DOI: https://doi.org/10.4038/sljas.v27i1.8255

Sri Lankan Journal of Applied Statistics, Vol (27-1)

Imbalance-Aware Bayesian Multinomial Logistic Regression Via Adaptive Entropy-Based Observation Weighting

F. M. Taiwo[1,*], C. Akcora[2], and S. Muthukumarana[3]

[1]Department of Statistics, University of Manitoba, Manitoba, Canada

[2]AI Institute, University of Central Florida, Florida, USA

[3]Department of Statistics, University of Manitoba, Manitoba, Canada

*Corresponding author: taiwom1@myumanitoba.ca

Received: 12th January, 2026/ Revised: 17th March, 2026/ Accepted: 23rd March, 2026/ Published: 31st March, 2026

ABSTRACT

Class imbalance poses a fundamental challenge in multi-class classification, where standard maximum-likelihood methods systematically favor the majority classes. We develop an Imbalance-Aware Bayesian Multinomial Logistic Regression (IA-BMLR) model that adapts a weighted loss framework, originally developed for variational autoencoders, to Bayesian discriminative classification. The weighting approach combines inverse-frequency class weights with entropy-based weights derived from a Local Entropy Score (LES) that emphasizes observations in ambiguous, high-entropy regions. We treat the entropy weight parameter γ as a learnable variable with a half-normal prior, rather than a fixed parameter, thereby enabling data-driven calibration via Bayesian inference. We evaluate IA-BMLR on nine benchmark datasets that span severe to mild class imbalance. On the Yeast (ten classes, imbalance ratio = 92.6) and the Lymphography (four classes, imbalance ratio = 40.5) datasets, IA-BMLR achieves a G-Mean of 0.302 and 0.456, respectively. In contrast, all baseline methods, including XGBoost, Random Forest, SVC, Multinomial logistic regression, and unweighted Bayesian Multinomial logistic regression, achieve an exact zero, indicating prediction failure for at least one minority class. IA-BMLR achieves the highest G-Mean across three of nine datasets, demonstrating its reliability in minority-class prediction while maintaining competitive performance on standard metrics.

Keywords: Bayesian inference, class imbalance, multinomial logistic regression, weighted likelihood, entropy weighting, multi-class classification

1 Introduction

Classification is a fundamental task in machine learning (ML), underpinning applications across domains such as fraud detection (Di Martino et al., 2012), medical diagnosis (Rawashdeh et al., 2020), fault prediction (Shatnawi, 2012), text classification (Liu et al., 2024), educational prediction (Al-Ashoor and Abdullah, 2022), and oil-spill detection in satellite images (Kubat et al., 1998). A pervasive challenge in many of these domains is class imbalance, where one or more classes are significantly underrepresented in the data (Stando et al., 2024; Bai et al., 2024). This imbalance often leads to poor model performance on minority classes, which are frequently the classes of greatest interest (e.g., medical diagnosis of rare diseases, fraud detection, fault classification in manufacturing systems, and species identification in ecological studies). Despite substantial research efforts, imbalanced classification remains a difficult problem, particularly in multi-class settings (Guo et al., 2024).

To mitigate class imbalance, researchers have proposed solutions at both the data and algorithm levels (Tanha et al., 2020). Data-level methods typically involve resampling techniques, such as random over- and under-sampling, and synthetic approaches, such as SMOTE (Chawla et al., 2002), to rebalance the class distribution. Some strategies selectively duplicate minority samples based on heuristics, rather than creating synthetic examples (Kotsiantis et al., 2006; Ganganwar, 2012). Algorithm-level approaches modify the learning process by incorporating mechanisms such as cost-sensitive learning, where different misclassification costs are assigned to each class, or using specialized architectures like ensemble and deep learning methods (Tanha et al., 2020). Hybrid methods combine data- and algorithm-level methods. While effective in binary settings, many of these methods do not scale naturally to multi-class classification, requiring manual specification of sampling ratios or cost parameters, and do not provide the principled uncertainty quantification offered by Bayesian models.

Alongside these approaches, Bayesian modeling offers an underutilized yet principled alternative for imbalanced multi-class classification. Bayesian statistics provides a probabilistic framework for updating prior beliefs with evidence from observed data (van de Schoot et al., 2021), yielding posterior distributions that quantify both parameter estimates and predictive uncertainty. This is particularly valuable in settings where uncertainty estimation is critical, such as education, healthcare, and finance. More recently, Lazic (2026) proposed a Bayesian weighted-likelihood framework that accounts for class imbalance by reweighting likelihood contributions according to class proportions, primarily demonstrated for binary and ordered outcomes. However, Bayesian methods, particularly for multi-class outcomes, remain relatively underexplored in the literature. We investigate Bayesian Multinomial Logistic Regression (BMLR), which generalizes logistic regression to categorical outcomes with more than two classes (Kalina, 2022).

In this paper, we develop an Imbalance-Aware Bayesian Multinomial Logistic Regression model (IA-BMLR) that accounts for class imbalance by modifying the loss function, specifically, the log-likelihood within a Bayesian framework. This formulation aligns with cost-sensitive learning strategies commonly used in frequentist machine learning, yet it integrates smoothly into Bayesian workflows. Unlike existing Bayesian imbalance-handling approaches that rely on fixed class-level adjustments or post hoc probability corrections, IA-BMLR introduces observation-level class and entropy weighting within a fully Bayesian multi-class discriminative model, enabling parameter inference and predictive uncertainty to reflect imbalance-driven cost asymmetries. Our developed framework is informed by the weighted-likelihood approach of Lazic (2026) and uncertainty-aware weighting strategies in deep generative models (Zare et al., 2025), which we adapt to the multi-class Bayesian classification setting. We evaluate IA-BMLR's effectiveness on real-world imbalanced datasets, benchmarking it against established models commonly used for such classification tasks, including Support Vector Classifier (SVC), Multinomial Logistic Regression (MLR), Random Forest (RF), and XGBoost (XGB). Our contributions are as follows:

1. **Bayesian adaptation:** We adapt the weighted loss framework of Zare et al. (2025) to Bayesian multinomial logistic regression, by incorporating observation-specific weights directly into the likelihood, enabling posterior uncertainty quantification for imbalanced classification.

2. **Learnable entropy parameter:** We treat the entropy weight parameter γ as a random variable rather than a fixed parameter, enabling data-driven calibration.

3. **Empirical validation:** We demonstrate on nine benchmark datasets that IA-BMLR achieves competitive performance on the evaluation metrics used and a non-zero geometric mean score on a severely imbalanced dataset where all baseline methods achieve zero geometric mean on minority classes.

The remainder of this paper is organized as follows. Section 2 presents the IA-BMLR methodology, implementation, and experimental setup. Section 3 presents the results and discussion, and Section 4 concludes with a summary, limitations, and directions for future research.

2 Methodology

In this section, we develop a comprehensive framework for Bayesian classification under class imbalance that combines inverse-frequency class weighting with entropy-based observation weighting within a weighted-likelihood formulation. We use multinomial logistic regression as the base model in this

study. First, we discuss the unweighted variant of our model, which is the standard Bayesian Multinomial Logistic Regression model. Second, we introduce the Local Entropy Score (LES), which quantifies classification difficulty at the observation level using Shannon entropy of local neighborhood class distributions. Third, we develop the observation weighting scheme that combines class weights with entropy weights. Finally, we present the complete Imbalance-Aware Bayesian Multinomial Logistic Regression (IA-BMLR) model, which incorporates a learnable entropy-weighting parameter.

