TCSA
https://dergipark.org.tr/tr/pub/tcsa
Vol.1 No.2
May 2024: 1-3
Accepted 12 Dec 2024
Published 20 Dec 2024

Transactions on Computer Science and Applications

ISSN(E): 3023-8129

Histogram-Based Feature Selection for Binary Classification

Selman Delil[*,a], Melih AĞRAZ[b], Birol Kuyumcu[c]
[a] Istanbul Technical University, Istanbul, Türkiye, 34349, ORCID: https://orcid.org/0000-0001-8149-3561
[b] Giresun University, Giresun, Türkiye, 28200, ORCID: https://orcid.org/0000-0002-6597-7627
[c] Openzeka, Ankara, Türkiye, 06530, ORCID: https://orcid.org/0000-0002-6366-6198

*** Corresponding author email address**: selmandelil@gmail.com, **phone:** +90 5538777634

Abstract

This paper presents a novel method for feature selection in binary classification tasks based on histogram-based scoring. By leveraging the distribution differences between feature values associated with positive and negative classes, we generate a score to determine the most informative features. The method, called Histogram-Based Feature Selection (HBFS) has been tested against a variety of datasets and compared to the Fisher Score for performance assessment. Our findings indicate that HBFS either matches or outperforms Fisher Score in most datasets.

Keywords: Machine Learning, Feature Selection, Histogram-Based Feature Selection, Fisher Score

1. Introduction

1.1 Background

Feature selection is an essential preprocessing step in machine learning, particularly in high-dimensional data scenarios. High-dimensional datasets, characterized by a large number of features relative to the number of samples, pose significant challenges, including overfitting, computational inefficiency, and reduced model interpretability (Li et al., 2023). To address these issues, feature selection aims to identify and retain only the most informative features, improving model accuracy, and efficiency.

Among the many methods developed for feature selection, the Fisher Score (Duda et al., 2001) remains one of the most widely used due to its simplicity, efficiency, and strong theoretical foundation. It evaluates the discriminative power of individual features by analyzing the ratio of inter-class variance to intra-class variance. This univariate approach is computationally efficient, making it particularly suitable for high-dimensional datasets in fields such as bioinformatics, text classification, and image recognition (Abiodun et al., 2021). However, Fisher Score operates under the assumption that feature contributions are independent and linear, which can limit its effectiveness in capturing more complex relationships in modern datasets (Gan & Zhang., 2021).

This study introduces a Histogram-Based Feature Selection (HBFS) method that builds upon the principles of distribution-based feature selection. Unlike the Fisher Score, which focuses on variance-based separability, HBFS quantifies the differences in feature value distributions across class labels using histogram comparisons. By addressing the limitations of traditional methods, the proposed HBFS method aims to provide a robust and scalable solution for binary classification tasks. The effectiveness of HBFS is evaluated by directly comparing its performance to the Fisher Score, leveraging the latter's well-established baseline status to highlight the advantages of the proposed method.

1.2 Related Works

Feature selection has been extensively studied, with methods broadly categorized into filter, wrapper, and embedded approaches (He et al., 2005). Among these, filter-based methods like the Fisher Score remain popular due to their computational efficiency and statistical scalability (Guyon & Elisseeff, 2003). The Fisher Score is particularly effective in assessing the relevance of individual features in high-dimensional datasets, as demonstrated in applications ranging from gene expression analysis to handwritten digit recognition. Despite its widespread use, its inability to capture interactions between features or account for complex data distributions has motivated the development of alternative approaches (Gan & Zhang., 2021).

Advancements in feature selection have progressively leveraged distributional properties to capture complex patterns in high-dimensional spaces. Jagdhuber et al. (2020) demonstrated the benefits of cost-constrained feature selection using genetic algorithms, highlighting the potential of incorporating constraints to optimize performance.

Recently, Khan et al. (2024) introduced a weighted scoring method tailored for imbalanced datasets, further advancing the field. Additionally, they explored histogram-based approaches, using normalized histograms to enhance robustness and stability in feature selection. These techniques collectively underscore the growing importance of distribution-based scoring in binary classification tasks.

Expanding on these advancements, this study introduces a novel Histogram-Based Feature Selection (HBFS) method for binary classification. By leveraging differences in the

distributions of continuous features across class labels, the proposed HBFS method aims to address the limitations of traditional techniques. To further enhance its effectiveness, a refinement step inspired by the Minimum Redundancy Maximum Relevance (MRMR) method (Peng et al., 2005) is incorporated to reduce redundancy among selected features.

2. Methodology

2.1 Fisher Score

The Fisher Score is a widely used similarity-based feature selection method designed to evaluate the discriminative power of individual features in classification tasks. It measures the ratio of inter-class variance to intra-class variance for each feature, making it particularly effective in identifying features that contribute to class separability (Duda et al., 2001). This method has been widely applied in domains such as gene expression analysis, image recognition, and text classification, where distinguishing relevant features from irrelevant ones is critical.

