MiGRoW: A Multi-Perspective Feature Weighting Scheme Using Gini, Entropy, and Mutual Information

Md Rakibul Islam Midul[1,], Aryan Kumar Singh[2], Md Shohan Mia[2], and Dipjyoti Das[2]

[1] Department of Electronics and Communication Engineering, National Institute of Technology Silchar

[2] Department of Computer Science and Engineering, National Institute of Technology Silchar

Corresponding author(s):dipjyoti@ece.nits.ac.in **Contributing author(s):**rakibuli_ug_22@ece.nits.ac.in aryan21_ug@cse.nits.ac.in md.shohan21_ug@cse.nits.ac.in

Abstract. Feature weighting improves model effectiveness by correctly assigning significance to input features. This paper introduces **MiGRoW** (Mutual information, Gini, Redundancy, and Weighting), a new feature weighting system that combines global and local relevance estimation and redundancy adjustment. Local significance is expressed via Gini impurity in K-Means-based grouped subsets, feature variability is measured in terms of entropy, global significance is measured by mutual information, and redundancy in features is reduced using mutual interaction analysis. By integrating both global and local views, MiGRoW avoids the drawbacks of traditional global-only weighting schemes and supports adaptive and context-sensitive importance estimation. The method is tested on five benchmark datasets—four from the UCI ML repository and one from Kaggle—on a variety of classifiers involving Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, and Multi-Layer Perceptron. Experimental outcomes show that MiGRoW outperforms state-of-the-art techniques like RAW, LASSO, USP, WB, and TabTransformer consistently across metrics like accuracy, precision, recall, and F1-score. The suggested framework presents a light, interpretable, and efficient solution for feature weighting in high-dimensional and heterogeneous data settings.

Keywords: Feature weighting, Mutual Information, Gini Impurity, Redundancy Control, Entropy, Clustering, Random Forest.

1 Introduction

Feature weighting refers to the process of assigning varying levels of importance to different features in a dataset. It plays a fundamental role in machine learning by improving interpretability, reducing overfitting, and guiding models toward better generalization. Not all features contribute equally in predictive modeling tasks; some have greater predictive potential, while others may generate noise. Recognizing and leveraging these disparities is critical to developing robust models. Traditional methods to feature importance usually view the dataset as a homogenous entity, assuming that features are equally relevant throughout the data distribution. In real-world datasets, a feature's relevance may change over different areas or segments of the data, generally referred to as local regions [4] . Local feature weighting adjusts this variance by altering the relevance of features depending on specific characteristics of individual data segments.

Traditional feature weighting techniques provide basic mechanisms for the selection of relevant features and handling of class imbalance [17]. State-of-the-art (SOTA) feature weighting techniques can be divided into statistical and learning-based techniques. Techniques such as RAW utilize the original feature space without any transformation, and LASSO [22] uses coefficient regularization to reduce less important features. Sampling techniques such as Undersampling (USP) [9] and Weighted Bootstrapping (WB) [3] modify instance distributions for handling class imbalance. More recently, transformer-based models like Tab Transformer (TabT) [10] have improved feature learning by capturing contextual feature interactions using attention mechanisms. Although these models are effective, they come with the disadvantage of high computational expenses, memory requirement, and a lack of interpretability.

To address these issues, we introduce MiGRoW (Mutual information, Gini, Redundancy, and Weighting)—an innovative and explainable feature weighting algorithm that brings together local and global relevance estimations with redundancy penalty. Different from current approaches, MiGRoW unifies local feature relevance through Gini impurity from clustered subgroups, feature variance through entropy, and global feature relevance through mutual information. In addition to penalizing redundancy, it adds interaction penalties between features based on mutual information to further eliminate redundancy. This composite weighting framework enables MiGRoW to capture subtle patterns in heterogeneous data sets, providing enhanced performance while ensuring scalability and interpretability.
The key contributions of this work are identified as follows:

- A new feature weighting scheme, **MiGRoW**, is introduced by combining local relevance (via Gini impurity in clustered subsets), feature variability (via entropy), global relevance (via mutual information), and redundancy control (via mutual feature interaction).
- The inadequacies of traditional global-only relevance models are overcome by the inclusion of data-driven local context via clustering to facilitate more adaptive and more precise feature estimation.
- The suggested approach is tested on five varied datasets, four from the UCI ML repository and one from Kaggle, employing various classifiers like Random Forest, Logistic Regression, Naive Bayes, K-Nearest Neighbors, and Multi-Layer Perceptron.

– Comparison with existing state-of-the-art methods—such as LASSO, RAW, USP, WB, and TabT—show that **MiGRoW** is a light-weight, interpretable, and competitive solution, especially designed for heterogeneous data environments.

The rest of the paper is organized as follows: Section 2 summarizes relevant literature and emphasizes the shortcomings of current feature weighting methods. Section 3 outlines the methodology of the proposed MiGRoW framework. Section 4 presents the experimental settings, and Section 5 discusses the result analysis. Finally, Sections 6 and 7 provide the discussion and conclusions with future directions.