Given a dataset of N observations, let $\boldsymbol{X} \in \mathbb{R}^{N \times p}$ be the matrix of p explanatory variables (features), where the i-th row $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip})$ corresponds to observation i. Let $\boldsymbol{Y} = (y_1, y_2, \ldots, y_N)$ denote the vector of categorical responses, where each $y_i \in \{1, 2, \ldots, K\}$ indicates the class label for observation i among K possible classes. Let n_k be the number of samples in class k, with $N = \sum_{k=1}^{K} n_k$. We describe the dataset as imbalanced when the class frequencies vary substantially, characterized by the imbalance ratio, IR = $\max_k n_k / \min_k n_k \gg 1$.

2.1 Unweighted Bayesian Multinomial Logistic Regression (U-BMLR)

We assume that y_i follows a categorical distribution denoted by Cat(K; $\boldsymbol{\theta}$) with $K > 0$ categories and event probabilities $\boldsymbol{\theta} = (\theta_1, \theta_2, \ldots, \theta_K)$. The probability mass function for the categorical-distributed response variable is

$$p(y_i|\boldsymbol{\theta}) = \prod_{k=1}^{K} \theta_{ik}^{[y_i=k]}$$

where $[y_i = k]$ evaluates to 1 if $y_i = k$ and 0 otherwise. The parameters θ_{ik} are constrained to satisfy $\theta_{ik} \geq 0$ and $\sum_k \theta_{ik} = 1$, because they represent probabilities. Let x_{ij} be the element in the i-th row and j-th column associated with y_i for $j = 1, 2, \ldots, p$, that can be included in the model through the probability θ_{ik} for each category k. The linear combination of the predictors is represented as

$$\eta_{ik} = \beta_{k0} + \mathbf{x}_i^\top \boldsymbol{\beta}_k,$$

where β_{k0} is the intercept and $\boldsymbol{\beta}_k \in \mathbb{R}^p$ is the coefficient vector for class k. Given that $\boldsymbol{\theta}$ must sum to 1, and each θ_{ik} is in the interval [0, 1], we obtain $\theta_{ik} = P(y_i = k \mid \boldsymbol{x}_i)$ using the softmax function as a link function given by:

$$\theta_{ik} = \frac{\exp(\eta_{ik})}{\sum_{k=1}^{K} \exp(\eta_{ik})} \tag{1}$$

The **likelihood function** for N observations is then defined as:

$$p(\boldsymbol{Y}|\boldsymbol{X},\boldsymbol{\beta}) = \prod_{i=1}^{N}\prod_{k=1}^{K}\theta_{ik}^{[y_i=k]} = \prod_{i=1}^{N}\prod_{k=1}^{K}\left(\frac{\exp(\eta_{ik})}{\sum_{k=1}^{K}\exp(\eta_{ik})}\right)^{[y_i=k]} \tag{2}$$

In this regression model, it is essential to impose certain restrictions for parameter identifiability. Thus, we assume that for a chosen reference category r (usually the majority class), set $\beta_{r0} = 0$ and $\beta_{rj} = 0$, for all j. The **log-likelihood function** is then given as

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{N}\sum_{k=1}^{K}[y_i = k]\log\theta_{ik} = \sum_{i=1}^{N}\log P(y_i|\mathbf{x}_i,\boldsymbol{\beta}).$$

Under class imbalance, the log-likelihood is dominated by the majority classes, since they contribute more terms to the objective.

2.2 Weights Formulation In this Subsection, we will discuss the formulation of the observation weights.

2.2.1 Local Entropy Score (LES)

To quantify classification difficulty at the observation level, we introduce LES. For an observation i with feature vector $\mathbf{x}_i$, let $\mathcal{N}_\kappa(\mathbf{x}_i)$ denote the set of κ nearest neighbors of $\mathbf{x}_i$ in feature space. The local class distribution is

$$\hat{p}_{ik} = \frac{1}{\kappa}\sum_{j\in\mathcal{N}_\kappa(\mathbf{x}_i)}\mathbb{I}(y_j = k), \quad k = 1,\ldots,K.$$

The LES for observation i is the normalized Shannon entropy given as:

$$H_i = -\sum_{k=1}^{K}\hat{p}_{ik}\log(\hat{p}_{ik}+\epsilon)/\log K$$

where $\epsilon > 0$ is a small constant for numerical stability (we use $\epsilon = 10^{-10}$). When $H_i = 0$, it means that all neighbors belong to the same class, and when $H_i = \log K$, it means that the neighbors are distributed across all K classes. H_i maps into $[0,1]$ by normalizing by $\log K$. We note that observations near

decision boundaries have higher LES values than the observations in class-homogeneous regions.

2.2.2 Class Weights

To address class imbalance, we use inverse-frequency weights. For observation i belonging to class $k = y_i$, the class weight is defined as

$$w_{class}(i) = \frac{N}{K \times n_{y_i}}.$$

$w_{class}(i)$ is large for small n_k, which are the minority classes, and small for the majority classes. This ensures a balanced contribution of the classes to the likelihood.

2.2.3 Entropy Weights

We emphasize observations near decision boundaries by defining an entropy term $(1 + H_i)^\gamma$ where $i = 1, \ldots, N$, and $\gamma \geq 0$ is a learnable parameter. Directly multiplying the class weights by this term inflates the total weight when $\gamma > 0$, since $(1 + H_i)^\gamma \geq 1$ for all observations, and larger values occur near the boundaries. This produces a non-uniform inflation that misrepresents both the total weight and the intended class balance. To avoid this, we normalize the entropy weights within each class so that the total weight of each class remains unchanged. Let

$$S_k(\gamma) = \sum_{j:y_j=k} (1 + H_j)^\gamma$$

denote the within-class sum of entropy terms. The normalized entropy weight for observation i is then given as

$$w_{entropy}(i; \gamma) = \frac{n_{y_i}(1 + H_i)^\gamma}{S_{y_i}(\gamma)}.$$

This normalization ensures that $\sum_{i:y_i=k} w_{entropy}(i; \gamma) = n_k$, which means the entropy weights are only redistributed within each class.

2.2.4 Observation Weights

We obtained the final weight for observation i with label k by combining the class and normalized entropy weights as follows:

$$w_i = \frac{N}{K \times n_{y_i}} \times \frac{n_{y_i}(1+H_i)^\gamma}{S_{y_i}(\gamma)} = \frac{N}{K} \times \frac{(1+H_i)^\gamma}{S_{y_i}(\gamma)}.$$

This formulation ensures that the total sum of the weights is

$$\sum_{i=1}^{N} w_i = \sum_{k=1}^{K} \sum_{i:y_i=k} \frac{N}{K} \times \frac{(1+H_i)^\gamma}{S_{y_i}(\gamma)} = \sum_{k=1}^{K} \frac{N}{K} \times 1 = N.$$

Thus, the class weights controls the balance between classes, ensuring that each class contributes equally to inference, while normalized entropy weights distribute emphasis among observations within each class.

2.3 Imbalance-Aware Bayesian Multinomial Logistic Regression (IA-BMLR) IA-BMLR is a variant of BMLR that modifies the log-likelihood by incorporating the observation weights (see Subsection 2.2.4). The weighted log-likelihood is defined as:

$$\begin{aligned} \ell_w(\boldsymbol{\beta};\gamma) &= \sum_{i=1}^{N} w_i \times \log P(y_i|\mathbf{x}_i,\boldsymbol{\beta}) \\ &= \sum_{i=1}^{N} \left(\frac{N}{K} \times \frac{(1+H_i)^\gamma}{S_{y_i}(\gamma)} \right) \times \log P(y_i|\mathbf{x}_i,\boldsymbol{\beta}) \end{aligned} \tag{3}$$

where $S_{y_i}(\gamma) = \sum_{j:y_j=y_i}(1+H_j)^\gamma$ is the within-class normalization term for the class of observation i. This weighted log-likelihood provides the advantage of class balancing and boundary emphasis. Class weighting mitigates the numerical dominance of majority classes, while entropy-based weighting emphasizes observations in locally informative regions of the feature space. For our model, we define independent normal priors on all the regression coefficients and a half-normal ($\mathcal{HN}$) prior on the entropy parameter as follows:

$$\begin{aligned} \beta_{kj} &\sim \mathcal{N}(0,\sigma_\beta) \quad \text{for } j = 0,\ldots,p \text{ and } k = 1,\ldots,K \\ \gamma &\sim \mathcal{HN}(\sigma_\gamma) \end{aligned} \tag{4}$$

where $\sigma_\beta, \sigma_\gamma > 0$ are the prior scale respectively. For our model, we set these to 1. The choice of a half-normal prior for γ is motivated by the fact that under the normalized formulation, $\gamma = 0$ corresponds to no entropy emphasis and

the weights reduce to pure class weighting. Therefore, the prior concentrates mass near zero while allowing the data to push γ upward when boundary emphasis genuinely improves inference. The joint posterior of IA-BMLR model is thus

$$p(\boldsymbol{\beta}, \gamma | \boldsymbol{X}, \boldsymbol{Y}) \propto p(\boldsymbol{Y} | \boldsymbol{X}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \times p(\boldsymbol{\beta}) \times p(\gamma).$$