Mathematical Definition

Given a dataset with *n* samples and *d* features, let $X = [x_1, x_2, \ldots, x_n]$ represent the samples, and let $y \in \{1,2,\ldots,C\}$ denote the class labels. The Fisher Score for a feature *j* is computed as:

$$\text{Fisher Score}\,(j) = \frac{\sum_{c=1}^{C} N_c \left(\mu_j^{(c)} - \mu_j\right)^2}{\sum_{c=1}^{C} N_c \sigma_j^{(c)^2}}$$

Where:

N_c is the number of samples in class c,
$\mu_j^{(c)}$ is the mean of feature j for class c,
μ_j is the mean of feature j across all samples,
$\sigma_j^{(c)^2}$ is the variance of feature j for class c.

The Fisher Score quantifies how well a feature separates samples from different classes while minimizing variation within the same class.

Key Properties

1. **Class Separability**: Features with higher Fisher Scores indicate better discriminative capability as they maximize inter-class variance while minimizing intra-class variance.
2. **Univariate Method**: The Fisher Score evaluates each feature independently, without considering feature interactions or correlations.
3. **Efficiency**: As a similarity-based method, it is computationally efficient, making it suitable for high-dimensional datasets.

2.2 Histogram-Based Feature Selection (HBFS)

For each feature, HBFS generates two histograms: one for instances with positive class labels (Y=1) and the other for instances with negative class labels (Y=0). The histograms are normalized so that the total area sums to 1, ensuring a consistent basis for comparison. The bins are set to 100 to provide sufficient granularity in representation. If the histograms are identical, all differences will be 0, resulting in a score of 0. As the overlap between the histograms decreases and the distributions become more distinct, the sum of absolute differences increases, ultimately reaching a maximum value of 2.

The absolute difference between these histograms is calculated and summed to generate a feature score $S(fi)$ defined as:

$$S(fi) = \sum_{j=1}^{100} | Hpos(j) - Hneg(j) |$$

where $Hpos(j)$ and $Hneg(j)$ are the normalized histogram values for positive and negative classes at bin *j*, respectively. If two distributions overlap entirely, the score will be 0; if they are completely distinct, the score will reach a maximum of 2. Features are then ranked by their scores, allowing for selection based on a chosen threshold. For all features, the HBFS scores are calculated and ranked in descending order to select features based on their scores.

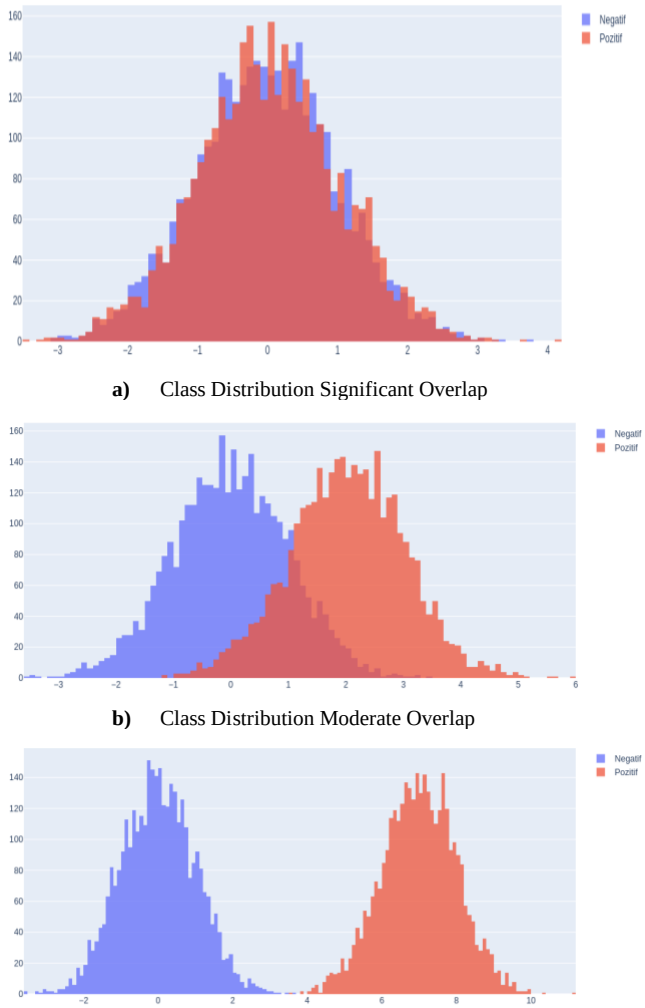
Input: Dataset D with features F={f_1,f_2,...,f_n} and class labels Y={0,1}
Output: Ranked list of features based on HBFS score

1. For each feature f_i in *F*:
   - Extract values for class *Y*=1 and *Y*=0 separately.
   - Create histograms for both classes with bin size = 100.
   - Normalize histograms such that the total area equals 1.
   - Calculate the absolute difference between the two histograms.
   - Sum the differences to obtain the HBFS score for feature f_i.
2. Rank all features based on their HBFS scores in descending order.
3. Return the ranked list of features.