2 Literature Reviews

Hussain et al. [11] created two clustering algorithms: Weighted Multiview K-Means (W-MV-KM) and Weighted Multiview K-Means with L2 Regularization. These approaches improve on the traditional K-Means algorithm by assigning optimal weights to both features and data views. The L2 regularization in W-MV-KM-L2 improves model resilience while preventing overfitting. Experimental results show that W-MV-KM-L2 consistently beat other current approaches in terms of clustering accuracy over a wide range of synthetic and real-world datasets.

Abid-Althaqafi et al. [1] have introduced a feature weighting approach that applies importance-based weights to improve classification accuracy, which was evaluated on the ArPFN dataset using Scikit-learn and 10-fold cross-validation. While the technique increases performance by stressing crucial features, it has numerous drawbacks, including high computational cost, noise sensitivity, and limited cross-domain flexibility.

Zhang et al. [24] have presented Tabular Feature Weighting with Transformer (TFWT), a technique that assigns optimum feature weights using self-attention and reinforcement learning. It improves classification accuracy by up to 27% compared to raw models, outperforming approaches such as LASSO and Tab-Transformer. However, it is still constrained by high computational requirements, hyperparameter sensitivity, and low interpretability.

Mamata et al. [7] have compared Term Frequency-Inverse Document Frequency (TF-IDF) with N-Gram feature weighting for text classification, concluding that TF-IDF was more successful, particularly with Random Forest, obtaining 93.81% accuracy and a 91.99% F1 score. However, the study lacks a comparison to deep learning approaches, which might boost outcomes even further.

Chen et al. [5] have introduced Feature Weighted Non-negative Matrix Factorization (FNMF), an improved NMF algorithm that adjusts feature priority to filter out noisy features. While it outperforms traditional methods on noisy datasets, its performance is strongly reliant on precise parameter adjustment.

Zhenmao Li et al. [15] have developed the Learning to Auto Weight (LAW) technique, which uses data-driven strategies to improve model stability and accuracy on noisy, unbalanced datasets. While exceeding MentorNet and Focal Loss, its performance is dependent on dataset structure and may require adaptation for new distributions.

Yan Xu et al. [23] have developed a K-means variant that uses Information Gain and ReliefF for feature weighting, improving clustering using an updated distance metric. The methodology increases accuracy and minimizes Sum of Squared Errors (SSE), but its success is dependent on parameter selection within weighting methods, such as ReliefF's kernel width.

Panday et al. [19] developed two unsupervised feature selection methods: meanFSFW and maxFSFW, which use cluster-based weighting using the Intelligent Minkowski Weighted K-means (imwk-means) algorithm. These methods improved clustering accuracy (ARI) while successfully reducing noise, exceeding Feature Similarity (FSFS) and Multi-Cluster Feature Selection (MCFS) across a variety of datasets. However, they need careful parameter tuning and a balance between clustering efficacy and decreasing dimensionality.

Masramon et al. [18] improved the Relief approach by incorporating Double Relief and pdReliefF, which iteratively update feature weights during distance computations. Especially pdReliefF shows better resistance to irrelevant and redundant features across datasets. However, their efficiency varies per dataset, and they remain vulnerable to noise and complexity, with progressive weighting providing modest advantage due to quick convergence.

Sun et al. [21] have developed a local-learning feature selection technique using L1 regularized logistic regression to optimize feature weighting. It excels in accuracy and resilience in high-dimensional instances, while effectively removing extraneous elements. However, its performance is sensitive to parameter changes, computational load increases with larger sample sizes, and it exclusively addresses multiclass difficulties.

This literature review outlines many feature weighting and selection strategies for clustering and classification, such as weighted k-means, transformer-based models, and local learning approaches. While these approaches usually enhance accuracy, robustness, and noise resistance, they frequently confront constraints including high computing cost, hyperparameter sensitivity, and reliance on precise parameter tuning

3 Proposed Methodology

This paper proposes a new feature weighting approach to enhance classification performance by considering local relevance, feature variability, and redundancy. The technique optimizes feature relevance and redundancy using local significance measurements, entropy-based variability, and mutual data-driven interaction penalties. The flowchart is shown in fig. 1 .

3.1 Data Normalization

To ensure comparability between features with various units and scales, all features are initially adjusted using z-score normalization.:

$$X'_{i,j} = \frac{X_{i,j} - \mu_j}{\sigma_j} \tag{1}$$

where $X_{i,j}$ denotes the original value of the j^{th} feature for the i^{th} sample, and μ_j and σ_j represent the mean and standard deviation of feature j, respectively [12].

3.2 Clustering for Local Context

The normalized dataset X' is partitioned into k clusters using the K-Means approach to account for local changes in feature relevance across different regions [16]:

$$\{C_1, C_2, \ldots, C_k\} = \text{KMeans}(X', k) \tag{2}$$

This clustering allows for calculating of feature importance within more homogeneous subpopulations.

