends up containing fewer instances compared to the other bins. Consequently, class 5 becomes significantly underrepresented in each dataset, highlighting the challenge of imbalance introduced by the discretisation process.

5.2 Performance Evaluation

The performance of the novel approaches is evaluated using a set of well-established metrics in the domain of imbalance, along with classical evaluation metrics [5].

– **Mean Absolute Error (MAE)**: measures the average magnitude of errors in predictions, without considering their direction. It is the mean of the absolute differences between predicted and actual values.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{12}$$

– **Absolute Mean Absolute Error (AMAE)**: this metric is similar to MAE but focuses specifically on absolute errors for ordinal classification problems. The MAE computed only for class j can be defined as:

$$\text{MAE}_j = \frac{1}{N_j}\sum_{i=1}^{N_j}|y_i - \hat{y}_i| \quad j \in 1, \ldots, K, \tag{13}$$

where N_j is the number of instances in class $\mathcal{C}_j$, and $\hat{y}_i$ is the predicted class for the i-th sample. Then, the AMAE is defined as:

$$\text{AMAE} = \frac{1}{K}\sum_{j=1}^{K}\text{MAE}_j \tag{14}$$

Both MAE and AMAE are bounded between 0 (perfect predictions) and $K - 1$ (the largest possible misclassification).

– **Correct Correlation Ratio (CCR)**: evaluates the correlation between the predicted class probabilities and the actual classes. Higher values indicate better model performance.

$$\text{CCR} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i = \hat{y}_i) \tag{15}$$

– **Mean Zero Error (MZE)**: measures the proportion of predictions that are exactly zero, which helps in assessing the accuracy of zero predictions in the context of imbalance.

$$\text{MZE} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}(y_i \neq \hat{y}_i) = 1 - \text{CCR} \tag{16}$$

Each of these metrics offers unique insights into different aspects of model performance, ensuring a comprehensive evaluation of the new method's effectiveness in handling imbalance.

5.3 Experimental Setup

A comprehensive set of experiments was conducted across all datasets to evaluate the performance of the implemented methods for addressing class imbalance. Each dataset was run 30 times with different partitions, ensuring a robust evaluation and validation of the methods. This approach allows for thorough testing and a reliable assessment of the performance under varying conditions.

A comprehensive cross-validation procedure was employed to determine the optimal parameters for the model. For each resample, a systematic approach was used to explore various parameter configurations. The parameter optimisation was conducted using GridSearchCV, a method that examines multiple combinations of parameters to identify the best-performing set. The evaluation of these configurations was based on the MAE, a metric chosen for its ability to measure the accuracy of predictions.

The grid search involved testing a range of parameter values, including the regularisation parameter C and the maximum number of iterations. The regularisation parameter C controls the trade-off between fitting the training data and avoiding overfitting, while the maximum number of iterations sets a limit on the number of iterations for the optimisation algorithm. The cross-validation strategy used split the dataset into two folds. This ensured that each fold maintained the class proportions and applied data shuffling to introduce randomness. This setup provided a robust evaluation of parameter settings, leading to reliable and well-informed conclusions about the optimal model configuration.

5.4 Comparison to baseline methods.

In this section, we present the results of our experiments aimed at evaluating the impact of the implemented improvements in the `mord` package for handling class imbalance. Specifically, we assessed the performance of the newly implemented methods, LogATc and LogITc, designed to enhance the models to manage imbalanced datasets.

The performance of these enhanced models was tested across a variety of datasets, including both discretised continuous datasets and inherently ordinal datasets, following the methodology outlined in the previous section. The results from these experiments were analyzed to determine the effectiveness of the updated models in addressing class imbalance. Comparisons were made between the performance of the enhanced `mord` package and its previous version, as well as against earlier studies, to validate the improvements and highlight the advantages of the new model configurations.

Dataset	LogAT	LogATc
ERA	**1.222 ± 0.046**	1.449 ± 0.073
ESL	**0.308 ± 0.035**	0.509 ± 0.108
LEV	**0.415 ± 0.029**	0.689 ± 0.102
SWD	**0.449 ± 0.03**	0.705 ± 0.05
automobile	**0.592 ± 0.061**	0.653 ± 0.075
balance-scale	0.15 ± 0.048	**0.107 ± 0.021**
car	0.08 ± 0.012	0.08 ± 0.012
contact-lenses	0.478 ± 0.168	**0.422 ± 0.194**
eucalyptus	**1.003 ± 0.059**	1.022 ± 0.045
newthyroid	**0.032 ± 0.017**	0.036 ± 0.021
pasture	0.485 ± 0.252	**0.433 ± 0.176**
squash-stored	**0.449 ± 0.143**	0.521 ± 0.179
squash-unstored	**0.246 ± 0.108**	0.274 ± 0.119
tae	0.702 ± 0.179	**0.699 ± 0.172**
toy	**0.998 ± 0.095**	1.575 ± 0.119
winequality-red	**0.439 ± 0.019**	1.501 ± 0.059

Table 3. Comparison of MAE results for LogAT and LogATc.

Dataset	LogIT	LogITc
ERA	**2.787 ± 0.075**	2.815 ± 0.073
ESL	**0.314 ± 0.031**	0.648 ± 0.259
LEV	**0.424 ± 0.027**	1.012 ± 0.187
SWD	**0.528 ± 0.053**	1.034 ± 0.042
automobile	**0.693 ± 0.097**	0.744 ± 0.098
balance-scale	0.204 ± 0.027	**0.107 ± 0.021**
car	0.084 ± 0.011	**0.08** ± 0.013
contact-lenses	0.567 ± 0.194	**0.422 ± 0.243**
eucalyptus	**0.992 ± 0.041**	0.999 ± 0.043
newthyroid	**0.032 ± 0.017**	0.036 ± 0.021
pasture	0.422 ± 0.219	**0.441 ± 0.234**
squash-stored	**0.441 ± 0.151**	0.521 ± 0.169
squash-unstored	**0.244 ± 0.123**	0.262 ± 0.11
tae	0.674 ± 0.094	**0.671 ± 0.098**
toy	1.917 ± 0.153	**1.762 ± 0.139**
winequality-red	**0.438 ± 0.018**	1.874 ± 0.175

Table 4. Comparison of MAE results for LogIT and LogITc.