The graphical representations included in Fig. 1 illustrate the distribution of feature values for both positive (Y = 1) and negative (Y = 0) classes, as well as the resulting in histograms used to compute the HBFS scores. These graphics provide a visual demonstration of how feature distributions differ across classes, and the impact of histogram normalization. By examining these histograms, it is easier to understand the degree of overlap and the distinctiveness of each feature, which directly influences the calculated HBFS scores.

The first histogram (Fig. 1-a) shows the distribution where the positive and negative class distributions significantly overlap, highlighting the challenge of distinguishing between these classes. The second histogram (Fig. 1-b) represents a scenario with moderate overlap, whereas the third histogram (Fig. 1-c) shows a clear

separation between the distributions of positive and negative classes. These visuals demonstrate the varying discriminative power of different features, which the HBFS method uses to calculate the feature importance scores.



a) Class Distribution Significant Overlap

b) Class Distribution Moderate Overlap

c) Class Distribution Clear Separation

Fig. 1. HBFS Score Calculation as Class Distribution

2.3 HBFS Refined Approach

The HBFS method was extended to consider feature-specific ranges. Instead of applying a fixed histogram range, the minimum and maximum values of each feature were utilized as bounds. The histogram granularity (number of bins) is adjustable as a hyperparameter (e.g., alpha = 100). The scoring process remains consistent, with features initially ranked in descending order based on their scores.

To further refine the feature selection process and reduce redundancy, a two-step approach was implemented. First, the top 10 features with the highest HBFS scores were selected. Next, from the remaining pool, 240 additional features were chosen based on their minimal correlation with the highest-scoring feature. This approach ensured a final subset of 250 features, balancing discriminative power and diversity. To address potential redundancy among selected features, a refinement step inspired by the **Minimal Redundancy Maximal Relevance (mRMR)** criterion (Peng et al., 2005) was introduced. This refinement involves calculating correlations among selected features using:

$$Corr(f_i, f_j) = \frac{Cov(f_i, f_j)}{\sigma_{\{f_i\}}\,\sigma_{\{f_j\}}}$$

where $Corr(f_i, f_j)$ is the covariance between features f_i and f_j, while $\sigma_{\{f_i\}}$ and $\sigma_{\{f_j\}}$ are their standard deviations. This refinement aims to reduce redundancy and select features that are both highly informative and minimally correlated.

The refined framework aims to select features that not only demonstrate high relevance to the target classification variable but also exhibit minimal redundancy among themselves. By integrating this step, the selection process effectively enhances the quality of the feature subset, ensuring that each feature contributes unique information while maintaining strong predictive power for the classification task. This dual focus on relevance and redundancy is essential for achieving optimal model performance in high-dimensional data environments.

3. Experimental Setup

3.1. Dataset

In this study, we utilize a collection of widely recognized datasets (**Table 1**) to evaluate the proposed method. These datasets, commonly used in feature selection and classification research, are characterized by a high number of features relative to the number of samples. This property, known as high-dimensional, low-sample-size data, poses significant challenges in machine learning, particularly in overfitting and computational efficiency. These datasets cover diverse domains, including high-dimensional and low-sample-size gene expression datasets, as well as synthetic and real-world data with challenging classification problems. The selected datasets (*Datasets: Feature selection, n.d.*) are as follows:

1. **ALLAML (**Davide, 2019**)**: This gene expression dataset comprises 72 instances with 7,129 features. It involves a binary classification task distinguishing acute lymphoblastic leukemia (ALL) from acute myeloid leukemia (AML). As a high-dimensional, low-sample-size dataset, it reflects typical challenges in biomedical research.

2. **GLI_85** (GEO Accession viewer, n.d.; Freije et al., 2004.)**:** Consisting of 85 instances and 22,283 features, this dataset includes glioma gene expression data. It is used to classify glioblastomas versus normal samples, making it a benchmark for evaluating feature selection in biological data.

3. **SMK_CAN_187**(Spira et al., 2007; Gustafson et al., 2010) : This dataset features 187 samples and 19,993 gene expression features. It is another high-dimensional dataset that poses significant challenges for

dimensionality reduction and classification. There are two classes – 123 cancer samples and 64 normal tissue samples.

4. **Arcene** (Datasets: Feature selection, n.d.; Wayback machine, n.d.;)**:** Derived from mass spectrometry data, this dataset includes 200 instances and 10,000 features. It is designed for binary classification tasks and is characterized by noise and high dimensionality, making it suitable for evaluating feature selection methods. Two classes – evenly distributed (100 instances per class).