3.3 Local Gini Importance

In each cluster C_j, a Random Forest classifier [4] is trained to evaluate the value of each feature i using the Gini importance metric:

$$GI_{i,j} = \text{GiniImportance}(i, C_j) \tag{3}$$

Averaging over all clusters yields the overall Gini relevance of feature i:

$$GI_i = \frac{1}{k} \sum_{j=1}^{k} GI_{i,j} \tag{4}$$

This technique identifies feature significance that may vary across data subsets.

3.4 Entropy-Based Variability

To account for the informativeness of features based on their value distributions, the normalized entropy E_i of each feature i is calculated as follows::

$$E_i = -\sum_{v \in V_i} p(v) \log p(v) \tag{5}$$

where V_i represents the set of unique values for feature i, and $p(v)$ is the experimental probability of value v. Features with higher entropy show more variability and hence are allocated higher weights. [20].

3.5 Redundancy Penalty via Mutual Information

To reduce redundancy among features, the mutual information (MI) between each feature i and the goal variable Y is calculated as:

$$MI_i = \sum_{x_i} \sum_{y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} \tag{6}$$

Furthermore, redundancy R_i is evaluated by calculating the average MI between feature i and all other features, which reflects overlapping information. High redundancy characteristics are punished to prevent the selection of correlated features[6].

$$R_i = \frac{1}{m-1} \sum_{\substack{j=1 \\ j \neq i}}^{m} MI(i, j) \tag{7}$$

3.6 Final Feature Weight Calculation

The final weight w_i for each feature i is computed by integrating all the relevance and redundancy components as:

$$w_i = GI_i \times E_i \times MI_i \times (1 - \alpha \times R_i) \tag{8}$$

Here, GI_i denotes the average Gini importance (local relevance), E_i is the entropy (variability), MI_i represents the mutual information between feature i and the target Y (global relevance), and R_i quantifies redundancy as the average mutual information between feature i and all other features. The hyperparameter $\alpha \in [0, 1]$ controls the penalty imposed for redundancy.

Algorithm 1 MiGRoW Algorithm

Require: Dataset X, target variable Y, number of clusters k, redundancy penalty coefficient α

Ensure: Feature weights $\mathbf{w} = \{w_1, w_2, \ldots, w_m\}$ for m features

Normalize features X' via z-score normalization
Partition X' into k clusters $\{C_1, \ldots, C_k\} \leftarrow \text{KMeans}(X', k)$
for $j = 1$ to k **do**
    Train Random Forest classifier on cluster C_j and compute Gini importances $GI_{\cdot,j}$
end for
Compute average Gini importance: $GI_i \leftarrow \frac{1}{k}\sum_{j=1}^{k} GI_{i,j}$ for each feature i
for each feature i **do**
    Compute entropy E_i of feature i
    Compute mutual information MI_i between feature i and target Y
    Estimate redundancy R_i as average MI between feature i and other features:

$$R_i = \frac{1}{m-1} \sum_{j=1, j \neq i}^{m} MI(x_i, x_j)$$

    Compute final feature weight:

$$w_i = GI_i \times E_i \times MI_i \times (1 - \alpha \times R_i)$$

end for
return w

4 Experimental Settings

4.1 Datasets

In our study, we evaluated the proposed method with four UCI ML [8] and one Kaggle reciprocity dataset. The datasets are summarized in Table 1.

i)Online Shoppers Purchasing Intention Dataset(OS) [9] sourced from UCI, it contains a mix of real and integer both type attributes which is related to user behaviour on an e-commerce website.

ii)MAGIC Gamma Telescope Dataset (MA) [13] from UCI, this dataset simulates gamma particle detection in a gamma telescope.

iii)Smoking and Drinking Dataset with Body Signal (SD) [14] sourced from the Korean National Health Insurance Service and hosted on Kaggle, includes physiological body signal features.

iv)Car Evaluation Dataset(CE) [2]from UCI, this dataset involves evaluation of cars based on various attributes like price, maintenance cost and safety.

4.2 Classification Models

This study assesses the proposed model on a diversity of classification tasks, utilizing algorithms such as Random Forests (RF), Logistic Regression (LR),