Dataset	LogATc	LogReg
balance-scale	0.107 ± 0.021	0.107 ± 0.021
car	0.08 ± 0.012	**0.073 ± 0.014**
contact-lenses	0.422 ± 0.194	**0.383 ± 0.215**
pasture	**0.433 ± 0.176**	0.437 ± 0.18
tae	0.699 ± 0.172	**0.61 ± 0.117**

Table 5. Comparison of MAE results for LogATc winners and Logistic Regressor

Dataset	LogITc	LogReg
balance-scale	0.107 ± 0.021	0.107 ± 0.021
car	0.08 ± 0.013	**0.073 ± 0.014**
contact-lenses	0.422 ± 0.243	**0.383 ± 0.215**
pasture	0.441 ± 0.234	**0.437 ± 0.18**
tae	0.671 ± 0.098	**0.61 ± 0.117**
toy	1.762 ± 0.139	**1.154 ± 0.081**

Table 6. Comparison of MAE results for LogITc winners and Logistic Regressor.

The comparison presented in Tables 3 and 4 does not clearly indicate consistent improvements by the enhanced versions, as LogATc and LogITc outperform their predecessors in only 4 and 6 out of 15 datasets, respectively. However, the results are still notable. Despite not achieving a universal performance boost, the enhanced models deliver comparable MAE values across most datasets, leading to a significant reduction in AMAE error. This is evident from the following table 7 8, which highlights the overall effectiveness of the proposed improvements.

When comparing the cost sensitive models with the traditional Logistic Regressor, LogATc and LogITc are not able to improve the MAE metric. However, as shown in Tables 5 and 6, the mean MAE value of the cost sensitive methods falls within the interval formed by the MAE mean ± standard deviation of the classic Logistic Regression model.

The following table provides a comparative summary of the AMAE results across different methods, illustrating how LogATc and LogITc fare against LogReg in terms of average mean absolute error.

Dataset	LogAT	LogATc
ERA	1.454 ± 0.064	**1.315 ± 0.102**
ESL	**0.428 ± 0.102**	0.504 ± 0.109
LEV	0.64 ± 0.053	**0.603 ± 0.068**
SWD	0.609 ± 0.028	**0.549 ± 0.055**
automobile	0.776 ± 0.115	**0.597 ± 0.106**
balance-scale	0.25 ± 0.149	**0.104 ± 0.027**
car	0.219 ± 0.041	**0.122 ± 0.024**
contact-lenses	0.581 ± 0.295	**0.37 ± 0.227**
eucalyptus	**1.055 ± 0.06**	1.09 ± 0.043
newthyroid	0.052 ± 0.034	**0.044 ± 0.032**
pasture	0.485 ± 0.252	**0.433 ± 0.176**
squash-stored	**0.488 ± 0.169**	0.523 ± 0.204
squash-unstored	**0.233 ± 0.148**	0.281 ± 0.152
tae	0.704 ± 0.181	**0.7 ± 0.173**
toy	**1.189 ± 0.13**	1.499 ± 0.164
winequality-red	**1.08 ± 0.045**	1.239 ± 0.144

Table 7. Comparison of AMAE results for LogAT and LogATc.

Dataset	LogIT	LogITc
ERA	**2.486 ± 0.157**	2.503 ± 0.172
ESL	**0.429 ± 0.105**	0.631 ± 0.167
LEV	**0.633 ± 0.061**	0.811 ± 0.121
SWD	**0.623 ± 0.044**	0.73 ± 0.05
automobile	0.901 ± 0.104	**0.615 ± 0.09**
balance-scale	0.421 ± 0.019	**0.104 ± 0.027**
car	0.261 ± 0.039	**0.118 ± 0.025**
contact-lenses	0.656 ± 0.315	**0.333 ± 0.202**
eucalyptus	**1.061 ± 0.038**	1.066 ± 0.045
newthyroid	0.052 ± 0.034	**0.044 ± 0.032**
pasture	**0.422 ± 0.219**	0.441 ± 0.234
squash-stored	**0.476 ± 0.184**	0.514 ± 0.202
squash-unstored	**0.259 ± 0.175**	0.272 ± 0.157
tae	0.676 ± 0.092	**0.673 ± 0.097**
toy	1.779 ± 0.205	**1.627 ± 0.214**
winequality-red	**1.072 ± 0.032**	1.178 ± 0.072

Table 8. Comparison of AMAE results for LogIT and LogITc.

Dataset	LogATc	LogReg
ERA	**1.315 ± 0.102**	1.512 ± 0.07
LEV	0.603 ± 0.068	**0.594 ± 0.07**
SWD	0.549 ± 0.055	**0.495 ± 0.049**
automobile	**0.597 ± 0.106**	0.734 ± 0.138
balance-scale	0.104 ± 0.027	0.104 ± 0.027
car	0.122 ± 0.024	**0.075 ± 0.036**
contact-lenses	0.37 ± 0.227	**0.306 ± 0.239**
eucalyptus	**1.09 ± 0.043**	1.142 ± 0.114
newthyroid	0.044 ± 0.032	**0.033 ± 0.038**
pasture	**0.433 ± 0.176**	0.437 ± 0.18
tae	0.7 ± 0.173	**0.61 ± 0.115**

Table 9. Comparison of AMAE results for LogATc winners and Logistic Regressor (LogReg)

Dataset	LogITc	LogReg
automobile	**0.615 ± 0.09**	0.734 ± 0.138
balance-scale	0.104 ± 0.027	0.104 ± 0.027
car	0.118 ± 0.025	**0.075 ± 0.036**
contact-lenses	0.333 ± 0.202	**0.306 ± 0.239**
newthyroid	0.044 ± 0.032	0.033 ± 0.038
tae	0.673 ± 0.097	**0.61 ± 0.115**
toy	1.627 ± 0.214	**1.051 ± 0.097**

Table 10. Comparison of AMAE results for LogITc winners and Logistic Regressor (LogReg).