5. **Madelon** (Datasets: Feature selection, n.d.; Wayback machine, n.d.;)**:** A synthetic dataset with 2,600 instances and 500 features. This dataset is explicitly constructed for binary classification tasks, where relevant features are intentionally masked by irrelevant ones, making feature selection critical. Two balanced classes – 1,300 instances per class.

6. **Gisette** (Datasets: Feature selection, n.d.; Wayback machine, n.d.;)**:** With 7,000 instances and 5,000 features, this dataset is derived from handwritten digit recognition tasks. It involves distinguishing between digits "4" and "9," showcasing real-world classification challenges. Two classes – 3,500 instances per class.

7. **Prostate_GE** (Datasets: Feature selection, n.d.)**:** This dataset includes 102 instances and 5,966 gene expression features. It is used for classifying prostate cancer samples versus normal tissue samples, highlighting challenges in processing biological data. Two classes – 52 prostate cancer samples and 50 normal tissue samples.

Table 1
Dataset summary.

No	Dataset	# of Sample	# of Features	Type
1	ALLAML	72	7129	Biological Data
2	GLI_85	85	22283	Biological Data
3	SMK_CAN_187	187	19993	Biological Data
4	arcene	200	10000	Mass Spectrometry
5	madelon	2600	500	Artificial
6	gisette	7000	5000	Digit Recognition
7	Prostate_GE	102	5966	Biological Data

3.2 Feature selection

Features are selected in incremental steps based on their rankings using Fisher Score and HBFS. Feature vector sizes vary from 10 to 250 in increments of 10. This step evaluates the impact of the number of selected features on model performance.

We employ the Fisher Score to rank and select features based on their relevance to classification tasks. This allows us to reduce the dimensionality of datasets while preserving the most informative features. The implementation of the Fisher Score used in this study is based on the scikit-feature package, a feature selection library built on the design principles of the scikit-learn project (Buitinck et al., 2013).

The proposed HBFS method was tested against the Fisher Score method using seven datasets: ALLAML, GLI_85, SMK_CAN_187, Arcene, Madelon, Gisette, and Prostate_GE. Each dataset contains high-dimensional features with binary class labels.

The training process is conducted incrementally to analyze the impact of the number of selected features on model performance:

1. **Feature Subset Selection**: Features are incrementally added in subsets of size 10, starting from the top 10 ranked features and increasing to 250 features.
2. **Model Training**: For each subset, the Extra Trees Regressor is trained on the selected features from the training data.
3. **Evaluation**: The model's performance is evaluated on the test data using the Weighted F1 Score, which balances precision and recall while considering class imbalance.

3.3. Model and Training

The machine learning model used in this study is the **Extra Trees Regressor** (Extremely Randomized Trees Regressor), which is an ensemble learning method (Geurts et al., 2006) designed to improve predictive performance and control overfitting. It builds multiple regression trees by introducing randomness during tree construction, such as selecting random split thresholds for features. This approach enhances generalization, particularly in high-dimensional datasets, making it well-suited for the feature selection tasks in this study.

Motivation for Extra Trees Regressor

The Extra Trees Regressor is particularly suitable for this study due to its:

- **Ability to Handle High-Dimensional Data**: The randomization in tree construction reduces overfitting, making it effective in high-dimensional, low-sample-size datasets.
- **Feature Importance Assessment**: The method provides insights into the importance of features, aligning with the study's focus on feature selection.

This systematic training and evaluation approach ensures a robust assessment of the feature selection methods and their impact on model performance.

Model Configuration

The Extra Trees Regressor is configured with the following parameters:

- **Number of Estimators**: 70 trees are built in the

ensemble.

- **Random State**: A fixed value of 123 ensures reproducibility of results across multiple runs.

Data Splitting

Each dataset is divided into training and test subsets:

- **Training Set**: 80% of the data is used for training the model.
- **Test Set**: 20% of the data is reserved for evaluating the model's performance. The split is performed using the train_test_split function from scikit-learn with a fixed random state to ensure consistency.

3.4. Evaluation

The model is trained on the selected features incrementally, starting from 10 features and increasing by 10 up to 250 features. For each subset of features, the following steps are performed:

1. Train the model on the training set using the specified number of features.
2. Evaluate the model on the test set.
3. Record the **Weighted F1 Score** for each feature subset.

Both methods were evaluated by selecting features based on ranking and assessing classification performance using F1-score:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall}$$

To validate the effectiveness of the proposed feature selection system, classification performances of features selected using HBFS were compared with those selected using Fisher Score.

