International Journal of Statistical Sciences *ISSN 1683-5603*
Vol. 25(2), November, 2025, pp 159-172
DOI: https://doi.org/10.3329/ijss.v25i2.85778
© 2025 Dept. of Statistics, Univ. of Rajshahi, Bangladesh

Star Classification Using Machine Learning: A Comparative Analysis of Random Forest and LightGBM on SDSS Data

Yasir Arafat[1], Rasna Begum[1], Md. Saifur Rahman[1] and Md. Kaderi Kibria[1*]

[1]Department of Statistics, Hajee Mohammad Danesh Science and Technology University, Dinajpur-5200, Bangladesh

[*]Correspondence should be addressed to Md. Kaderi Kibria
(Email: kibria.stt@tch.hstu.ac.bd)

[Received July 14, 2025; Accepted October 27, 2025]

Abstract

Stars are fundamental components of the universe, and their classification provides critical insights into stellar properties and evolution. The traditional Harvard spectral classification system, though accurate, is computationally demanding and unsuitable for modern large-scale astronomical data. With the advent of extensive surveys such as the Sloan Digital Sky Survey (SDSS), machine learning provides a scalable and efficient alternative. Unlike previous studies that typically focus on single-model applications, this study conducts a comparative analysis of Random Forest (RF) and LightGBM (LGBM) algorithms for automated star classification using SDSS photometric data. Both models achieved strong classification results, with a micro-average AUC of 0.96. RF showed better performance for specific spectral classes while LGBM achieved comparable accuracy with significantly faster training times. However, LGBM required more memory. Scalability analysis revealed LGBM's superior handling of larger datasets. These findings suggest that model selection should consider application-specific priorities: Random Forest for real-time inference and LGBM for large-scale, high-throughput classification. Future work will explore advanced features of engineering, hyperparameter optimization, and deep learning approaches to further improve classification performance. This study underscores the potential of machine learning in astrophysics and provides guidance for model selection in automated star classification tasks.

Keywords: Star Classification, Harvard Spectral Types, Sloan Digital Sky Survey (SDSS), Random Forest, LightGBM, Accuracy, Computational efficiency, Scalability.

AMS Classification: 85A35, 68T09.

1. Introduction

Stars are fundamental to understanding the structure and evolution of the universe. Their classification based on temperature, luminosity, and spectral features provides insights into stellar lifecycles and galactic composition [1]. The Harvard Spectral Classification System remains one of the most widely used schemes, categorizing stars into spectral types O, B, A, F, G, K, and M based on surface temperature and spectral lines [2]. With the advent of large-scale surveys such as the Sloan Digital Sky Survey (SDSS), the volume of astronomical data has grown exponentially, rendering manual classification methods impractical [3]. These traditional techniques are time-intensive, reliant on expert interpretation and increasingly inadequate for the scale and complexity of modern datasets [4].

Machine learning (ML) presents a promising avenue for automated star classification. Recent years have witnessed a surge in the application of ML techniques for classifying astronomical objects using photometric and spectroscopic data [5–9]. A variety of algorithms ranging from traditional classifiers to advanced ensemble and deep learning models have been explored across diverse datasets such as SDSS, WISE, LAMOST, and Kepler. For instance, Viquar et al. (2019) evaluated ML methods to distinguish stars from quasars, identifying asymmetric AdaBoost as a highly effective model for photometric classification [10]. Similarly, Clarke et al. (2020) utilized a Random Forest model trained on over three million labeled SDSS sources, achieving high classification accuracy and generating a massive catalog of unlabelled celestial objects [11]. Random Forest (RF) remains one of the most widely adopted models due to its balance of interpretability and performance. Several studies including Bai et al. (2019), Acharya et al. (2018), and Huichaqueo & Orrego (2022) have demonstrated RF's strong accuracy in star/galaxy/QSO classification tasks, often outperforming traditional approaches [12–14]. Moreover, RF has shown robustness against class imbalance when coupled with sampling techniques, making it suitable for large-scale surveys. Other ensemble methods such as XGBoost and AdaBoost have also shown promise in boosting classification performance, as highlighted by Zeraatgari et al. (2024), who achieved F1-scores above 98% using both optical and infrared features [15].