This posterior is analytically intractable; thus, we use numerical methods such as the Markov Chain Monte Carlo (MCMC) (see Van Ravenzwaaij et al. (2018) for an introduction) to sample from the posterior distribution. Using MCMC, we obtain a collection of posterior samples $\{\beta^{(m)}\}_{m=1}^{M}$ across multiple chains C, where M is the sampling iterations and C is the number of chains.

2.4 Posterior Predictive Distribution and Uncertainty Quantification

One of the central motivations for the Bayesian framework is its ability to quantify predictive uncertainty, which goes beyond the point predictions available from the frequentist methods. For a new observation $\mathbf{x}^*$, the posterior predictive distribution is expressed as:

$$P(y^* = k | \mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \int P(y^* = k | \mathbf{x}^*, \beta) p(\beta, \gamma | \mathbf{X}, \mathbf{y}) d\beta d\gamma.$$

This integral is then approximated using the posterior samples as follows:

$$\hat{P}(y^* = k | \mathbf{x}^*, \mathbf{X}, \mathbf{y}) = \frac{1}{M} \sum_{m=1}^{M} P(y^* = k | \mathbf{x}^*, \beta^{(m)}). \tag{5}$$

where $P(y^* = k | \mathbf{x}^*, \beta^{(m)})$ is the softmax probability computed under the m-th posterior draw of the parameters. Equation 5 represents the posterior predictive probability, which averages predictions across the posterior distribution of the parameters and therefore accounts for parameter uncertainty. The predicted class is then obtained as

$$\hat{y}^* = \text{argmax}_k P(y^* = k | \mathbf{x}^*, \mathbf{X}, \mathbf{y}).$$

We quantify predictive uncertainty at the observation level using the 95% Highest Density Interval (HDI):

$$\text{HDI}_{ik} = \text{HDI}_{95\%}[\{P(y^* = k | \mathbf{x}_i^*, \beta^{(m)})\}_{m=1}^{M}].$$

A narrow HDI indicates that the model is confident in its estimated probability for class k, while a wide HDI reflects greater uncertainty. When the HDIs of the two most likely classes overlap for a given observation, it suggests that the model cannot clearly distinguish between those classes. This provides insight into prediction uncertainty that point predictions alone cannot reveal.

This kind of observation-level uncertainty quantification arises naturally in the Bayesian framework, whereas many commonly used frequentist classifiers (such as MLR, RF, SVC, and XGB) typically report only point predictions or point probability estimates. The algorithm for the IA-BMLR model is stated in Algorithm 1.

Algorithm 1 IA-BMLR

Require: Training data $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N_{train}}$; hyperparameters: prior scales $(\sigma_\beta, \sigma_\gamma)$, LES neighbors κ (default 10), MCMC settings (samples M, tuning T, chains C, target acceptance α)

Ensure: Posterior samples $\{\boldsymbol{\beta}^{(m)}, \gamma^{(m)}\}_{m=1}^{M \cdot C}$

Preprocessing:

Standardize features to zero mean and unit variance

Identify reference class: $r \leftarrow \arg\max_k n_k$

Compute Local Entropy Scores:

Fit κ-NN model on training data $\{\mathbf{x}_i\}_{i=1}^{N_{train}}$

for $i = 1$ to N_{train} **do**

Find neighbors $\mathcal{N}_\kappa(\mathbf{x}_i)$; compute local class distribution $\{\hat{p}_{ik}\}$

$H_i \leftarrow -\sum_k \hat{p}_{ik} \log(\hat{p}_{ik} + \epsilon) / \log K$

end for

Define Bayesian Model:

$\gamma \sim \text{HalfNormal}(\sigma_\gamma)$ ▷ $\sigma_\gamma = 1$

$\boldsymbol{\beta}_k \sim \mathcal{N}(\mathbf{0}, \sigma_\beta \mathbf{I})$ for $k \neq r$; $\boldsymbol{\beta}_r = \mathbf{0}$ ▷ $\sigma_\beta = 1$

Compute Within-Class Normalized Weights:

$h_i \leftarrow (1 + H_i)^\gamma$ for $i = 1, \ldots, N_{train}$

$S_k \leftarrow \sum_{j:y_j=k} h_j$ for $k = 1, \ldots, K$ ▷ within-class sums

$w_i \leftarrow (N/K) \times (h_i / S_{y_i})$ ▷ ensures $\sum_i w_i = N$

Weighted log-likelihood: $\sum_i w_i(\gamma) \times \log P(y_i \mid \mathbf{x}_i, \boldsymbol{\beta})$

MCMC Sampling:

Run No-U-Turn Sampler (NUTS) with C chains, T tuning iterations, M samples, target acceptance α

return Posterior samples $\{\boldsymbol{\beta}^{(m)}, \gamma^{(m)}\}$

2.5 Model evaluation Before interpreting our results, it is crucial to assess the model's fit to the data and its predictive performance. In our study, we measure the convergence of the sampling chains used, primarily through visual inspection of trace plots, which display the sampled value at each iteration. A numerical assessment involves the Gelman-Rubin statistic ($\hat{R}$) (Gelman and Rubin, 1992), which compares the variance within chains to the variance between chains (Sorensen and Vasishth, 2016). If distributions have converged, the trace plot will look like a hairy caterpillar (Lee and Wagenmakers, 2013) and $\hat{R}$ will be less than 1.1 for all regression parameters (Brooks and Gelman, 1998). When multiple chains are used, the plot shows chain over-

lap upon convergence, whereas non-convergence is indicated by one or more chains being separated.

We use metrics such as accuracy, AUC, precision, recall/balanced accuracy, and F1-score to measure the model's overall predictive performance, and the geometric-mean score (G-Mean) to assess whether the model achieves non-zero recall across all classes. Accuracy is a straightforward measure of a model's overall ability to correctly classify observations, defined as the proportion of correct classifications (true positives and true negatives) among the total number of cases examined. However, accuracy alone may not provide a complete picture, particularly in imbalanced datasets where one class substantially outnumbers another. Thus, it is expedient to consider other metrics, such as precision (positive predictive value), which measures the proportion of positive predictions that are true positives, indicating the model's ability to accurately identify relevant instances. Recall (or sensitivity) assesses the proportion of actual positives correctly identified by the model, reflecting its ability to capture all relevant instances. The F1-score, which is the harmonic mean of precision and recall, offers a balance between the two in cases with an uneven class distribution. G-Mean computes the geometric mean of class-wise recall, penalizing complete failure on any class, while AUC summarizes the model's ability to discriminate between classes across all possible classification thresholds. By evaluating these metrics together, we gain a comprehensive understanding of the model's predictive performance, balancing precision against recall and providing a nuanced view of its performance across different aspects of the classification task. These metrics, combined with visual and numerical convergence checks, provide a robust framework for assessing the model's fit to the data and its predictive accuracy.