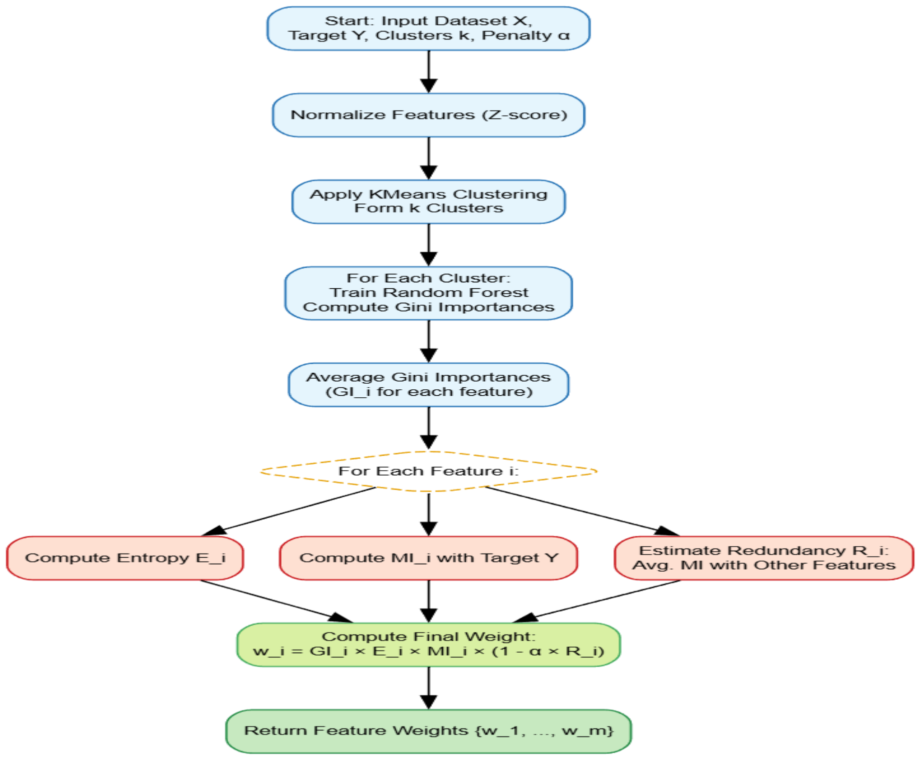

Fig. 1: Flowchart of the MiGRoW Feature Weighting Algorithm

Naive Bayes (NB), K-Nearest Neighbors (KNN), and Multilayer Perceptrons (MLP). The performance of each classifier is compared with and without the integration of our method to evaluate its effectiveness. To assess the effectiveness of our strategy, we evaluate the performance of each classifier with and without our strategy.

4.3 Baseline Models

To demonstrate the effectiveness of our proposed MiGRoW technique, we compare it against four widely recognized baseline techniques. Specifically, LASSO and TabTransformer are used for feature preprocessing, whereas Weighted Bootstrapping (WB) and Undersampling (USP) are used for sample weight preprocessing.

i)Undersampling (USP)[9] is a method that balances the dataset by reducing the number of samples from the majority class, aligning it with the size of the minority class. This technique contributes to mitigate bias and improves model fairness in model predictions.

Table 1: Summary of all datasets

Dataset	Instances	Features	Classes
OS	12330	17	2
MA	19020	10	2
SD	991346	23	2
CE	1728	6	4
ST	1000	20	2

ii)Least Absolute Shrinkage and Selection Operator (LASSO)[4] is a regression technique that improves model performance by absolute magnitude of the regression coefficients. This technique aids in filtering out irrelevant features, simplifying the model structure and enhances the prediction accuracy.

iii)Weighted Bootstrapping (WB)[3] is a resampling technique that assigns different weights in dataset instances, affecting their selection during the resampling process. This technique is particularly effective in operating imbalanced datasets with more importance to classes presented.

iv)TabTransformer (TabT)[22] is a transformer-based solution that is designed for tabular datasets. It converts categorical features into embeddings and applies attention techniques to accept intricate feature relationships.

4.4 Metrics

To assess the effectiveness of our proposed technique, we rely on the following core performance metrics which had been summarized in Table 2:

i)Accuracy (Acc): It represents the ratio of correctly classified instances both true positive and true negative over the whole number of components in the dataset.

ii)Precision (Prec): It indicates the proportion of true positive predictions to all positive predictions made by the model. It shows how reliable the model is when it predicts positive class.

iii)Recall (Rec): Also referred to as sensitivity, this metric evaluate the model's ability to accurately identify all actual positive instances. It shows how to accurately evaluate the positive class.

iv)F-Measure (F1 Score): It gives a balanced average of accuracy and recall, gives the balanced evaluation of the model accuracy, making it especially effective when dealing with class imbalance.

5 Result Analysis

This section provides a comparative study of different feature weighting methods—RAW, USP, Lasso, WB, TabTransformer (TabT), and MiGRow—on the performance of a Multi-Layer Perceptron (MLP) classifier on four datasets: OS,

Table 2: Performance Indicators and Their Mathematical Formulations

Indicators	**Equations**
Accuracy (Acc)	$\frac{T^+ + T^-}{T^+ + T^- + F^+ + F^-}$
Precision (Prec)	$\frac{T^+}{T^+ + F^+}$
Recall (Rec)	$\frac{T^+}{T^+ + F^-}$
F-Measure (F1 Score)	$\frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$

MA, SD, and CE. The measures used for evaluation are Accuracy (Acc), Precision (Prec), Recall (Rec), and F1-score (F1). The acuracy comparison of techniques across OS dataset is shown in fig 2 , MA dataset in fig 3, SD dataset in fig 4 , CE dataset in fig 5 for various model.