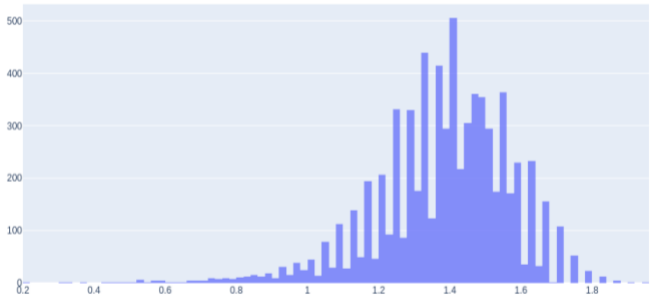
4. Results and Discussion

4.1 HBFS Score Distributions

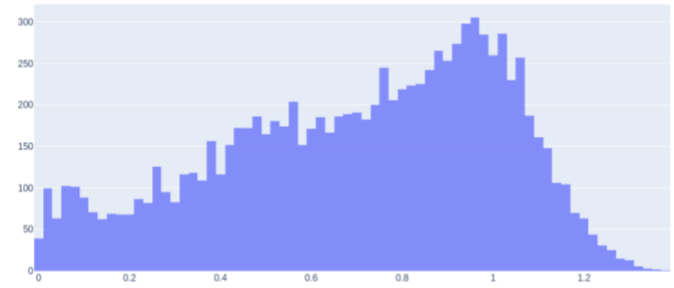
The HBFS score distributions for the datasets are presented in **Fig. 2 (a)** to **(g)**. These histograms provide a visual overview of how the feature scores are distributed across different datasets, offering insights into the discriminative power of features. In general, features with higher HBFS scores indicate better separation between the positive and negative classes, while lower scores suggest significant overlap between class distributions.

Most datasets exhibit a skewed or asymmetric distribution, where the majority of features tend to have relatively low scores, while only a smaller subset achieves higher discriminative values. This pattern underscores that while many features contribute marginally to class differentiation, a select few have significant impact, guiding the need for effective feature selection strategies. The variability in these distributions suggests that different datasets may require customized thresholds for feature selection, depending on the shape and spread of their score distributions.

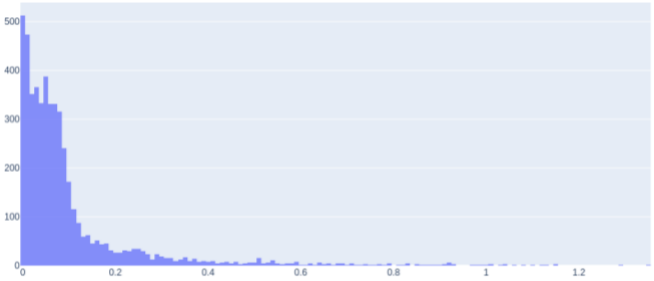
The histograms show that some datasets, such as **GLI_85 (Fig. 2-d)** and **ALLAML** (Fig. 2-a), exhibit a broader range of moderately high scores, indicating a higher number of informative features. In contrast, datasets like **Gisette (Fig. 2-c)** and **Madelon (Fig. 2-e)** display a concentration of features with lower scores, implying a more challenging feature selection process with fewer clearly discriminative features. These visual patterns highlight the importance of tailoring feature selection approaches to the specific characteristics of each dataset to optimize classification performance.



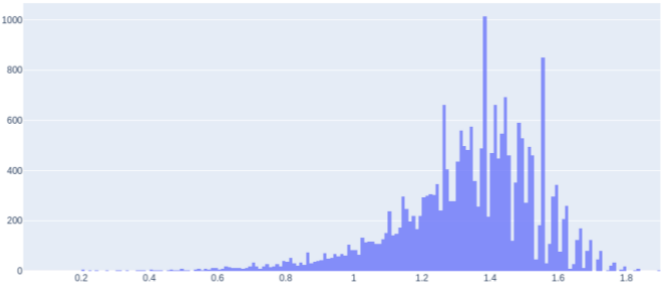
a) HBFS Score Distribution for ALLAML Dataset



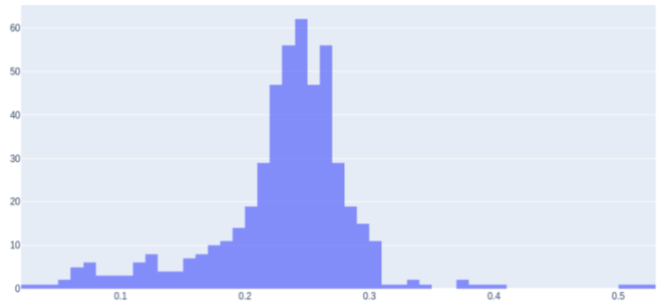
b) HBFS Score Distribution for Arcene Dataset



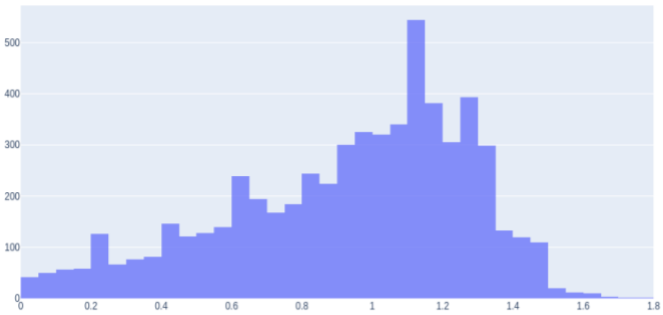
c) HBFS Score Distribution for Gisette Dataset