2.6 Experiments In this section, we describe the application of our model to nine real-world imbalanced datasets obtained from the KEEL (Derrac et al., 2015) and UCI Machine Learning repositories (Kelly et al., 2011). These datasets span a range of imbalance severities and multiple application domains. Our goal for this analysis is to first examine the overall performance of our model in a multi-class classification context compared with its unweighted variant U-BMLR, and other established methods, RF, SVC, XGB, MLR, for classifying imbalanced datasets, and secondly to explore our model's capability to classify minority classes specifically. Table 1 gives a summary of the dataset characteristics. These datasets are grouped by imbalance ratio, defined as the ratio of the largest class size to the smallest class size.
We ensured there were no missing values in our datasets and performed data preprocessing as follows:

- We split the dataset into training and test sets using five-fold stratified cross-validation (CV). For datasets where the smallest class has fewer than five samples, the number of folds is reduced to the minimum class count (minimum of two).

Table 1: Dataset characteristics. The imbalance ratio (IR) is calculated as the largest class size divided by the smallest class size.

Dataset	Abbrev.	Samples	Features	Classes	IR
Severe Imbalance (ratio > 40)					
Page Blocks	PAG	548	10	5	164.0
Yeast	YEA	1484	8	10	92.6
Ecoli	ECO	336	7	8	71.5
Lymphography	LYM	148	18	4	40.5
Moderate Imbalance (ratio 5-10)					
Glass	GLA	214	9	6	8.4
New Thyroid	THY	215	5	3	5.0
Mild Imbalance (ratio < 5)					
Contraceptive	CON	1473	9	3	1.9
Hayes-Roth	HAY	132	4	3	1.7
Wine	WIN	178	13	3	1.5

- We numerically encode the outcome variables within each fold for each dataset as needed. This encoding ensures that our datasets are compatible with the probabilistic programming language PyMC.

- Data standardization of numerical columns. This is done so that features are on the same scale, preventing larger values from dominating the learning process. It also enhances the performance of the MCMC algorithm in our probabilistic programming. Specifically, we transform each feature using StandardScaler ($x - \mu/\sigma$), which removes the mean (μ) and scales the data to unit variance (σ). In addition, the categorical columns are one-hot encoded. All these are done within each fold, preventing data leakage from the test to the training set.

2.7 Experimental Setup We applied our Bayesian model to the training dataset, using the feature vector $\mathbf{X}_i$ as the independent variable and Y as the dependent variable. As described earlier, multinomial logistic regression extends the binary logistic model to handle more than two outcome categories by estimating a set of coefficients for each category relative to a chosen reference category. The Bayesian model is implemented in Python using the PyMC package (Abril-Pla et al., 2023). PyMC utilizes Markov Chain Monte Carlo (MCMC) methods to sample the posterior distribution of the regression parameters. After an initial burn-in period, it collects samples once the posterior distributions have stabilized (Bertolini et al., 2023). To compute the posterior distributions of the multinomial logistic regression coefficients, we employed four MCMC chains. Each chain ran for 5,000 iterations with a burn-in period

of 5,000 iterations. For computing LES, we set the number of nearest neighbors to $\kappa = 10$ for all datasets. We assessed convergence using the metrics discussed in Subsection 2.5.

The posterior distribution can be used to generate a posterior predictive distribution for future observations. For each held-out fold observation, the model produces a probability vector of length equal to the number of classes, where each element in the vector is the probability that the observation belongs to that class. Classification of a new observation is then carried out by assigning actual labels to the class with the highest probability estimate (i.e., argmax). In addition, for the two Bayesian models, we compute the 95% Highest Density Interval (HDI) of the posterior predictive probabilities across MCMC samples, providing observation-level uncertainty estimates as discussed in Subsection 2.4.

Furthermore, we assess the sensitivity to the choice of κ by conducting a sensitivity analysis that examines IA-BMLR's performance for $\kappa \in \{5, 10, 15, 20\}$ across all nine benchmark datasets, using only the first fold's train-test split. We report the results of this analysis in Tables 9–13 of the Appendix. The code to implement our model is provided on GitHub[4].

3 Results and Discussion

This section presents comprehensive results across six performance metrics on nine benchmark datasets using five-fold stratified CV. Tables 2-7 report the mean and standard deviations (indicating variability) across folds for each metric. In these tables, models are listed in columns and datasets in rows, and bold values indicate the best-performing model for each dataset.

Table 2: Macro-Averaged F1-Score (mean $\pm$ std) across five-fold CV. Bold indicates the best model per dataset.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	0.722 ±0.051	0.751 ±0.126	**0.801 ±0.012**	0.629 ±0.060	0.400 ±0.053	0.702 ±0.129
YEA	0.443 ±0.036	0.526 ±0.033	0.522 ±0.028	0.540 ±0.043	**0.552 ±0.049**	0.508 ±0.095
ECO	0.590 ±0.009	0.621 ±0.037	0.633 ±0.033	**0.637 ±0.011**	0.631 ±0.046	0.625 ±0.018
LYM	**0.745 ±0.123**	0.693 ±0.004	0.645 ±0.058	0.510 ±0.099	0.585 ±0.005	0.626 ±0.037
GLA	0.609 ±0.051	0.555 ±0.075	0.546 ±0.070	**0.751 ±0.060**	0.617 ±0.071	0.736 ±0.078
THY	0.941 ±0.040	0.938 ±0.041	0.936 ±0.048	**0.959 ±0.040**	0.928 ±0.040	0.941 ±0.047
CON	0.506 ±0.026	0.490 ±0.030	0.490 ±0.031	0.497 ±0.015	0.527 ±0.015	**0.539 ±0.021**
HAY	0.587 ±0.074	0.579 ±0.036	0.569 ±0.040	0.821 ±0.033	**0.844 ±0.059**	0.821 ±0.045
WIN	0.983 ±0.023	**0.989 ±0.013**	0.983 ±0.014	0.984 ±0.022	0.983 ±0.023	0.972 ±0.031

When performance is evaluated across F1, accuracy, precision, recall/balanced accuracy, and AUC, no single model consistently dominates across all datasets. Instead, the best-performing model varies across datasets and metrics. For example, classical machine learning models such as RF, SVC, and XGB achieve the highest scores on several datasets (e.g., GLA and HAY), while the proposed IA-BMLR model performs competitively and achieves

[4]https://github.com/Phunmey/IA-BMLR

Table 3: Accuracy (mean ± std) across five-fold CV. Bold indicates best model per dataset.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	0.909 ±0.021	0.953 ±0.014	0.951 ±0.005	0.951 ±0.008	0.927 ±0.005	**0.965 ±0.007**
YEA	0.481 ±0.018	0.595 ±0.027	0.588 ±0.026	**0.613 ±0.029**	0.600 ±0.033	0.600 ±0.044
ECO	0.786 ±0.006	0.866 ±0.015	0.878 ±0.015	0.872 ±0.021	**0.881 ±0.024**	0.878 ±0.003
LYM	0.838	**0.885 ±0.007**	0.872 ±0.034	0.845 ±0.034	0.831 ±0.007	0.831 ±0.007
GLA	0.561 ±0.050	0.645 ±0.066	0.640 ±0.051	**0.799 ±0.010**	0.724 ±0.039	0.790 ±0.053
THY	0.958 ±0.031	0.958 ±0.027	0.958 ±0.031	**0.972 ±0.027**	0.953 ±0.025	0.958 ±0.034
CON	0.511 ±0.026	0.516 ±0.030	0.516 ±0.030	0.523 ±0.007	0.555 ±0.012	**0.561 ±0.015**
HAY	0.568 ±0.079	0.561 ±0.051	0.545 ±0.046	0.796 ±0.037	**0.833 ±0.053**	0.795 ±0.051
WIN	0.983 ±0.023	**0.989 ±0.014**	0.983 ±0.014	0.983 ±0.022	0.983 ±0.022	0.972 ±0.030