The analysis of AMAE results in Tables 7, 8 shows that both LogATc and LogITc models outperform their earlier versions on several datasets. For instance, LogATc demonstrates improved performance over LogAT in terms of AMAE. Similarly, LogITc surpasses LogIT on datasets like automobile, balance-scale, and car. These results highlight the effectiveness of the enhancements made in the enhanced versions of both models.

In comparison to the traditional Logistic Regressor (LogReg), the enhanced models generally achieve superior results on most datasets. LogATc outperforms LogReg. However, there are instances where LogReg surpasses the enhanced models, as observed in datasets like car and contact-lenses. Similar to the case with the MAE metric, in many datasets where LogReg outperforms the enhanced models, the mean AMAE for the enhanced models still falls within the interval defined by the standard deviation and the mean AMAE of LogReg. These findings indicate that while the enhanced models perform well in many cases, LogReg remains competitive in certain datasets.

Dataset	LogAT	LogATc
ERA	**0.732 ± 0.025**	0.746 ± 0.025
ESL	**0.291 ± 0.031**	0.457 ± 0.075
LEV	**0.381 ± 0.027**	0.563 ± 0.065
SWD	**0.429 ± 0.028**	0.6 ± 0.033
automobile	**0.512 ± 0.049**	0.542 ± 0.06
balance-scale	0.116 ± 0.026	**0.094 ± 0.019**
car	0.079 ± 0.012	**0.077 ± 0.011**
contact-lenses	0.339 ± 0.111	**0.3 ± 0.127**
eucalyptus	0.679 ± 0.033	**0.678 ± 0.027**
newthyroid	**0.032 ± 0.017**	0.036 ± 0.021
pasture	0.43 ± 0.177	**0.411 ± 0.149**
squash-stored	**0.408 ± 0.126**	0.456 ± 0.154
squash-unstored	**0.246 ± 0.108**	0.274 ± 0.119
tae	0.539 ± 0.099	**0.539 ± 0.093**
toy	**0.705 ± 0.026**	0.797 ± 0.034
winequality-red	**0.406 ± 0.017**	0.84 ± 0.02

Table 11. Comparison of MZE results for LogAT and LogATc.

Dataset	LogIT	LogITc
ERA	0.895 ± 0.004	0.895 ± 0.004
ESL	**0.297 ± 0.028**	0.473 ± 0.102
LEV	**0.389 ± 0.025**	0.709 ± 0.084
SWD	**0.475 ± 0.036**	0.765 ± 0.014
automobile	**0.542 ± 0.051**	0.578 ± 0.058
balance-scale	0.143 ± 0.013	**0.094 ± 0.019**
car	0.081 ± 0.01	**0.077 ± 0.012**
contact-lenses	0.394 ± 0.127	**0.278 ± 0.154**
eucalyptus	**0.659 ± 0.021**	0.667 ± 0.026
newthyroid	**0.032 ± 0.017**	0.036 ± 0.021
pasture	**0.389 ± 0.154**	0.404 ± 0.179
squash-stored	**0.397 ± 0.125**	0.459 ± 0.157
squash-unstored	**0.241 ± 0.116**	0.262 ± 0.11
tae	0.503 ± 0.051	**0.497 ± 0.051**
toy	0.86 ± 0.032	**0.848 ± 0.028**
winequality-red	**0.402 ± 0.017**	0.916 ± 0.045

Table 12. Comparison of MZE results for LogIT and LogITc.

Dataset	LogATc	LogReg
balance-scale	0.094 ± 0.019	0.094 ± 0.019
car	0.077 ± 0.011	**0.067 ± 0.012**
contact-lenses	0.3 ± 0.127	**0.244 ± 0.129**
eucalyptus	0.678 ± 0.027	**0.669 ± 0.041**
pasture	**0.411 ± 0.149**	0.422 ± 0.161
tae	0.539 ± 0.093	**0.474 ± 0.079**

Table 13. Comparison of MZE results for LogATc winners and Logistic Regressor (LogReg)

Dataset	LogITc	LogReg
ERA	0.895 ± 0.004	**0.791 ± 0.02**
balance-scale	0.094 ± 0.019	0.094 ± 0.019
car	0.077 ± 0.012	**0.067 ± 0.012**
contact-lenses	0.278 ± 0.154	**0.244 ± 0.129**
tae	0.497 ± 0.051	**0.474 ± 0.079**
toy	0.848 ± 0.028	**0.757 ± 0.04**

Table 14. Comparison of MZE results for LogITc winners and Logistic Regressor (LogReg).

In the comparison of MZE in Tables 1112 results between the different models, we observe distinct performance trends. LogATc generally improves over the original LogAT in certain datasets like balance-scale, car, and contact-lenses, although there are cases where LogAT outperforms its enhanced version, such as in the ERA and toy datasets. Similarly, LogITc improves over LogIT in datasets like contact-lenses and toy, while LogIT performs better in others, such as SWD and ESL.

Comparing the enhanced models to the traditional Logistic Regressor (LogReg), the results are more mixed. LogATc holds an advantage in datasets like balance-scale, but LogReg outperforms LogATc in several datasets such as car, contact-lenses, and eucalyptus. For LogITc, LogReg surpasses it in key datasets such as ERA, car, and toy. In many of the datasets where LogReg performs better, the mean MZE for the enhanced models still falls within the interval defined by the standard deviation of the mean MZE for LogReg in some datasets.

	LogAT	LogATc	LogIT	LogITc
MAE	11	4	9	6
AMAE	6	9	8	7
MZE	9	6	10	5

Table 15. Comparison of the number of datasets where each method (LogAT, LogATc, LogIT, and LogITc) achieves the lowest values across the MAE, AMAE, and MZE metrics.(**Ordinal datasets**).