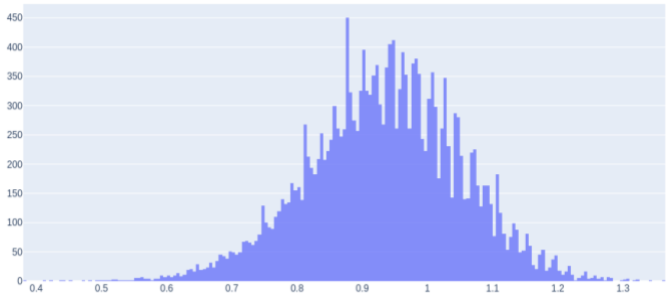
d) HBFS Score Distribution for GLI_85 Dataset



e) HBFS Score Distribution for Madelon Dataset



f) HBFS Score Distribution for Prostate_GE Dataset



g) HBFS Score Distribution for SMK_CAN_187 Dataset

Fig. 2. HBFS Score Distributions

4.2 Comparison with Fisher Score

The results summarized in **Table 2** illustrate the performance of the Histogram-Based Feature Selection (HBFS) method compared to the Fisher Score across seven diverse datasets. Notably, HBFS outperformed Fisher Score in four datasets and matched its performance in the remaining three. The refinement process applied to HBFS (termed HBFS Refine) further enhanced the results, achieving the highest F1-scores in six out of seven datasets.

Table 2
Results of the HBFS vs Fisher Score Comparison.

No	Dataset	Fisher Score Feature size	Fisher Score Best F1	HBFS Score Feature size	HBFS Score Best F1	HBFS Score Feature size	HBFS Score Refine Best F1
1	ALLAML	20	**1**	20	**1**	20	**1**
2	Arcene	130	0.95	170	**0.975**	210	**1**
3	Gisette	250	0.908	160	**0.971**	220	**0.973**
4	GLI_85	30	**1**	10	**1**	10	**1**
5	Madelon	180	0.842	160	**0.883**	10	0.863
6	Prostate_GE	230	0.905	10	**0.952**	50	**1**
7	SMK_CAN_187	40	**0.87**	150	**0.87**	150	**0.87**

When comparing HBFS with Fisher scores, HBFS demonstrates superior performance in 4 out of 7 datasets, with the remaining 3 achieving the same high accuracy. This indicates that HBFS not only matches but often exceeds Fisher in feature selection efficiency, providing both improved F1 scores and more compact feature sets in several cases.

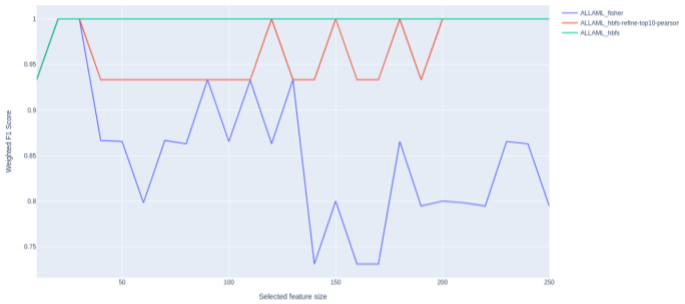
4.3 Feature Selection Performance

The HBFS method demonstrated consistent performance across the datasets, achieving notably higher F1-scores compared to the Fisher Score for datasets like Arcene (0.975 vs. 0.950), Gisette (0.971 vs. 0.908), and Prostate_GE (0.952 vs. 0.905) as shown in **Table 2**. When applying HBFS Refine, additional improvement was observed in datasets such as Prostate_GE, which reached an F1-score of 1.000 with refined feature selection compared to 0.952 without refinement.

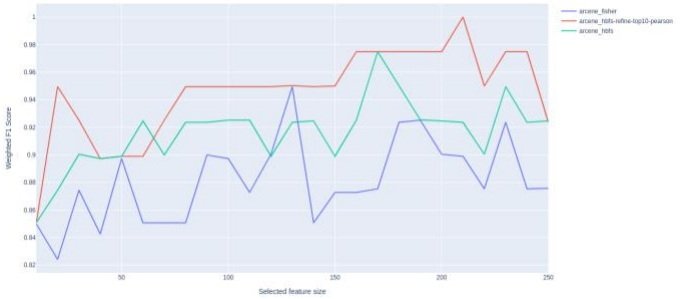
The Fisher score, while providing accurate results, did not achieve the best outcome across any dataset on its own. Among the 3 datasets where Fisher and HBFS produced equal performance, only one dataset required fewer features to achieve the same result. The HBFS method outperformed Fisher in 4 out of 7 datasets, with the remaining 3 achieving equal results.

Finally, HBFS Refine stands out as the most effective method, delivering the best results in 6 out of 7 datasets, surpassing both Fisher and standard HBFS in almost every case. This demonstrates that the refined HBFS method not only maintains high performance but also optimizes feature selection for better efficiency.