Table 4: Macro-averaged precision (mean ± std) across five-fold CV. Bold indicates best model per dataset.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	0.655 ±0.070	0.791 ±0.126	**0.872 ±0.056**	0.662 ±0.084	0.411 ±0.076	0.727 ±0.116
YEA	0.423 ±0.035	0.545 ±0.046	0.553 ±0.050	0.558 ±0.056	**0.584 ±0.054**	0.532 ±0.106
ECO	0.579 ±0.011	0.623 ±0.027	0.635 ±0.025	**0.659 ±0.016**	0.633 ±0.044	0.657 ±0.003
LYM	**0.712 ±0.124**	0.691 ±0.002	0.685 ±0.015	0.545 ±0.142	0.664 ±0.004	0.663 ±0.003
GLA	0.600 ±0.085	0.552 ±0.064	0.557 ±0.070	**0.806 ±0.070**	0.663 ±0.082	0.764 ±0.107
THY	0.952 ±0.056	0.974 ±0.024	**0.982 ±0.013**	0.978 ±0.029	0.980 ±0.011	0.963 ±0.037
CON	0.513 ±0.023	0.501 ±0.030	0.502 ±0.030	0.502 ±0.011	0.539 ±0.008	**0.546 ±0.018**
HAY	0.590 ±0.065	0.608 ±0.046	0.602 ±0.026	0.831 ±0.031	**0.876 ±0.049**	0.829 ±0.045
WIN	0.982 ±0.024	**0.988 ±0.014**	0.983 ±0.014	0.984 ±0.022	0.984 ±0.022	0.973 ±0.030

Table 5: Macro-averaged recall / balanced accuracy (mean ± std) across five-fold CV. Bold indicates best model per dataset.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	**0.919 ±0.015**	0.748 ±0.102	0.775 ±0.014	0.643 ±0.087	0.402 ±0.046	0.707 ±0.138
YEA	**0.549 ±0.039**	0.546 ±0.036	0.528 ±0.028	0.538 ±0.041	0.544 ±0.051	0.504 ±0.094
ECO	0.625 ±0.002	0.643 ±0.025	0.649 ±0.028	0.624 ±0.006	**0.650 ±0.029**	0.617 ±0.020
LYM	**0.795 ±0.121**	0.695 ±0.005	0.628 ±0.079	0.497 ±0.077	0.548 ±0.006	0.613 ±0.057
GLA	0.717 ±0.034	0.580 ±0.092	0.558 ±0.073	**0.746 ±0.046**	0.611 ±0.071	0.742 ±0.068
THY	0.944 ±0.048	0.914 ±0.056	0.906 ±0.069	**0.945 ±0.051**	0.895 ±0.058	0.924 ±0.055
CON	0.521 ±0.021	0.486 ±0.030	0.486 ±0.030	0.497 ±0.015	0.524 ±0.016	**0.539 ±0.021**
HAY	0.605 ±0.087	0.581 ±0.048	0.567 ±0.058	0.823 ±0.032	**0.847 ±0.064**	0.822 ±0.044
WIN	0.986 ±0.018	**0.991 ±0.011**	0.984 ±0.014	0.986 ±0.019	0.983 ±0.023	0.975 ±0.026

the best results on others (e.g., LYM and PAG for certain metrics). On macro-averaged recall/balanced accuracy (which measures the model's ability to correctly identify observations across all classes), IA-BMLR achieves the most wins across datasets, and its AUC performance is comparable to the best baselines. This divergence reflects the trade-off inherent in our model: up-weighting minority-class observations shifts the decision boundary to expand minority-class regions at the expense of majority-class regions, thereby improving minority-class recall at the cost of some overall discrimination.

Based on the G-Mean results in Table 7, we observe that IA-BMLR yields the best mean values across the datasets. IA-BMLR achieves the highest G-Mean on three of nine datasets, second best on four of the datasets, and, more importantly, it is the only method to achieve a non-zero G-Mean on the severely imbalanced Yeast dataset, which has 10 classes and an imbal-

Table 6: Macro-averaged one-vs-rest AUC (mean ± std) across five-fold CV. Bold indicates best model per dataset.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	0.980 ±0.014	0.987 ±0.009	0.982 ±0.009	0.990 ±0.005	0.975 ±0.015	**0.994 ±0.002**
YEA	0.856 ±0.023	0.867 ±0.026	0.869 ±0.023	**0.880 ±0.014**	0.871 ±0.032	0.877 ±0.020
ECO	0.853 ±0.027	0.891 ±0.014	0.928 ±0.004	0.868 ±0.032	0.906 ±0.005	**0.931 ±0.022**
LYM	0.971 ±0.005	**0.972 ±0.006**	0.967 ±0.011	0.964 ±0.008	0.715 ±0.012	0.937 ±0.012
GLA	0.875 ±0.024	0.876 ±0.023	0.858 ±0.023	**0.956 ±0.012**	0.916 ±0.028	0.936 ±0.023
THY	**0.998 ±0.002**	0.997 ±0.003	0.997 ±0.003	0.998 ±0.003	0.998 ±0.002	0.996 ±0.003
CON	0.703 ±0.018	0.702 ±0.020	0.702 ±0.020	0.704 ±0.010	0.721 ±0.015	**0.729 ±0.017**
HAY	0.815 ±0.058	0.810 ±0.059	0.799 ±0.068	0.934 ±0.032	0.879 ±0.043	**0.956 ±0.017**
WIN	**1.000 ±0.001**	**1.000 ±0.001**	0.999 ±0.001	**1.000 ±0.001**	**1.000 ±0.001**	0.997 ±0.005

Table 7: Multiclass geometric mean (mean ± std) across five-fold CV. Bold indicates best model; 0.000 indicates failure on at least one class.

Dataset	IA-BMLR	U-BMLR	MLR	RF	SVC	XGB
PAG	**0.913 ±0.018**	0.508 ±0.362	0.710 ±0.033	0.232 ±0.328	0.000	0.295 ±0.417
YEA	**0.302 ±0.248**	0.000	0.000	0.000	0.000	0.000
ECO	0.000	0.000	0.000	0.000	0.000	0.000
LYM	**0.456 ±0.456**	0.000	0.000	0.000	0.000	0.000
GLA	0.515 ±0.260	0.000	0.000	**0.579 ±0.292**	0.000	0.421 ±0.348
THY	0.938 ±0.055	0.907 ±0.060	0.900 ±0.074	**0.941 ±0.054**	0.885 ±0.062	0.920 ±0.057
CON	0.516 ±0.025	0.470 ±0.033	0.470 ±0.033	0.483 ±0.023	0.507 ±0.023	**0.529 ±0.028**
HAY	0.566 ±0.076	0.555 ±0.042	0.541 ±0.049	0.807 ±0.038	**0.829 ±0.069**	0.807 ±0.051
WIN	0.986 ±0.019	**0.991 ±0.012**	0.984 ±0.014	0.985 ±0.020	0.983 ±0.023	0.974 ±0.028

ance ratio of 92.6, achieving 0.302 ± 0.248, while every other model achieves exactly zero across all five folds. This indicates that all baselines fail to predict at least one minority class, whereas IA-BMLR alone successfully predicts all ten classes. The standard deviation of IA-BMLR on this dataset reflects the natural variation in how well minority classes are recovered across the different fold partitions, rather than model instability. On the Page Blocks dataset, with an imbalance ratio of 164.0, IA-BMLR achieves 0.913 ± 0.018, substantially outperforming U-BMLR (0.508 ± 0.362), RF (0.232 ± 0.328), and XGB (0.295 ± 0.417). The large standard deviations of these models on this dataset show a considerable variability across the folds, in contrast to IA-BMLR's stable performance. SVC achieves a G-Mean score of zero across all folds despite its competitive performance on aggregate metrics.