5.1 Performance Trends Across Datasets

- **OS Dataset:** MiGRow had the best overall performance with an accuracy of 0.897 and an F1-score of 0.894, far exceeding other methods. Lasso and TabT also performed quite well but were restricted by low recall values.
- **MA Dataset:** MiGRow once more outperformed the rest with 0.874 in accuracy and 0.871 in F1-score. TabT achieved balanced recall and precision, with the remaining techniques yielding moderate gains over RAW.
- **SD Dataset:** For this dataset, most methods found it difficult. Nevertheless, MiGRow delivered the best performance in all the metrics (accuracy: 0.740), demonstrating its strength even when faced with tough situations.
- **CE Dataset:** General performance was good for all methods except one, but MiGRow still outshone them marginally with an accuracy and F1-score of 0.971. RAW and TabT also performed very well, suggesting that the CE dataset prefers easy or well-distributed feature representations.

5.2 Comparative Insights

- **MiGRow Dominance:** MiGRow consistently outperformed all other feature weighting methods across datasets and metrics, indicating its effectiveness in learning informative feature representations suitable for classification tasks.
- **Underperformance of RAW:** The RAW method (no feature weighting) generally resulted in the weakest performance, especially on more complex datasets like OS and MA, confirming the importance of proper feature weighting.
- **Variable Performance of USP and Lasso:** USP and Lasso were also dataset-dependent in their performance, at times outperforming RAW but not the consistency and strength demonstrated by MiGRow.

– **Competitiveness of TabT and WB:** TabTransformer and WB demonstrated moderate-to-good performance and closely matched MiGRow in certain instances but were less stable between datasets. The performance of Random Forest in Table 3, Logistic Regression in Table 4, Naive Bayes in Table 5, K-Nearest Neighbors in Table 6, Multi-Layer Perceptron in Table7 is shown cross all datasets and metrics

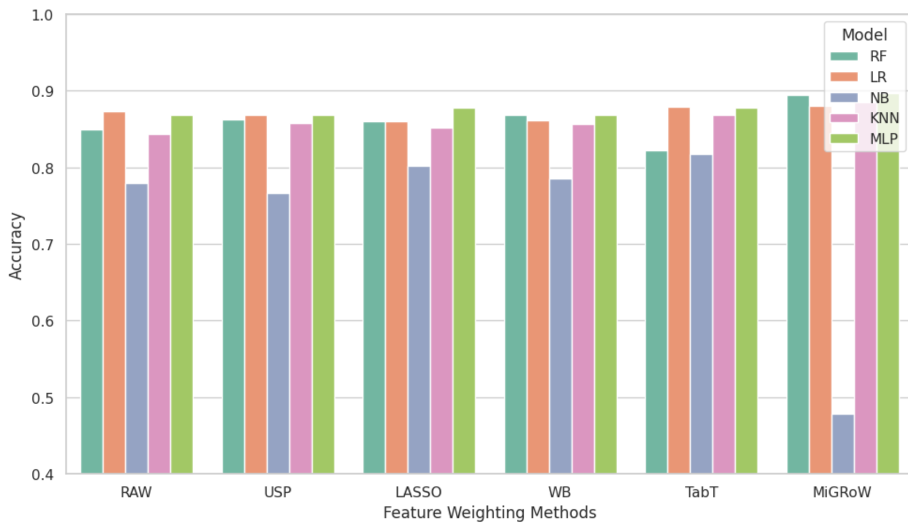

Fig. 2: Accuracy Comparison of Techniques Across OS Dataset for various models

Feature	**OS Dataset**				**MA Dataset**				**SD Dataset**				**CE Dataset**			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
RAW	0.850	0.727	0.722	0.725	0.821	0.827	0.761	0.777	0.720	0.706	0.702	0.701	0.953	0.960	0.954	0.966
USP	0.863	0.714	0.714	0.734	0.817	0.801	0.818	0.807	0.682	0.706	0.616	0.658	0.866	0.871	0.865	0.864
Lasso	0.860	0.747	0.700	0.719	0.822	0.816	0.789	0.798	0.707	0.694	0.736	0.714	0.847	0.843	0.847	0.843
WB	0.868	0.765	0.715	0.736	0.823	0.876	0.748	0.772	0.690	0.753	0.562	0.644	0.977	0.966	0.977	0.971
TabT	0.822	0.788	0.715	0.743	0.851	0.860	0.821	0.834	0.708	0.710	0.708	0.707	0.954	0.961	0.954	0.955
MiGRoW	0.873	0.787	0.724	0.843	0.808	0.674	0.852	0.750	0.868	0.812	0.716	0.988	0.770	0.869	0.819	0.720