The Table 15 compares four methods (LogAT, LogATc, LogIT, and LogITc) based on the number of datasets in which each achieved the lowest error values across three metrics: MAE, AMAE, and MZE. LogAT performed best in 11 datasets for MAE, while LogATc excelled in 9 datasets for AMAE. Meanwhile, LogITc achieved the lowest MZE in 5 datasets. This comparison highlights the differing strengths of each method, with LogAT excelling in minimizing MAE, and LogATc and LogITc demonstrating strong performance in AMAE and MZE, respectively.

The performance analysis of these models across MAE, MZE, and AMAE underscores key differences, particularly in the context of class imbalance in the datasets. AMAE, adjusted to account for class imbalance, is critical for evaluating performance on datasets with a high imbalance ratio, where minority classes can disproportionately influence the overall error. While MAE and MZE provide a general view of error, AMAE offers a more balanced perspective in imbalanced scenarios, reflecting a fairer assessment. The results reveal that LogATc tends to perform better in managing class imbalance compared to LogITc. This can be attributed to the nature of the loss functions used by each model. LogITc employs the 0-1 loss function, which is more sensitive to classification errors, especially when misclassifications occur near class boundaries. This sensitivity can lead to higher error rates in scenarios where predictions are close to class thresholds. In contrast, LogATc uses an absolute error loss function, which may provide a more nuanced measure of error and is less affected by minor classification discrepancies. Therefore, while both enhanced models aim to address class imbalance, LogATc's approach appears to offer a more robust performance across various metrics, particularly in datasets with significant imbalance.

Let's now examine the results for the discretised datasets.

Dataset	LogAT	LogATc
abalone	**0.243 ± 0.009**	0.463 ± 0.018
auto	**0.246 ± 0.035**	0.321 ± 0.033
diabetes	**0.633 ± 0.13**	0.836 ± 0.163
housing	**0.3 ± 0.027**	0.358 ± 0.033
machine	**0.159 ± 0.038**	0.232 ± 0.05
pyrim	**0.497 ± 0.101**	0.53 ± 0.11
stock	0.318 ± 0.023	**0.309 ± 0.021**
triazines	**0.698 ± 0.059**	0.997 ± 0.141
wpbc	**1.134 ± 0.102**	1.222 ± 0.119

Table 16. Comparison of MAE results for LogAT and LogATc.

Dataset	LogIT	LogITc
abalone	**0.262 ± 0.007**	0.475 ± 0.021
auto	**0.251 ± 0.031**	0.328 ± 0.033
diabetes	**0.685 ± 0.146**	1.205 ± 0.224
housing	**0.313 ± 0.028**	0.356 ± 0.039
machine	**0.16 ± 0.038**	0.229 ± 0.049
pyrim	**0.517 ± 0.107**	0.604 ± 0.096
stock	0.312 ± 0.022	**0.303 ± 0.021**
triazines	**0.822 ± 0.14**	1.354 ± 0.167
wpbc	**1.421 ± 0.151**	1.522 ± 0.161

Table 17. Comparison of MAE results for LogIT and LogITc.

As shown in Table 1617, the MAE increases for the cost methods compared to their original versions. However, as demonstrated in the following tables, there are significant improvements in minimizing AMAE (see Table 1819), which accounts for class imbalance. In terms of MZE, while the precision of the enhanced methods decreases slightly, the drop is not drastic, as seen in Table 2021. This indicates that despite a trade-off in MAE, the enhanced methods offer advantages in handling imbalanced datasets and maintaining reasonable classification accuracy.

Dataset	LogAT	LogATc
abalone	0.805 ± 0.064	**0.434 ± 0.064**
auto	0.365 ± 0.054	**0.34 ± 0.061**
diabetes	0.915 ± 0.191	**0.573 ± 0.169**
housing	0.404 ± 0.06	**0.332 ± 0.048**
machine	0.43 ± 0.121	**0.332 ± 0.113**
pyrim	0.612 ± 0.14	**0.529 ± 0.21**
stock	0.292 ± 0.026	**0.268 ± 0.023**
triazines	1.381 ± 0.091	**1.09 ± 0.186**
wpbc	1.347 ± 0.118	**1.181 ± 0.139**

Table 18. Comparison of AMAE results for LogAT and LogATc.

Dataset	LogIT	LogITc
abalone	0.851 ± 0.071	**0.471 ± 0.054**
auto	0.368 ± 0.044	**0.343 ± 0.06**
diabetes	1.061 ± 0.219	**0.76 ± 0.129**
housing	0.466 ± 0.051	**0.357 ± 0.054**
machine	0.398 ± 0.135	**0.349 ± 0.11**
pyrim	0.615 ± 0.142	**0.569 ± 0.224**
stock	0.285 ± 0.025	**0.258 ± 0.022**
triazines	1.556 ± 0.282	**1.401 ± 0.277**
wpbc	1.654 ± 0.167	**1.472 ± 0.155**

Table 19. Comparison of AMAE results for LogIT and LogITc.

Dataset	LogAT	LogATc
abalone	**0.224 ± 0.008**	0.416 ± 0.015
auto	**0.242 ± 0.033**	0.312 ± 0.032
diabetes	**0.515 ± 0.107**	0.636 ± 0.092
housing	**0.275 ± 0.027**	0.335 ± 0.03
machine	**0.143 ± 0.031**	0.213 ± 0.042
pyrim	**0.445 ± 0.08**	0.459 ± 0.101
stock	0.31 ± 0.023	**0.302 ± 0.022**
triazines	**0.564 ± 0.047**	0.683 ± 0.058
wpbc	**0.669 ± 0.043**	0.675 ± 0.045

Table 20. Comparison of MZE results for LogAT and LogATc.