On the Glass dataset, IA-BMLR achieves 0.515 ± 0.260 while U-BMLR, MLR, and SVC all achieve a G-Mean score of zero. We also observed that the Ecoli dataset proved challenging for all models, with a G-Mean score of zero across all models and folds. These indicate how overwhelming the properties, severe imbalance (> 40), and small sample size (< 500), of these datasets can be even for imbalance-aware approaches. The Lymphography dataset similarly shows universal difficulty, though IA-BMLR achieves 0.456 ± 0.456. The large standard deviation reflects considerable variability across the two folds and is consistent with the tiny sample size (N = 148, two-fold CV), preventing reliable estimation. Overall, our model's performance demonstrates

that the weighted likelihood formulation prevents failure of the minority class in most scenarios in which such failure would otherwise occur.

When we compare IA-BMLR with its unweighted variant, U-BMLR, and MLR, we observe that U-BMLR achieves a G-Mean score of zero on four of nine datasets, whereas IA-BMLR achieves a G-Mean score of zero on only one dataset (ECO). MLR exhibits a pattern similar to U-BMLR, confirming that the Bayesian framework alone does not prevent minority-class failure; the weighted likelihood is the critical component. In terms of recall and balanced accuracy, IA-BMLR outperforms U-BMLR and MLR across seven of the nine datasets. This direct comparison between the models demonstrates that the weighted likelihood captures IA-BMLR's advantage in imbalanced classification tasks. To understand when entropy weighting is beneficial, we examine the learned values of γ across datasets. Table 8 reports the posterior means $\pm$ cross-fold standard deviation, and 95% HDI from the last fold's ArviZ summary.

Table 8: Posterior summary of the entropy parameter $\hat{\gamma}$ by dataset. Mean $\pm$ std are computed across CV folds. 95% HDI is from the summary of the last fold.

Dataset	**Imbalance Ratio**	$\hat{\gamma}$ **(Mean $\pm$Fold Std)**	**95% HDI**
PAG	164.0	0.286 $\pm$0.025	[0.000, 0.895]
YEA	92.6	0.040 $\pm$0.003	[0.000, 0.124]
ECO	71.5	0.354 $\pm$0.004	[0.000, 0.990]
LYM	40.5	0.567 $\pm$0.057	[0.000, 1.426]
GLA	8.4	0.327 $\pm$0.083	[0.000, 1.341]
THY	5.0	0.396 $\pm$0.048	[0.000, 1.034]
CON	1.9	0.139 $\pm$0.035	[0.000, 0.318]
HAY	1.7	0.484 $\pm$0.021	[0.000, 1.343]
WIN	1.5	0.373 $\pm$0.027	[0.000, 1.099]

Under the normalized model formulation, γ controls only the distribution of weights within each class, thereby capturing the true impact of entropy weighting. The posterior means show no clear relationship between $\hat{\gamma}$ and the imbalance ratio. The Yeast dataset (IR =92.6), despite being one of the most severely imbalanced datasets, learns the lowest $\hat{\gamma}$ (0.040 ± 0.003), while the Lymphography dataset (IR =40.5) on the contrary, learns the highest $\hat{\gamma}$ (0.567 ± 0.057). This is consistent with γ responding to the local geometric structure of decision boundaries within each class, a dataset with complex overlapping class boundaries may benefit more from entropy-based boundary emphasis than a severely imbalanced dataset whose classes are well-separated in feature space. On datasets where $\hat{\gamma} \approx 0$, with narrow 95% HDIs concentrated near zero (e.g., [0.000, 0.124]), the model learns that class weights alone are sufficient and that additional emphasis on locally informative observations does not improve prediction. Conversely, higher values of $\hat{\gamma}$ observed suggest that entropy-based weighting provides additional benefit beyond class

weighting alone in these cases. In this context, the practical implication is that the learnable γ provides automatic adaptation of our model. When entropy weighting helps, the model learns a positive γ; when it does not, the model reduces to class-weighted Bayesian multinomial logistic regression, with $\gamma \approx 0$. This form of data-driven adaptation is an advantage over fixing γ as a parameter.

The degree to which γ can reflect local boundary structure is, however, constrained by the linearity of the multinomial logistic regression predictor. A more flexible predictor, such as one based on Gaussian processes or spline functions, could potentially yield more pronounced and interpretable $\hat{\gamma}$ values by capturing local boundary structure more effectively. Exploring this type of extension is a natural direction for future work.

To further quantify the nature of the uncertainty produced by the two Bayesian models (IA-BMLR and U-BMLR), Figure 1 presents the per-class difference in posterior predictive HDI width between IA-BMLR and U-BMLR across all nine datasets. For each class in each dataset, a positive value indicates that IA-BMLR produces wider predictive intervals than U-BMLR for observations in that class, and a negative value indicates the reverse. We expect a well-calibrated, imbalance-aware model to show positive differences on minority classes (where genuine uncertainty is high) and near-zero differences on majority classes, where both models should agree.

Figure 1 shows that for datasets with severe imbalance (PAG, ECO, LYM), IA-BMLR consistently produces wider predictive intervals across most classes. This suggests that the imbalance-aware weighted likelihood appropriately reflects increased uncertainty when data for certain classes are limited. The YEA datasets show a similar but weaker pattern, with mostly positive differences and only very small negative values for a few classes. GLA dataset shows a more mixed pattern. IA-BMLR shows greater uncertainty for one glass type, while for the remaining types, the differences are small or slightly negative differences. In contrast, the differences are close to zero for datasets with relatively mild imbalance (CON, HAY, and WIN). This indicates that IA-BMLR does not unnecessarily inflate uncertainty when class imbalance is limited. Overall, the results suggest that the proposed method adapts its uncertainty estimates to the degree of class imbalance present in the data.

4 Conclusion

In this study, we present the Imbalance-Aware Bayesian Multinomial Logistic Regression model (IA-BMLR), which adapts the weight-loss framework of Zare et al. (2025) to Bayesian discriminative multi-class classification. Our model incorporates observation-specific weights combining inverse-frequency class weights with entropy-based boundary emphasis directly into the likelihood function. One of our contributions is to treat the entropy weight parameter γ as a learnable variable with a half-normal prior, enabling automatic calibration without manual tuning. In addition, we adopted the within-