Table 3: Performance of Random Forest (RF) across all datasets and metrics

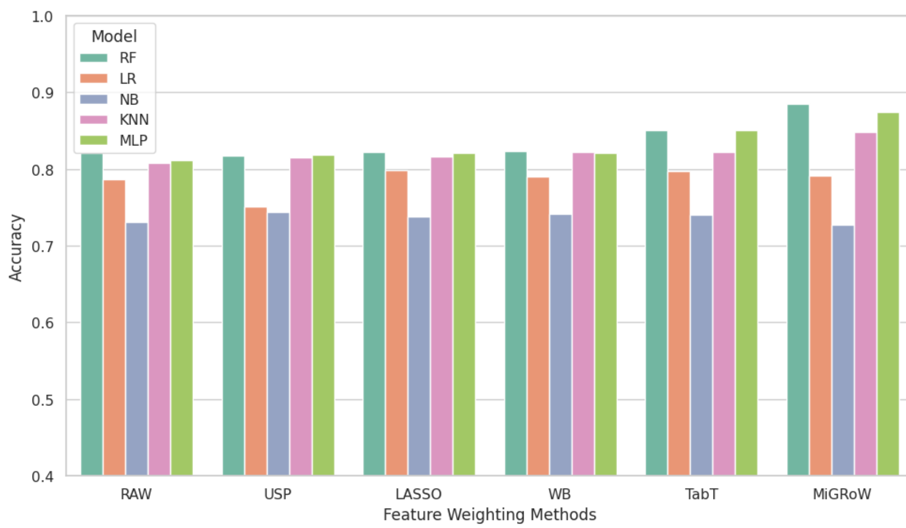

Fig. 3: Accuracy Comparison of Techniques Across MA Dataset for various models

Feature	OS Dataset				MA Dataset				SD Dataset				CE Dataset			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
RAW	0.873	0.789	0.658	0.694	0.787	0.777	0.741	0.752	0.724	0.701	0.701	0.701	0.916	0.917	0.916	0.916
USP	0.869	0.701	0.781	0.770	0.751	0.754	0.779	0.746	0.710	0.708	0.709	0.709	0.789	0.805	0.789	0.778
Lasso	0.860	0.771	0.633	0.665	0.799	0.789	0.748	0.761	0.724	0.724	0.724	0.724	0.838	0.786	0.838	0.809
WB	0.861	0.775	0.638	0.671	0.790	0.788	0.747	0.758	0.710	0.710	0.705	0.708	0.965	0.955	0.965	0.960
TabT	0.879	0.799	0.655	0.693	0.797	0.786	0.755	0.765	0.725	0.725	0.725	0.725	0.916	0.917	0.916	0.916
MiGRow	0.880	0.869	0.880	0.861	0.792	0.789	0.792	0.785	0.726	0.725	0.725	0.725	0.812	0.746	0.812	0.778

Table 4: Performance of Logistic Regression (LR) across all datasets and metrics

Feature	OS Dataset				MA Dataset				SD Dataset				CE Dataset			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
RAW	0.780	0.659	0.729	0.676	0.731	0.739	0.661	0.666	0.679	0.681	0.680	0.679	0.806	0.865	0.806	0.823
USP	0.766	0.649	0.721	0.664	0.744	0.716	0.666	0.677	0.692	0.694	0.681	0.687	0.789	0.805	0.789	0.778
Lasso	0.802	0.680	0.771	0.703	0.738	0.732	0.771	0.703	0.669	0.670	0.659	0.664	0.711	0.670	0.711	0.743
WB	0.786	0.662	0.727	0.680	0.742	0.728	0.727	0.708	0.692	0.699	0.672	0.685	0.766	0.933	0.766	0.820
TabT	0.817	0.687	0.777	0.712	0.741	0.742	0.668	0.677	0.688	0.688	0.688	0.688	0.806	0.865	0.806	0.823
MiGRoW	0.478	0.829	0.478	0.525	0.727	0.725	0.727	0.701	0.692	0.692	0.692	0.692	0.806	0.865	0.806	0.823

Table 5: Performance of Naive Bayes (NB) across all datasets and metrics

6 Discussion

The findings prove that feature weighting has a profound impact on the performance of models on tabular data. Out of all the methods, **MiGRow** out-