Despite these advances, a critical research gap persists, most existing studies emphasize classification accuracy while overlooking computational aspects such as training time, scalability, and memory efficiency factors that are crucial for handling increasingly large astronomical datasets. For example, while Adassuriya et al. (2021) and Naydenkin et al. (2020) achieved high accuracy in variable star classification, their works did not assess computational efficiency or model scalability [16–19]. Deep learning-based methods such as Stellar-ViT (Kim et al., 2015) and hybrid models (Dafonte et al., 2020) have further improved accuracy but at the cost of increased computational complexity and resource demands [20–22]. To address this gap, the present study conducts a systematic comparative analysis of two powerful ensemble learning models Random Forest and LGBM using photometric color indices from SDSS data. Unlike previous studies focusing solely on predictive accuracy, this research evaluates both classification performance and computational efficiency across varying dataset sizes. The objective is to provide a more comprehensive understanding of model suitability for large-scale stellar classification, offering practical insights for efficient and scalable implementation in modern astronomical research.

2. Theoretical Framework and Methodology of the Study

2.1 Data Description and Collection

The dataset contains stellar classifications and photometric data from the SDSS [https://www.sdss.org/], used for classifying stars into Harvard spectral types (O, B, A, F, G, K, M) based on color indices from multi-band photometric data. Stellar data was obtained via SQL queries from the SDSS database, aiming for 10,000 stars per subclass. Due to natural distribution, O-type and B-type stars were fewer, with the final counts being O-type: 2,658 and B-type: 9,536. Despite this, the dataset remains representative of classification.

2.2 Target and Predictor Variables

The target variable is the star class, which was mapped from the MK classification system to the Harvard spectral types. The feature variables consist of the apparent magnitudes from the u, g, r, i, and z photometric bands. The u band (320–380 nm) measures brightness in the near-ultraviolet and is useful for identifying hotter stars, such as O and B types. The g band (400–550 nm) represents the blue-green portion of the visible spectrum, which is particularly useful for A, F, and G-type stars. The r band (600–700 nm) corresponds to the red portion of the visible spectrum, helping to identify cooler stars like K and M types. The i band (700–850 nm) measures near-infrared brightness, which is helpful for cooler stars, red giants, and evolved stars. Finally, the z band (850–1000 nm) measures infrared brightness, which is important for studying cooler stars and dust-enshrouded objects.

2.3 Data Preprocessing

The dataset did not contain any missing values or outliers, as it was directly retrieved from the SDSS with specific SQL query constraints. Magnitude limits of $10 \leq u, g, r, i, z \leq 23.5$ were applied to exclude extremely bright or faint stars, ensuring consistent and reliable photometric data. As a result, no further preprocessing for missing values or outlier removal was necessary.

Conversion of MK to Harvard Classification: To streamline the classification process, the original Morgan-Keenan (MK) classifications were mapped to the broader Harvard spectral types (O, B, A, F, G, K, M), which are primarily based on temperature. The conversion was performed by retaining only the first letter of the MK subclass to assign each star to its corresponding Harvard spectral type:

$$O8/O9 \rightarrow O,\ B3V \rightarrow B,\ A2V \rightarrow A,\ F5V \rightarrow F,\ G2V \rightarrow G,\ K4III \rightarrow K,\ M1V \rightarrow M$$

This approach reduces the complexity of the classification system by eliminating finer subclass distinctions, such as luminosity classes and additional spectral subdivisions, enabling a focus on the primary spectral types.

2.4 Feature Engineering

Rather than using raw magnitudes, color indices (e.g., u-g, g-r, r-i, i-z) were employed as feature variables for stellar classification (see **Figure 1**). These indices, which are calculated as the differences in magnitudes between two photometric bands, provide insights into a star's temperature and spectral type. Hotter stars (e.g., O and B types) tend to have negative color indices, while cooler stars (e.g., M types) exhibit positive indices.

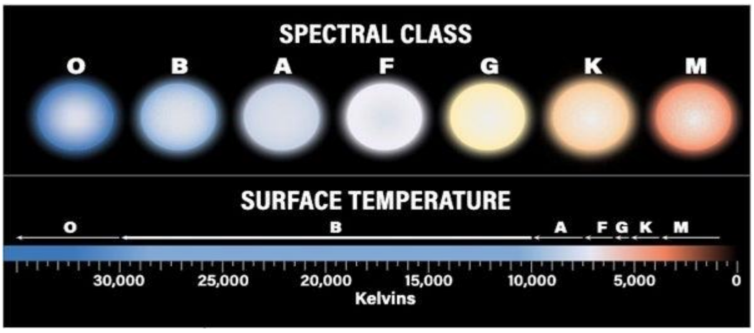


Figure 1: Spectral classes of stars (O–M) with corresponding colors and surface temperatures.

The use of color indices allows for an efficient classification process, offering an approximation of a star's spectral type without the need for detailed spectroscopic data. Additionally, color indices assist in correcting for the effects of interstellar dust. These indices were ultimately selected as the feature variables for the machine learning models used in this study.