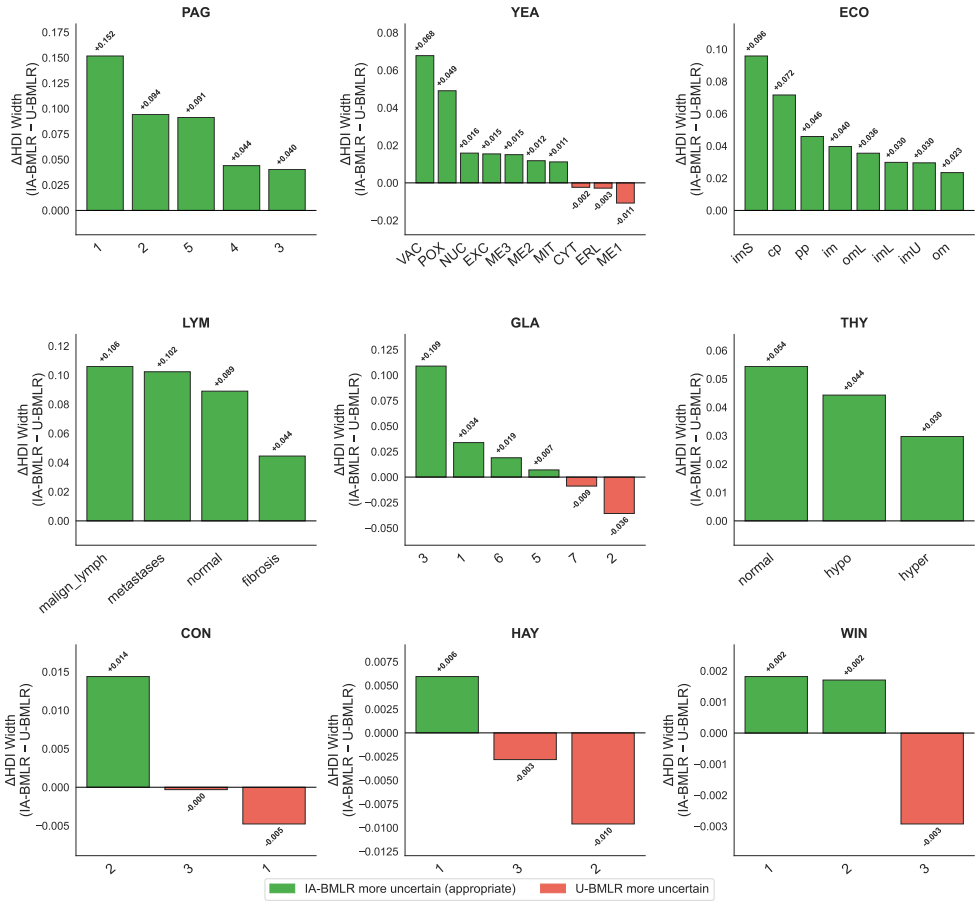

Fig. 1: Per-class difference in predictive uncertainty between IA-BMLR and U-BMLR. Bars show the difference in the width of the 95% posterior predictive Highest Density Interval (HDI) for each class, computed as IA-BMLR − U-BMLR. Positive values (green) indicate that IA-BMLR expresses greater uncertainty for that class, while negative values (red) indicate greater uncertainty under U-BMLR. Within each dataset, classes are ordered from the largest to the smallest difference in HDI width.

class normalization of the entropy weights, which ensures $\sum_i w_i = N$ for any value of γ. This guarantees that the weighted likelihood treats the data as providing exactly N units of information, preserving posterior calibration regardless of the learned entropy weighting. The normalized model formulation also clarifies the conceptual separation between the two weight components: class weights handle between-class balancing, while normalized entropy weights handle within-class redistribution of emphasis toward boundary observations. Furthermore, we assessed the performance of IA-BMLR against established machine learning methods on imbalanced datasets.

The experimental results show that although no model consistently outperforms the others, on standard metrics like F1-score, tree-based ensemble models, such as RF, often perform well, while IA-BMLR performs competitively. The performance of these models can be attributed to facts about their flexibility in modeling complex decision boundaries. However, on metrics explicitly designed for imbalanced classification, particularly G-Mean, IA-BMLR demonstrates an advantage, achieving the best performance on three datasets and second-best on four of nine. Also, IA-BMLR is the only model with a non-zero G-Mean on two severely imbalanced datasets, where all baselines fail. This pattern illustrates the trade-off in imbalance-aware models: up-weighting minority-class observations shifts the decision boundary to expand minority-class regions at the expense of majority-class regions, thereby improving minority-class recall at the cost of majority-class precision. When we evaluate using the F1-score, which weighs precision and recall equally, this trade-off may appear unfavorable when minority recall is prioritized over majority precision. However, when we assess using G-Mean, which requires non-zero recall across all classes, the trade-off indicates that it is beneficial.

Focusing on the Yeast dataset results, with 10 classes and an imbalance ratio of 92.6, it is clear that this is a challenging multiclass imbalance problem. The failure of all baseline models (G-Mean equal to zero) indicates that, without explicit handling of imbalance, even established models such as XGBoost and Random Forest will ignore the smallest minority classes. These models are designed to optimize objectives that implicitly weight observations equally, so that when a class comprises a smaller percentage of the data, its contribution to the objective is negligible. IA-BMLR's inverse-frequency class weights directly address this by ensuring minority classes contribute proportionally to inference. Beyond point predictions, the Bayesian framework enables quantification of observation-level uncertainty through the posterior predictive HDI. Our results show that IA-BMLR produces wider predictive intervals for minority classes than U-BMLR, indicating greater uncertainty where classification is most difficult, an aspect that frequentist baselines do not provide. Our model is most valuable when (1) there is severe class imbalance accompanied by an adequate sample size and (2) uncertainty quantification is desired.

Some limitations of our work include, first, the higher computational cost of our model relative to optimization-based models, owing to MCMC sampling. In scenarios requiring rapid training or deployment on very large datasets, alternative sampling methods, such as variational inference approximations, will be employed. Second, our model is unable to overcome fundamental data limitations when the minority class contains very few observations. The weighted likelihood in our model is designed to ensure that minority classes contribute to inference; however, if those classes are too small to estimate reliable decision boundaries, our predictions will be poor, and no amount of reweighting can compensate for insufficient information. Third, by computing local entropy scores, we introduce additional preprocessing cost due to

nearest-neighbor searches, which may affect scalability for very large datasets. This means that for very large datasets, we may require approximate nearest neighbor methods to maintain tractability.

Our future work will focus on developing variational inference approximations for scaling the model. Also, our model will be extended to hierarchical and multi-label settings. Additionally, a prior sensitivity analysis of the entropy parameter γ, particularly across alternative prior families or scales, would help assess the robustness of the learned weighting. Extending IA-BMLR to more flexible predictors, such as those based on Gaussian processes or spline functions, could yield more pronounced and interpretable $\hat{\gamma}$ values by capturing local boundary structure more effectively, and it is a promising direction for future work. Finally, a theoretical analysis of the properties of our weighted likelihood will be an important direction for future work.

References

Abril-Pla O., Andreani V., Carroll C., Dong L., Fonnesbeck C.J., Kochurov M., Kumar R., Lao J., Luhmann C.C., Martin O.A., Osthege M., Vieira R., Wiecki T., and Zinkov R. (2023) PyMC: a modern and comprehensive probabilistic programming framework in Python, *PeerJ Computer Science*, 9:e1516. DOI: 10.7717/peerj-cs.1516

Al-Ashoor A. and Abdullah S. (2022) Examining techniques to solving imbalanced datasets in educational data mining systems, *International Journal of Computers*, 21(2):205–213.

Bai L., Ju T., Wang H., Lei M., and Pan X. (2024) Two-step ensemble undersampling algorithm for massive imbalanced data classification, *Information Sciences*, 665:120351. DOI: 10.1016/j.ins.2023.120351

Bertolini R., Finch S.J., and Nehm R.H. (2023) An application of Bayesian inference to examine student retention and attrition in the STEM classroom, *Frontiers in Education*, 8:1073829. DOI: 10.3389/feduc.2023.1073829

Brooks S.P. and Gelman A. (1998) General methods for monitoring convergence of iterative simulations, *Journal of Computational and Graphical Statistics*, 7(4):434–455. DOI: 10.1080/10618600.1998.10474787

Chawla N.V., Bowyer K.W., Hall L.O., and Kegelmeyer W.P. (2002) SMOTE: synthetic minority over-sampling technique, *Journal of Artificial Intelligence Research*, 16:321–357. DOI: 10.1613/jair.953

Derrac J., Garcia S., Sanchez L., and Herrera F. (2015) KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, *Journal of Multiple-Valued Logic and Soft Computing*, 17:255–287.

Di Martino M., Decia F., Molinelli J., and Fernández A. (2012) Improving electric fraud detection using class imbalance strategies. In *Proceedings of the International Conference on Pattern Recognition Applications and Methods (IPRAM 2012)*.

Ganganwar V. (2012) An overview of classification algorithms for imbalanced datasets, *International Journal of Emerging Technology and Advanced Engineering*, 2(4):42–47.