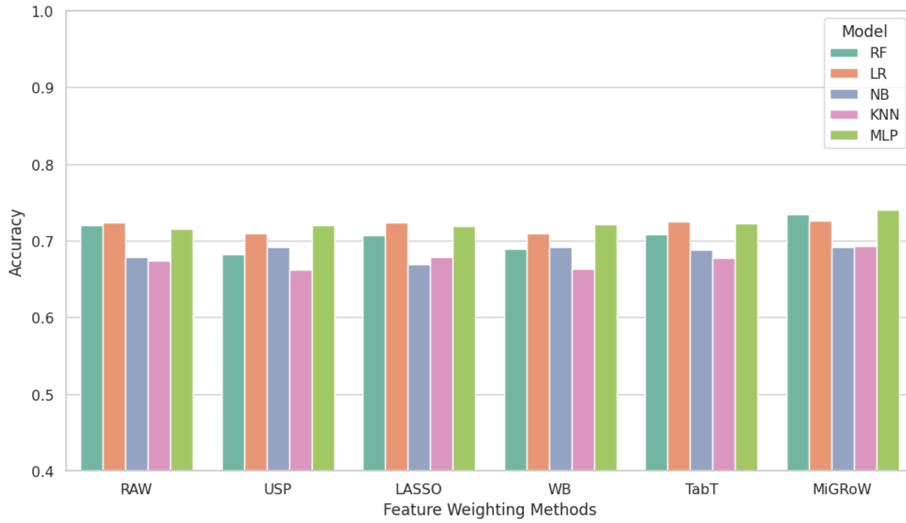

Fig. 4: Accuracy Comparison of Techniques Across SD Dataset for various models

Feature	**OS Dataset**				**MA Dataset**				**SD Dataset**				**CE Dataset**			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
RAW	0.843	0.707	0.631	0.654	0.808	0.838	0.749	0.766	0.674	0.656	0.636	0.624	0.852	0.860	0.853	0.838
USP	0.858	0.886	0.609	0.637	0.815	0.828	0.761	0.779	0.662	0.683	0.659	0.650	0.654	0.725	0.654	0.622
Lasso	0.852	0.851	0.812	0.826	0.816	0.831	0.758	0.776	0.679	0.679	0.679	0.679	0.852	0.849	0.853	0.848
WB	0.857	0.790	0.602	0.629	0.822	0.840	0.754	0.776	0.664	0.660	0.670	0.665	0.945	0.931	0.945	0.933
TabT	0.869	0.787	0.688	0.703	0.822	0.821	0.785	0.797	0.678	0.688	0.678	0.674	0.853	0.860	0.853	0.838
MiGRow	0.885	0.879	0.885	0.880	0.848	0.849	0.848	0.843	0.693	0.693	0.693	0.693	0.963	0.972	0.962	0.966

Table 6: Performance of K-Nearest Neighbors (KNN) across all datasets and metrics

Feature	**OS Dataset**				**MA Dataset**				**SD Dataset**				**CE Dataset**			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
RAW	0.868	0.788	0.662	0.697	0.812	0.822	0.772	0.785	0.716	0.716	0.716	0.716	0.988	0.990	0.989	0.989
USP	0.869	0.766	0.668	0.699	0.819	0.810	0.793	0.800	0.720	0.736	0.719	0.714	0.904	0.914	0.904	0.903
Lasso	0.878	0.806	0.700	0.735	0.821	0.820	0.780	0.793	0.719	0.725	0.718	0.717	0.873	0.860	0.873	0.865
WB	0.869	0.793	0.665	0.700	0.821	0.835	0.769	0.787	0.722	0.722	0.722	0.722	0.986	0.978	0.986	0.981
TabT	0.878	0.821	0.635	0.673	0.851	0.840	0.839	0.839	0.723	0.723	0.723	0.720	0.988	0.990	0.988	0.989
MiGRow	0.897	0.892	0.897	0.894	0.874	0.875	0.874	0.871	0.740	0.740	0.740	0.740	0.971	0.972	0.971	0.971

Table 7: Performance of Multi-Layer Perceptron (MLP) across all datasets and metrics

performed others on all datasets (OS, MA, SD, and CE) with better accuracy,

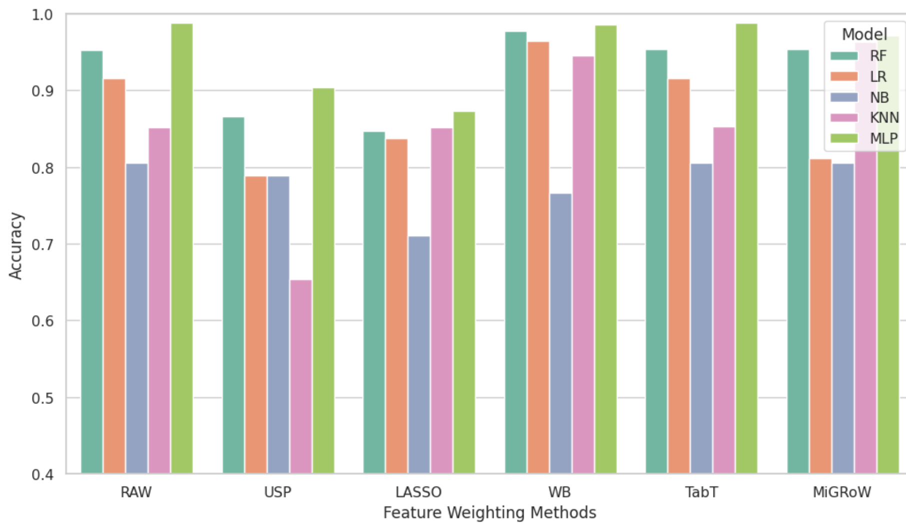

Fig. 5: Accuracy Comparison of Techniques Across CE Dataset for various model

precision, recall, and F1-score. This reflects its capability of learning important feature importance accurately and enhancing generalization.