Dataset	LogIT	LogITc
abalone	**0.232 ± 0.006**	0.38 ± 0.013
auto	**0.248 ± 0.032**	0.316 ± 0.031
diabetes	**0.513 ± 0.082**	0.751 ± 0.096
housing	**0.284 ± 0.024**	0.323 ± 0.034
machine	**0.141 ± 0.029**	0.207 ± 0.041
pyrim	**0.459 ± 0.083**	0.513 ± 0.079
stock	0.305 ± 0.021	**0.295 ± 0.021**
triazines	**0.607 ± 0.075**	0.739 ± 0.048
wpbc	**0.66 ± 0.035**	0.685 ± 0.037

Table 21. Comparison of MZE results for LogIT and LogITc.

	LogAT	LogATc	LogIT	LogITc
MAE	8	1	8	1
AMAE	0	9	0	9
MZE	8	1	8	1

Table 22. Comparison of the number of datasets where each method (LogAT, LogATc, LogIT, and LogITc) achieves the lowest values across the MAE, AMAE, and MZE metrics.(**discretised datasets**).

The strength of the implemented methods lies in their performance on discretised datasets, particularly those with a high imbalance ratio. For these datasets, the models achieve notably strong AMAE values, indicating that they handle class imbalance effectively. While the MAE and MZE values are higher compared to their base versions, they remain within acceptable ranges. This suggests that a slight reduction in overall precision has been traded for less severe misclassifications, leading to more "accurate" classifications that better reflect the ordinal nature of the data.

5.5 Comparison to other ordinal classification methods.

In this section, we will analyse and compare the results obtained in the previous section with those reported in the main review of the state-of-the-art [14]. We used the same data partitions so that statistical results can be compared. For experimental details we refer to [14]. For comparative purpose with previous studies, we first evaluate performance in global metrics (MAE and MZE) and then in imbalance ordinal classification metric (AMAE).

Specifically, we will focus on comparing our implemented methods with the two linear models in the survey, the Proportional Odds Model (POM) [11] and the Support Vector Ordinal Regression with Implicit Constraints (SVORIMLin, shorted to SLin in tables) [4].

The Proportional Odds Model (POM) is a commonly used method for ordinal classification, which assumes that the relationship between each pair of outcome categories is the same. This model fits a series of parallel regression lines for the cumulative logits of the ordered categories, making it a straightforward yet powerful approach for ordinal data. The POM can be viewed as a member of the wider family of cumulative link models that extends binary logistic regression to ordinal classification.

The Support Vector Ordinal Regression with Implicit Constraints (SVORIMLin) is an adaptation of support vector machines for ordinal classification. It introduces implicit constraints to ensure that the predicted values respect the natural order of the classes. SVORIMLin typically offers more flexibility compared to traditional linear models, potentially leading to better performance in certain cases. Table 23 presents comparative results showing the MAE and MZE for all the methods. Note these metrics evaluate global performance but do not account for unbalanced label distribution.

The updated versions of LogAT and LogIT (i.e., LogATc and LogITc) do not consistently outperform their original counterparts in global metrics. In several cases, the original versions (LogAT and LogIT) demonstrate lower error rates, as seen in datasets like LEV, SWD, and pasture, where LogAT surpasses LogATc. This pattern suggests that the modifications introduced in the newer versions may not provide universal improvements and could be dataset-dependent.

Additionally, the higher error rates observed in LogATc and LogITc on certain datasets, such as winequality-red, raise concerns about potential overfitting or insufficient handling of imbalance. These issues indicate that the newer versions may struggle with more complex or imbalanced datasets, reducing their overall effectiveness.

Conversely, the POM and SVORIMLin methods consistently demonstrate strong performance across specific datasets. For example, SVORIMLin achieves the best results on eucalyptus and toy, while POM outperforms other models on ERA, LEV, balance-scale, and winequality-red. This indicates that these models are particularly well-suited to managing certain types of data, especially those with inherent complexities or imbalances.

The comparison of Mean Zero Error values across various datasets, as presented in Table 24, provides several insights into the performance of the evaluated models. These conclusions are also linked to the Mean Absolute Error results discussed earlier.

Finally, Table 25 highlights the results in terms of AMAE. This table reaffirms that the newly implemented methods achieve better performance in AMAE compared to their predecessors, demonstrating that they can effectively compete with more complex methods such as POM and SvorimLin. Despite their simplicity, these new approaches manage to reduce error significantly, offering a competitive alternative for addressing class imbalance, particularly in challenging datasets.

The analysis indicates that the POM and SVORIMLin methods generally exhibit strong performance across specific datasets. These findings underscore that POM and SVORIMLin are particularly adept at handling certain types of data, as reflected in both MAE and MZE metrics.

In summary, while some models demonstrate overall robustness across various datasets, the effectiveness of updated versions of certain models, such as LogATc and LogITc, does not uniformly surpass their predecessors. Additionally, the performance of models in handling imbalance is critical, with certain methods like SVORIMLin proving to be particularly effective in such scenarios. The findings underscore the necessity for careful model selection tailored to the specific dataset characteristics.