Gelman A. and Rubin D.B. (1992) Inference from iterative simulation using multiple sequences, *Statistical Science*, 7(4):457–472. DOI: 10.1214/ss/1177011136

Guo J., Wu H., Chen X., and Lin W. (2024) Adaptive SV-Borderline SMOTE-SVM algorithm for imbalanced data classification, *Applied Soft Computing*, 150:110986. DOI: 10.1016/j.asoc.2023.110986

Kalina J. (2022) Model choice for regression models with a categorical response, *Journal of Applied Mathematics, Statistics and Informatics*, 18(1):59–71.

Kelly M., Longjohn R., and Nottingham K. (2011) *The UCI Machine Learning Repository*. Available at: https://archive.ics.uci.edu.

Kotsiantis S., Kanellopoulos D., and Pintelas P. (2006) Handling imbalanced datasets: a review, *GESTS International Transactions on Computer Science and Engineering*, 30(1):25–36.

Kubat M., Holte R.C., and Matwin S. (1998) Machine learning for the detection of oil spills in satellite radar images, *Machine Learning*, 30:195–215. DOI: 10.1023/A:1007452223027

Lazic S.E. (2026) A weighted-likelihood framework for class imbalance in Bayesian prediction models. *Computational Toxicology*, 100400

Lee M.D. and Wagenmakers E.J. (2013) *Bayesian Data Analysis for Cognitive Science: A Practical Course*. Cambridge University Press, New York.

Liu H., Huang X., and Liu X. (2024) Improve label embedding quality through global sensitive GAT for hierarchical text classification, *Expert Systems with Applications*, 238:122267. DOI: 10.1016/j.eswa.2023.122267

Rawashdeh H., Awawdeh S., Shannag F., Henawi E., Faris H., Obeid N., and Hyett J. (2020) Intelligent system based on data mining techniques for prediction of preterm birth for women with cervical cerclage, *Computational Biology and Chemistry*, 85:107233. DOI: 10.1016/j.compbiolchem.2020.107233

Shatnawi R. (2012) Improving software fault-prediction for imbalanced data. In *Proceedings of the 2012 International Conference on Innovations in Information Technology (IIT)*, pages 54–59. IEEE.

Sorensen T. and Vasishth S. (2016) Bayesian linear mixed models using Stan: a tutorial for psychologists, linguists, and cognitive scientists. *The Quantitative Methods for Psychology*, 12(3):175-200

Stando A., Cavus M., and Biecek P. (2024) The effect of balancing methods on model behavior in imbalanced classification problems. In *Proceedings of the Fifth International Workshop on Learning with Imbalanced Domains: Theory and Applications*, pages 16–30. PMLR.

Tanha J., Abdi Y., Samadi N., Razzaghi N., and Asadpour M. (2020) Boosting methods for multi-class imbalanced data classification: an experimental review, *Journal of Big Data*, 7(1):1–47. DOI: 10.1186/s40537-020-00349-y

Van Ravenzwaaij D., Cassey P., and Brown S.D. (2018) A simple introduction to Markov chain Monte-Carlo sampling, *Psychonomic Bulletin & Review*, 25(1):143–154. DOI: 10.3758/s13423-016-1015-8

Van de Schoot R., Depaoli S., King R., Kramer B., Märtens K., Tadesse M.G., Vannucci M., Gelman A., Veen D., Willemsen J., and Yau C. (2021) Bayesian statistics and modelling, *Nature Reviews Methods Primers*, 1(1):1–26. DOI: 10.1038/s43586-020-00001-2

Zare A., Zare A., Pezeshki P.S., Ebrahimi A., Vázquez-García I., and Celi L.A. (2025) Uncertainty-aware generative oversampling using an entropy-guided conditional variational autoencoder. arXiv preprint arXiv:2509.25334.

Appendix In this section, we present the results of the sensitivity analysis for κ. In each table, the rows are datasets while the columns are κ values. The range measures the sensitivity to this κ and is calculated as range = (maximum - minimum) across κ values.

Table 9: Balanced Accuracy across $\kappa \in \{5, 10, 15, 20\}$.

Dataset	κ=5	κ=10	κ=15	κ=20	Range
PAG	0.968	0.902	0.902	0.902	0.067
YEA	0.582	0.583	0.583	0.568	0.015
ECO	0.627	0.627	0.627	0.627	0.000
LYM	0.674	0.680	0.680	0.680	0.006
GLA	0.784	0.784	0.784	0.784	0.000
THY	1.000	1.000	1.000	1.000	0.000
CON	0.543	0.543	0.543	0.543	0.000
HAY	0.648	0.648	0.648	0.648	0.000
WIN	1.000	1.000	1.000	1.000	0.000

Table 10: Mean Local Entropy Score (LES Mean) across $\kappa \in \{5, 10, 15, 20\}$.

Dataset	κ=5	κ=10	κ=15	κ=20	Range
PAG	0.049	0.064	0.066	0.066	0.017
YEA	0.306	0.377	0.409	0.429	0.123
ECO	0.144	0.198	0.228	0.252	0.108
LYM	0.291	0.374	0.415	0.442	0.151
GLA	0.287	0.353	0.429	0.491	0.204
THY	0.100	0.121	0.145	0.174	0.073
CON	0.666	0.795	0.840	0.858	0.193
HAY	0.622	0.698	0.741	0.742	0.119
WIN	0.103	0.139	0.172	0.209	0.106

Table 11: Posterior mean of the entropy parameter $\hat{\gamma}$ across $\kappa \in \{5, 10, 15, 20\}$.

Dataset	κ=5	κ=10	κ=15	κ=20	Range
PAG	0.403	0.257	0.256	0.292	0.147
YEA	0.046	0.041	0.058	0.093	0.052
ECO	0.473	0.363	0.388	0.424	0.110
LYM	0.505	0.614	0.669	0.734	0.228
GLA	0.276	0.242	0.281	0.361	0.119
THY	0.287	0.354	0.463	0.692	0.405
CON	0.138	0.176	0.204	0.222	0.084
HAY	0.441	0.471	0.510	0.473	0.070
WIN	0.416	0.389	0.397	0.387	0.030

Table 12: Multiclass Geometric Mean across $\kappa \in \{5, 10, 15, 20\}$.

Dataset	κ=5	κ=10	κ=15	κ=20	Range
PAG	0.968	0.892	0.892	0.892	0.075
YEA	0.522	0.523	0.523	0.507	0.016
ECO	0.000	0.000	0.000	0.000	0.000
LYM	0.000	0.000	0.000	0.000	0.000
GLA	0.713	0.713	0.713	0.713	0.000
THY	1.000	1.000	1.000	1.000	0.000
CON	0.542	0.542	0.542	0.542	0.000
HAY	0.602	0.602	0.602	0.602	0.000
WIN	1.000	1.000	1.000	1.000	0.000

Table 13: Macro-Averaged F1-Score across $\kappa \in \{5, 10, 15, 20\}$.

Dataset	κ=5	κ=10	κ=15	κ=20	Range
PAG	0.814	0.773	0.773	0.773	0.041
YEA	0.494	0.496	0.496	0.486	0.009
ECO	0.581	0.581	0.581	0.581	0.000
LYM	0.622	0.628	0.628	0.628	0.007
GLA	0.677	0.677	0.677	0.677	0.000
THY	1.000	1.000	1.000	1.000	0.000
CON	0.534	0.534	0.534	0.534	0.000
HAY	0.605	0.605	0.605	0.605	0.000
WIN	1.000	1.000	1.000	1.000	0.000

The results show that the IA-BMLR's performance metrics are generally stable across the range of examined κ values. While the internal entropy-related quantities (LES and the posterior entropy parameter) vary with κ, as expected for neighborhood-based estimators, these variations do not translate into meaningful changes in classification performance. This means that our proposed method is robust to the choice of κ. Based on these results, we retain $\kappa = 10$ as a representative default for datasets of moderate sizes.