Conversely, classical methods such as RAW and USP experienced erratic performance with limited ability to adapt. Lasso, WB, and TabTransformer outperformed RAW but made their improvements on inconsistent bases across datasets.

MiGRow was the most stable feature weighting method overall, with robust and stable performance across different data conditions.

7 Conclusion and Future Work

The paper provided a comparison analysis of some feature weighting methods on four benchmark tabular datasets. From the experimental results, it shows that **MiGRow** outperforms other approaches with respect to classification performance in all cases, which confirms its effectiveness in extracting relevant and discriminative features.

In our future work, we intend to generalize MiGRow to process high-dimensional and imbalanced datasets, measure its computational cost, and extend it to combine with other learning paradigms like ensemble models. Further investigating its applicability in real-time and streaming data settings is also an attractive direction.

References

1. Abid-Althaqafi, N.R., Alsalamah, H.A., Ismail, W.N.: The impact of the weighted features on the accuracy of x-platform's user credibility detection using supervised

machine learning. IEEE Access **12**, 8471–8484 (2024)
2. Bohanec, M.: Car evaluation. uci machine learning repository (1997). DOI: https://doi. org/10.24432/C5JP48 (2023)
3. Breiman, L.: Bagging predictors. Machine learning **24**, 123–140 (1996)
4. Breiman, L.: Random forests. Machine learning **45**, 5–32 (2001)
5. Chen, M., Gong, M., Li, X.: Feature weighted non-negative matrix factorization. IEEE Transactions on Cybernetics **53**(2), 1093–1105 (2021)
6. Cover, T.M.: Elements of information theory. John Wiley & Sons (1999)
7. Das, M., Alphonse, P., et al.: A comparative study on tf-idf feature weighting method and its analysis using unstructured dataset. arXiv preprint arXiv:2308.04037 (2023)
8. Dua, D., Graff, C., et al.: Uci machine learning repository (2017)
9. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Transactions on knowledge and data engineering **21**(9), 1263–1284 (2009)
10. Huang, X., Khetan, A., Cvitkovic, M., Karnin, Z.: Tabtransformer: Tabular data modeling using contextual embeddings. arXiv preprint arXiv:2012.06678 (2020)
11. Hussain, I., Nataliani, Y., Ali, M., Hussain, A., Mujlid, H.M., Almaliki, F.A., Rahimi, N.M.: Weighted multiview k-means clustering with l2 regularization. Symmetry **16**(12), 1646 (2024)
12. Jain, A.K.: Fundamentals of digital image processing. Prentice-Hall, Inc. (1989)
13. Karthick, K., Agnes, S.A., Kumar, S.S., Alfarhood, S., Safran, M.: Identification of high energy gamma particles from the cherenkov gamma telescope data using a deep learning approach. IEEE Access **12**, 16741–16752 (2024)
14. Korean National Health Insurance Service: Smoking and drinking dataset with body signal. https://www.kaggle.com/datasets/yasserhesham/smoking-and-drinking-dataset-with-body-signal (2022), accessed: 2025-05-17
15. Li, Z., Wu, Y., Chen, K., Wu, Y., Zhou, S., Liu, J., Yan, J.: Learning to auto weight: Entirely data-driven and highly efficient weighting framework. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 4788–4795 (2020)
16. MacQueen, J.: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics. vol. 5, pp. 281–298. University of California press (1967)
17. Maraş, A., Erol, Ç.: Emerging trends in classification with imbalanced datasets: A bibliometric analysis of progression. Bilişim Teknolojileri Dergisi **15**(3), 275–288 (2022)
18. Masramon, G.P., Muñoz, L.A.B.: Toward better feature weighting algorithms: a focus on relief. arXiv preprint arXiv:1509.03755 (2015)
19. Panday, D., De Amorim, R.C., Lane, P.: Feature weighting as a tool for unsupervised feature selection. Information processing letters **129**, 44–52 (2018)
20. Shannon, C.E.: A mathematical theory of communication. The Bell system technical journal **27**(3), 379–423 (1948)
21. Sun, Y., Todorovic, S., Goodison, S.: Local-learning-based feature selection for high-dimensional data analysis. IEEE transactions on pattern analysis and machine intelligence **32**(9), 1610–1626 (2009)
22. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society Series B: Statistical Methodology **58**(1), 267–288 (1996)
23. Xu, Y., Fu, X., Li, H., Dong, G., Wang, Q.: A k-means algorithm based on feature weighting. In: MATEC Web of Conferences. vol. 232, p. 03005. EDP Sciences (2018)
24. Zhang, X., Wang, Z., Jiang, L., Gao, W., Wang, P., Liu, K.: Tfwt: Tabular feature weighting with transformer. arXiv preprint arXiv:2405.08403 (2024)