Dataset	LogAT	LogATc	LogIT	LogITc	POM	SLin
ERA	1.222 ± 0.046	1.449 ± 0.073	2.787 ± 0.075	2.815 ± 0.073	**1.218 ± 0.05**	*1.222 ± 0.036*
ESL	**0.308 ± 0.035**	0.509 ± 0.108	0.314 ± 0.031	0.648 ± 0.259	*0.31 ± 0.038*	**0.308 ± 0.035**
LEV	0.415 ± 0.029	0.689 ± 0.102	0.424 ± 0.027	1.012 ± 0.187	**0.409 ± 0.03**	*0.414 ± 0.032*
SWD	**0.449 ± 0.03**	0.705 ± 0.05	0.528 ± 0.053	1.034 ± 0.042	*0.45 ± 0.03*	0.451 ± 0.03
automobile	*0.592 ± 0.061*	0.653 ± 0.075	0.693 ± 0.097	0.744 ± 0.098	0.953 ± 0.687	**0.474 ± 0.09**
balance-scale	*0.15 ± 0.048*	**0.107 ± 0.021**	0.204 ± 0.027	**0.107 ± 0.021**	**0.107 ± 0.021**	**0.107 ± 0.021**
car	*0.08 ± 0.012*	*0.08 ± 0.012*	0.084 ± 0.011	*0.08 ± 0.013*	1.451 ± 0.548	**0.078 ± 0.01**
contact-lenses	0.478 ± 0.168	**0.422 ± 0.194**	0.567 ± 0.194	**0.422 ± 0.243**	0.533 ± 0.253	*0.461 ± 0.156*
eucalyptus	1.003 ± 0.059	1.022 ± 0.045	*0.992 ± 0.041*	0.999 ± 0.043	1.939 ± 0.254	**0.382 ± 0.028**
newthyroid	0.032 ± 0.017	0.036 ± 0.021	0.032 ± 0.017	0.036 ± 0.021	**0.028 ± 0.022**	*0.031 ± 0.024*
pasture	0.485 ± 0.252	0.433 ± 0.176	*0.422 ± 0.219*	0.441 ± 0.234	0.585 ± 0.204	**0.337 ± 0.115**
squash-stored	0.449 ± 0.143	0.521 ± 0.179	*0.441 ± 0.151*	0.521 ± 0.169	0.813 ± 0.248	**0.377 ± 0.138**
squash-unstored	*0.246 ± 0.108*	0.274 ± 0.119	**0.244 ± 0.123**	0.262 ± 0.11	0.826 ± 0.23	0.318 ± 0.112
tae	0.702 ± 0.179	0.699 ± 0.172	0.674 ± 0.094	0.671 ± 0.098	*0.628 ± 0.116*	**0.548 ± 0.079**
toy	0.998 ± 0.095	1.575 ± 0.119	1.917 ± 0.153	1.762 ± 0.139	*0.981 ± 0.039*	**0.96 ± 0.066**
winequality-red	0.439 ± 0.019	1.501 ± 0.059	*0.438 ± 0.018*	1.874 ± 0.175	**0.435 ± 0.017**	0.439 ± 0.018

Table 23. Comparison of Mean Absolute Error (MAE) across various classification methods for different datasets. The best result for each dataset is in bold face. Notable observations include consistent performance by POM and SVORIMLin on certain datasets, while updated versions of LogAT and LogIT do not always outperform their predecessors.

Dataset	LogAT	LogATc	LogIT	LogITc	POM	SLin
ERA	**0.732 ± 0.025**	0.746 ± 0.025	0.895 ± 0.004	0.895 ± 0.004	*0.744 ± 0.021*	0.75 ± 0.023
ESL	**0.291 ± 0.031**	0.457 ± 0.075	0.297 ± 0.028	0.473 ± 0.102	0.295 ± 0.034	*0.293 ± 0.033*
LEV	*0.381 ± 0.027*	0.563 ± 0.065	0.389 ± 0.025	0.709 ± 0.084	**0.377 ± 0.028**	0.384 ± 0.028
SWD	**0.429 ± 0.028**	0.6 ± 0.033	0.475 ± 0.036	0.765 ± 0.014	0.432 ± 0.03	*0.431 ± 0.03*
automobile	*0.512 ± 0.049*	0.542 ± 0.06	0.542 ± 0.051	0.578 ± 0.058	0.533 ± 0.194	**0.408 ± 0.068**
balance-scale	*0.116 ± 0.026*	**0.094 ± 0.019**	**0.143 ± 0.013**	**0.094 ± 0.019**	**0.094 ± 0.019**	**0.094 ± 0.019**
car	*0.079 ± 0.012*	**0.077 ± 0.011**	0.081 ± 0.01	**0.077 ± 0.012**	0.843 ± 0.306	**0.077 ± 0.01**
contact-lenses	0.339 ± 0.111	*0.3 ± 0.127*	0.394 ± 0.127	**0.278 ± 0.154**	0.383 ± 0.17	0.361 ± 0.099
eucalyptus	0.679 ± 0.033	0.678 ± 0.027	*0.659 ± 0.021*	0.667 ± 0.026	0.851 ± 0.016	**0.357 ± 0.023**
newthyroid	0.032 ± 0.017	0.036 ± 0.021	0.032 ± 0.017	0.036 ± 0.021	**0.028 ± 0.022**	*0.031 ± 0.024*
pasture	0.43 ± 0.177	0.411 ± 0.149	*0.389 ± 0.154*	0.404 ± 0.179	0.504 ± 0.154	**0.344 ± 0.122**
squash-stored	0.408 ± 0.126	0.456 ± 0.154	*0.397 ± 0.125*	0.459 ± 0.157	0.618 ± 0.152	**0.372 ± 0.133**
squash-unstored	*0.246 ± 0.108*	0.274 ± 0.119	**0.241 ± 0.116**	0.262 ± 0.11	0.651 ± 0.142	0.315 ± 0.113
tae	0.539 ± 0.099	0.539 ± 0.093	0.503 ± 0.051	0.497 ± 0.051	*0.496 ± 0.077*	**0.46 ± 0.072**
toy	**0.705 ± 0.026**	0.797 ± 0.034	0.86 ± 0.032	0.848 ± 0.028	*0.711 ± 0.026*	0.73 ± 0.016
winequality-red	0.406 ± 0.017	0.84 ± 0.02	**0.402 ± 0.017**	0.916 ± 0.045	**0.403 ± 0.015**	0.407 ± 0.015

Table 24. Comparison of MZE values across various datasets for different ordinal classification models. The best result for each dataset is in bold face. The table highlights the variability in model performance across different datasets, emphasising the importance of dataset-specific model selection.