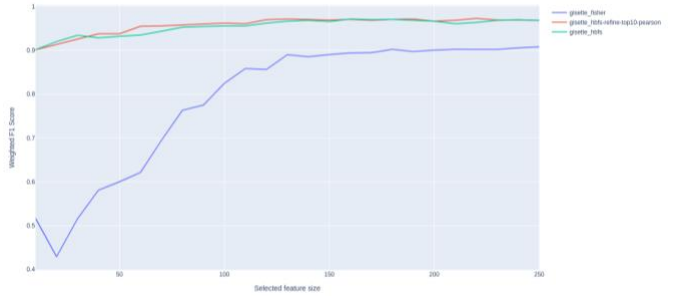
The experimental results illustrate the comparative performance of HBFS, Refined HBFS, and Fisher methods across multiple datasets, with F1 score distributions provided for each dataset to highlight performance variability and trends across the three methods (**Fig. 3, subgraphs (a) to (g)**). In the graphs, HBFS is represented iny **green**, Refined HBFS by **red**, and Fisher by **blue**, allowing for a clear visual distinction between the methods and their respective performance trajectories.



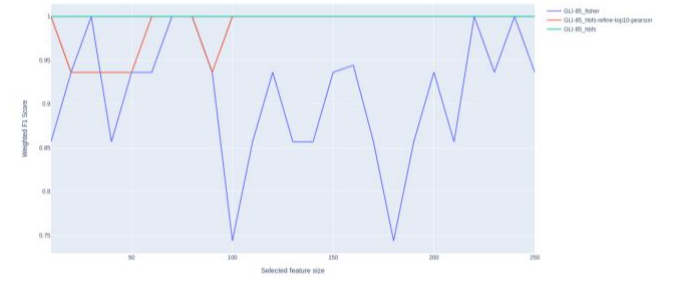
a) Weighted F1 Scores /ALLAML



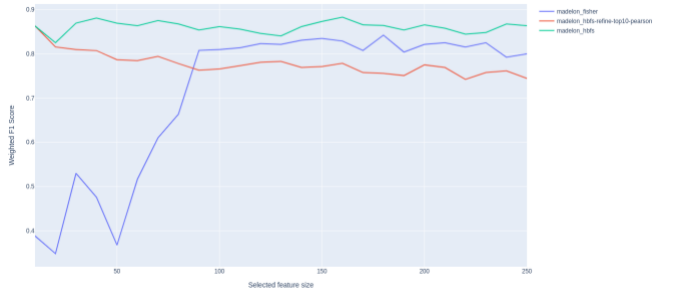
b) Weighted F1 Scores /Arcene



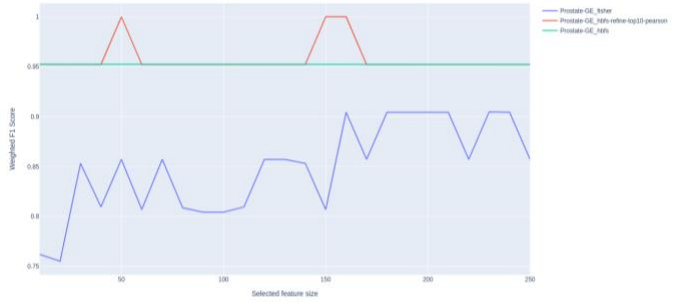
c) Weighted F1 Scores /Gisette



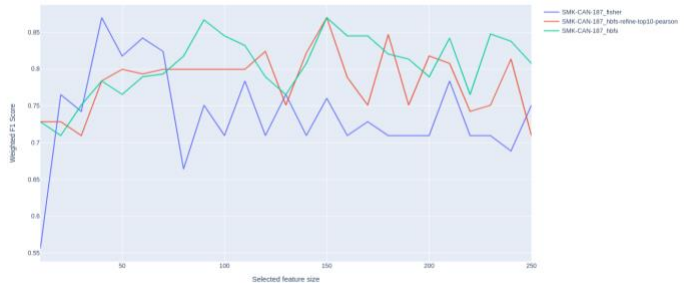
d) Weighted F1 Scores /GLI_85



e) Weighted F1 Scores /Madelon



f) Weighted F1 Scores /Prostate_GE



g) Weighted F1 Scores /SMK_CAN_187

Fig. 3. F1 Distributions: HBFS (green) / Refined HBFS (red) / Fisher (blue)

Fig. 3-a (ALLAML) demonstrates that both HBFS and Refined HBFS maintain consistently high Weighted F1 Scores, nearing 1.0 across the entire feature size spectrum. This stability underscores the robustness of these methods in datasets where feature discriminative power is uniformly distributed. In contrast, Fisher exhibits substantial fluctuations, particularly as the feature size increases, indicating its sensitivity to the selection of features and possible susceptibility to noise.

A similar trend is observed in **Fig. 3-b (Arcene)**, where Refined HBFS outperforms the other methods, especially at smaller feature sizes. This suggests that refining the feature selection process improves the identification of the most discriminative features in datasets with high dimensionality.

HBFS shows steady performance, while Fisher's variability persists, reinforcing its dependency on optimal feature set selection. Interestingly, **Fig. 3-c (Gisette)** highlights a convergence of all three methods as feature size increases, suggesting that when more features are included, the distinctions between these approaches become less significant. However, in the early stages, Refined HBFS demonstrates a clear advantage in achieving higher F1 Scores with fewer features.

5. Conclusion

This paper introduced a novel Histogram-Based Feature Selection (HBFS) method, which leverages histogram-based scoring to differentiate feature distributions across classes. The results demonstrated that the proposed HBFS method effectively identifies features that enhance classification accuracy, outperforming the traditional Fisher Score in multiple datasets. Furthermore, the inclusion of a redundancy reduction step significantly improved the method's capability, making it a robust and scalable tool for feature selection in high-dimensional classification tasks.

The findings emphasize the value of distribution-based approaches in feature selection, addressing the limitations of variance-based methods like the Fisher Score. Additionally,

the proposed refinement process highlighted the importance of balancing feature relevance and redundancy to achieve optimal classification performance.

Future research will aim to extend the HBFS method to multi-class classification scenarios, a critical step for broader applicability. Furthermore, incorporating adaptive binning strategies will be explored to enhance the robustness and flexibility of the method, particularly for datasets with complex or imbalanced distributions.

References

Li, K., Wang, F., Yang, L., & Liu, R. (2023). Deep feature screening: Feature selection for ultra high-dimensional data via deep neural networks. Neurocomputing, 538, 126186.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). Pattern Classification, Hoboken. In: NJ: Wiley.

Abiodun, E. O., Alabdulatif, A., Abiodun, O. I., Alawida, M., Alabdulatif, A., & Alkhawaldeh, R. S. (2021). A systematic review of emerging feature selection optimization methods for optimal text classification: the present state and prospective opportunities. Neural Computing and Applications, 33(22), 15091-15118.

Gan, M., & Zhang, L. (2021). Iteratively local fisher score for feature selection. *Applied Intelligence*, *51*, 6167-6181.

He, X., Cai, D., & Niyogi, P. (2005). Laplacian score for feature selection. Advances in neural information processing systems, 18.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Khan, Z., Ali, A., & Aldahmani, S. (2024). Feature Selection via Robust Weighted Score for High Dimensional Binary Class-Imbalanced Gene Expression Data. *arXiv preprint arXiv:2401.12667*.

Jagdhuber, R., Lang, M., Stenzl, A., Neuhaus, J., & Rahnenführer, J. (2020). Cost-Constrained feature selection in binary classification: adaptations for greedy forward selection and genetic algorithms. *BMC bioinformatics*, *21*, 1-21.

Peng, H., Long, F., & Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on pattern analysis and machine intelligence, 27(8), 1226-1238.

Datasets: Feature selection . (n.d.). Retrieved from https://jundongl.github.io/scikit-feature/datasets.html

Davide Nardone. (2019). Biological datasets for SMBA. https://doi.org/10.5281/zenodo.2709491

GEO Accession viewer. (n.d.). https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE441 2

Freije, W. A., Castro-Vargas, F. E., Fang, Z., Horvath, S., Cloughesy, T., Liau, L. M., Mischel, P. S., & Nelson, S. F. (2004). Gene expression profiling of gliomas strongly predicts survival. *Cancer research*, *64*(18), 6503–6510. https://doi.org/10.1158/0008-5472.CAN-04-0452

Spira, A., Beane, J. E., Shah, V., Steiling, K., Liu, G., Schembri, F.,Gilman, S., Dumas, Y. M., Calner, P., Sebastiani, P., Sridhar, S., Beamis, J., Lamb, C., Anderson, T., Gerry, N., Keane, J., Lenburg, M. E., & Brody, J. S. (2007). Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. Nature medicine, 13(3), 361–366. https://doi.org/10.1038/nm1556

Gustafson, A. M., Soldi, R., Anderlind, C., Scholand, M. B., Qian, J., Zhang, X., Cooper, K., Walker, D., McWilliams, A., Liu, G., Szabo, E., Brody, J., Massion, P. P., Lenburg, M. E., Lam, S., Bild, A. H., & Spira, A. (2010). Airway PI3K pathway activation is an early and reversible event in lung cancer development. *Science translational medicine*, *2*(26), 26ra25. https://doi.org/10.1126/scitranslmed.3000251

Wayback machine. (n.d.). https://web.archive.org/web/20150221003104/http://www.nipsfsc.ecs.soton.ac.uk/papers/NIPS2003-Datasets.pdf

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., ... & Varoquaux, G. (2013). API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine learning, 63, 3-42.