Dataset	LogAT	LogATc	LogIT	LogITc	POM	SLin
ERA	1.454 ± 0.064	**1.315 ± 0.102**	2.486 ± 0.157	2.503 ± 0.172	*1.384 ± 0.085*	1.455 ± 0.09
ESL	0.428 ± 0.102	0.504 ± 0.109	**0.429 ± 0.105**	0.631 ± 0.167	0.442 ± 0.095	0.477 ± 0.103
LEV	0.64 ± 0.053	**0.603 ± 0.068**	0.633 ± 0.061	0.811 ± 0.121	*0.627 ± 0.054*	0.632 ± 0.059
SWD	0.609 ± 0.028	**0.549 ± 0.055**	0.623 ± 0.044	0.73 ± 0.05	0.622 ± 0.031	*0.602 ± 0.041*
automobile	0.776 ± 0.115	*0.597 ± 0.106*	0.901 ± 0.104	0.615 ± 0.09	1.026 ± 0.8	**0.582 ± 0.098**
balance-scale	0.25 ± 0.149	**0.104 ± 0.027**	0.421 ± 0.019	**0.104 ± 0.027**	**0.104 ± 0.027**	**0.104 ± 0.027**
car	0.219 ± 0.041	*0.122 ± 0.024*	0.261 ± 0.039	**0.118 ± 0.025**	0.896 ± 0.271	0.187 ± 0.045
contact-lenses	0.581 ± 0.295	*0.37 ± 0.227*	0.656 ± 0.315	**0.333 ± 0.202**	0.51 ± 0.286	0.5 ± 0.265
eucalyptus	*1.055 ± 0.06*	1.09 ± 0.043	1.061 ± 0.038	1.066 ± 0.045	1.89 ± 0.238	**0.409 ± 0.035**
newthyroid	0.052 ± 0.034	**0.044 ± 0.032**	0.052 ± 0.034	**0.044 ± 0.032**	*0.05 ± 0.04*	0.055 ± 0.05
pasture	0.485 ± 0.252	0.433 ± 0.176	*0.422 ± 0.219*	0.441 ± 0.234	0.585 ± 0.204	**0.337 ± 0.115**
squash-stored	0.488 ± 0.169	0.523 ± 0.204	*0.476 ± 0.184*	0.514 ± 0.202	0.815 ± 0.251	**0.413 ± 0.157**
squash-unstored	**0.233 ± 0.148**	0.281 ± 0.152	*0.259 ± 0.175*	0.272 ± 0.157	0.791 ± 0.332	0.406 ± 0.163
tae	0.704 ± 0.181	0.7 ± 0.173	0.676 ± 0.092	0.673 ± 0.097	*0.629 ± 0.118*	**0.548 ± 0.079**
toy	1.189 ± 0.13	1.499 ± 0.164	1.779 ± 0.205	1.627 ± 0.214	1.213 ± 0.059	**1.203 ± 0.04**
winequality-red	*1.08 ± 0.045*	1.239 ± 0.144	**1.072 ± 0.032**	1.178 ± 0.072	1.083 ± 0.04	1.089 ± 0.041

Table 25. Comparison of AMAE values across various datasets for different ordinal classification models, including the original and updated versions of LogAT and LogIT, alongside POM and SVORIMLin. The table highlights the variability in model performance across different datasets, emphasising the importance of dataset-specific model selection

6 Conclusion.

In this study, we addressed a critical challenge in ordinal classification: handling imbalanced datasets where certain classes are significantly underrepresented. Imbalance in ordinal classification can severely distort model performance, as traditional methods often fail to adequately address the skewed distribution of classes.

To tackle this problem, we proposed an enhancement of two ordinal logistic regression models by incorporating class-weight adjustments to improve performance in minority classes and thus mitigate the adverse effects of imbalanced class distributions.

While LogATc and LogITc demonstrated improvements over their predecessors in terms of handling imbalanced data, the results indicate that these updated methods still fall short compared to established techniques such as POM and SVORIMLin. Specifically, although LogATc and LogITc showed promise in reducing average mean absolute error (AMAE), their performance in terms of mean zero error (MZE) did not match the robustness of POM and SVORIMLin.

It is noteworthy that the performance on discretised datasets significantly surpasses that on real ordinal datasets. One possible reason is that discretised datasets guarantee an ordinal relation of labels whilst the others datasets have ordinal labels but there is not guarantee that the order and distances between labels presents actual ordinal conditions. This can make single-model linear fitting difficult because it assumes this ordered arrangement in the output space. Future work should focus on this aspect by conducting a comprehensive review comparing the performance of these methods across discretised datasets.

The comparative analysis demonstrates that our methods introduce valuable improvements in key areas, although there is still room for further enhancement to match the top-performing approaches in the field. This emphasizes the complexity of addressing imbalanced datasets and presents exciting opportunities for future research to refine and optimize these solutions.Future work might include how to simultaneously improve AMAE while reducing the impact in global performance metrics. Indeed, this tradeoff has been an active research field in binary and multiclass tasks for years.

References

1. Alicioglu, G., Sun, B., Ho, S.S.: Assessing Accident Risk using Ordinal Regression and Multinomial Logistic Regression Data Generation. In: 2020 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (Jul 2020). https://doi.org/10.1109/IJCNN48605.2020.9207105, https://ieeexplore.ieee.org/abstract/document/9207105?casa_token=7k1mGLAtUDYAAAAA:VdTlBuDjeqOeOWaIAvc8adHLGeFoiQRH1E5gaOWM8abIofYf5hz1tEihCUvteuLjKNueEo7F_b0, iSSN: 2161-4407
2. Carbonell, J.G., Michalski, R.S., Mitchell, T.M.: 1 - AN OVERVIEW OF MACHINE LEARNING. In: Michalski, R.S., Carbonell, J.G., Mitchell, T.M. (eds.) Machine Learning, pp. 3–23. Morgan Kaufmann, San Francisco (CA) (Jan 1983). https://doi.org/10.1016/B978-0-08-051054-5.50005-4, https://www.sciencedirect.com/science/article/pii/B9780080510545500054
3. Cheng, J., Wang, Z., Pollastri, G.: A neural network approach to ordinal regression. In: 2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence). pp. 1279–1284. IEEE (2008)
4. Chu, W., Keerthi, S.S.: Support vector ordinal regression. Neural computation **19**(3), 792–815 (2007)
5. Cruz-Ramírez, M., Hervás-Martínez, C., Sánchez-Monedero, J., Gutiérrez, P.: Metrics to guide a multi-objective evolutionary algorithm for ordinal classification. Neurocomputing **135**, 21–31 (Jul 2014). https://doi.org/10.1016/j.neucom.2013.05.058, https://linkinghub.elsevier.com/retrieve/pii/S0925231213011399
6. Deng, W.Y., Zheng, Q.H., Lian, S., Chen, L., Wang, X.: Ordinal extreme learning machine. Neurocomputing **74**(1-3), 447–456 (2010)
7. Dudjak, M., Martinović, G.: An empirical study of data intrinsic characteristics that make learning from imbalanced data difficult. Expert Systems with Applications **182**, 115297 (2021). https://doi.org/https://doi.org/10.1016/j.eswa.2021.115297, https://www.sciencedirect.com/science/article/pii/S0957417421007272
8. Elreedy, D., Atiya, A.F.: A comprehensive analysis of synthetic minority oversampling technique (smote) for handling class imbalance. Information Sciences **505**, 32–64 (2019). https://doi.org/https://doi.org/10.1016/j.ins.2019.07.070

9. Fan, R.E., Chang, K.W., Hsieh, C.J., Wang, X.R., Lin, C.J.: LIBLINEAR: A library for large linear classification. Journal of Machine Learning Research **9**(61), 1871–1874 (2008), http://jmlr.org/papers/v9/fan08a.html
10. Fernández-Caballero, J.C., Martínez-Estudillo, F.J., Hervás-Martínez, C., Gutiérrez, P.A.: Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks. IEEE Transactions on Neural Networks **21**(5), 750 –770 (may 2010)
11. Fullerton, A.S., Xu, J.: The proportional odds with partial proportionality constraints model for ordinal response variables. Social science research **41**(1), 182–198 (2012)
12. García, V., Sánchez, J.S., Mollineda, R.A.: On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge-Based Systems **25**(1), 13–21 (Feb 2012). https://doi.org/10.1016/j.knosys.2011.06.013, https://www.sciencedirect.com/science/article/pii/S0950705111001286
13. Gentry, A.E., Jackson-Cook, C.K., Lyon, D.E., Archer, K.J.: Penalized Ordinal Regression Methods for Predicting Stage of Cancer in High-Dimensional Covariate Spaces. Cancer Informatics **14s2**, CIN.S17277 (Jan 2015). https://doi.org/10.4137/CIN.S17277, https://doi.org/10.4137/CIN.S17277, publisher: SAGE Publications Ltd STM
14. Gutiérrez, P.A., Perez-Ortiz, M., Sanchez-Monedero, J., Fernandez-Navarro, F., Hervás-Martínez, C.: [DATASETS]-Ordinal regression methods: Survey and Experimental Study, https://www.uco.es/grupos/ayrna/orreview
15. Gutierrez, 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 (Jan 2016). https://doi.org/10.1109/TKDE.2015.2457911, http://ieeexplore.ieee.org/document/7161338/
16. He, H., Garcia, E.A.: Learning from Imbalanced Data. IEEE Transactions on Knowledge and Data Engineering **21**(9), 1263–1284 (Sep 2009). https://doi.org/10.1109/TKDE.2008.239, https://ieeexplore.ieee.org/document/5128907, conference Name: IEEE Transactions on Knowledge and Data Engineering
17. Hosmer, Jr, D.W., Lemeshow, S., Sturdivant, R.X.: Applied Logistic Regression. John Wiley & Sons (4 2013)
18. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. IEEE transactions on Neural Networks **13**(2), 415–425 (2002)
19. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 27 (Mar 2019). https://doi.org/10.1186/s40537-019-0192-5, https://doi.org/10.1186/s40537-019-0192-5
20. King, G., Zeng, L.: Logistic Regression in Rare Events Data. Political Analysis **9**(2), 137–163 (Jan 2001). https://doi.org/10.1093/oxfordjournals.pan.a004868, https://www.cambridge.org/core/journals/political-analysis/article/logistic-regression-in-rare-events-data/1E09F0F36F89DF12A823130FDF0DA462
21. McCullagh, P.: Regression models for ordinal data. Journal of the Royal Statistical Society: Series B (Methodological) **42**(2), 109–127 (1980)
22. Mohammed, R., Rawashdeh, J., Abdullah, M.: Machine Learning with Oversampling and Undersampling Techniques: Overview Study and Experimental Results. In: 2020 11th International Conference on Information and Communication Systems (ICICS). pp. 243–248 (Apr 2020). https://doi.org/10.1109/ICICS49469.2020.239556, https://ieeexplore.ieee.org/abstract/document/9078901, iSSN: 2573-3346
23. Pedregosa, F., Bach, F., Gramfort, A.: On the Consistency of Ordinal Regression Methods. Journal of Machine Learning Research **18**(55), 1–35 (2017)
24. Rennie, J.D.: Ordinal logistic regression. MIT (2005)
25. Rennie, J.D., Srebro, N.: Loss functions for preference levels: Regression with discrete ordered labels. In: Proceedings of the IJCAI multidisciplinary workshop on advances in preference handling. vol. 1. AAAI Press, Menlo Park, CA (2005)
26. Thai-Nghe, N., Gantner, Z., Schmidt-Thieme, L.: Cost-sensitive learning methods for imbalanced data. In: The 2010 International Joint Conference on Neural Networks (IJCNN). pp. 1–8 (Jul 2010). https://doi.org/10.1109/IJCNN.2010.5596486, https://ieeexplore.ieee.org/abstract/document/5596486, iSSN: 2161-4407
27. Zhou, Z.H., Liu, X.Y.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. IEEE Transactions on Knowledge and Data Engineering **18**(1), 63–77 (Jan 2006). https://doi.org/10.1109/TKDE.2006.17, https://ieeexplore.ieee.org/document/1549828, conference Name: IEEE Transactions on Knowledge and Data Engineering