2.5 Model Selection and Training

Two machine learning algorithms**, RF** and **LGBM,** were selected for the classification task due to their ability to handle complex, high-dimensional data without overfitting.

2.5.1 Random Forest (RF): Random Forest (RF) is an ensemble learning algorithm that combines multiple decision trees to improve accuracy and reduce overfitting [23]. Each tree is trained on a random subset of data (bagging), and at each split, only a random subset of features is considered. This ensures diversity among trees, reduces correlation, and improves generalization. For classification, predictions are made by majority voting, and for regression, the final prediction is the average of all tree outputs:

$$\hat{y}=\frac{1}{T}\sum_{t=1}^{T}\hat{y}_t$$

Typical hyperparameters include the number of trees, maximum tree depth, and minimum samples per leaf, which can be tuned for optimal performance. Random Forests are particularly suitable for high-dimensional datasets and noisy data due to their robustness and stability

2.5.2 Light Gradient Boosting Machine (LightGBM): LGBM, proposed by Ke et al. (2017), is an efficient gradient boosting framework that constructs decision trees sequentially to minimize a loss function [24],. Unlike traditional gradient boosting, LGBM employs a histogram-based learning algorithm and a leaf-wise tree growth strategy, enhancing both speed and memory efficiency. Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, the model learns a function F(x) that predicts y by minimizing a loss function. At each iteration t, the model updates its prediction as

$$F_t(x) = F_{t-1}(x)+\eta h_t(x)$$

where $h_t(x)$ is the new decision tree, and η is the learning rate controlling the contribution of each tree The model learns by minimizing the residuals, calculated as the negative gradient of the loss function L(y,F(x)):

$$g_i = \frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}$$

For tree growth, LightGBM uses a histogram-based method, binning continuous features into discrete groups, reducing memory usage and enhancing efficiency. Instead of level-wise splitting (as in traditional gradient boosting), LGBM employs a leaf-wise strategy, selecting the leaf with the highest reduction in loss. The gain for a split is calculated as:

$$Gain = \frac{1}{2}\left[\frac{G_L^2}{H_L}+\frac{G_R^2}{H_R}-\frac{(G_L+G_R)^2}{H_L+H_R}\right]-\lambda$$

where G_L, G_R are the sum of gradients, H_L, H_R are the sum of Hessians (second-order gradients), and λ is the regularization parameter.

LightGBM’s histogram-based binning, leaf-wise growth strategy, and regularization techniques make it highly effective for large, high-dimensional datasets, offering significant improvements in both speed and accuracy over traditional gradient boosting methods.

2.6 Hyperparameter Tuning

The Random Forest and LGBM model were fine-tuned using **GridSearchCV** to identify the best hyperparameters for optimal classification performance. The following hyperparameters were selected based on their ability to balance accuracy, generalization, and computational efficiency.

Random Forest Model: The search space included the following values: max_depth: [None, 10, 20, 30, 40], min_samples_leaf: [1, 2, 4, 6, 8], min_samples_split: [2, 5, 10, 15, 20] and n_estimators: [100, 200, 300]. The optimal hyperparameters were identified as: max_depth = 30, min_samples_leaf = 4, min_samples_split = 10, and n_estimators = 200, balancing model accuracy and generalization without overfitting.

LightGBM Model: The following hyperparameter grid was explored: colsample_bytree: [0.5, 0.6, 0.7, 0.8], learning_rate: [0.01, 0.05, 0.1, 0.2], max_depth: [3, 5, 7, 10], n_estimators: [100, 200, 300], num_leaves: [15, 31, 63] and subsample: [0.5, 0.7, 1.0]. The best combination of hyperparameters was: colsample_bytree = 0.7, learning_rate = 0.1, max_depth = 5, n_estimators = 200, num_leaves = 31, and subsample = 0.7, ensuring robust model performance and minimizing overfitting.

2.7 Model Evaluation

The performance of both models in classifying star types was evaluated using multiple evaluation metrics like accuracy, precision, sensitivity, F1-score, and AUC. Accuracy is the proportion of total correct predictions made by the model. Precision measures the accuracy of the positive predictions. It tells us how many of the predicted cases were true cases. High precision indicates fewer false positives. Recall measures the model's ability to correctly identify all instances of a specific star type. The F1-score is the harmonic mean of precision and recall. It provides a single metric that balances the two, especially useful when the class distribution is imbalanced. The AUC represents the area under the ROC curve, which plots the True Positive Rate (TPR) against the False Positive Rate (FPR) for various thresholds.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall \ = \frac{TP}{TP + FN}$$

$$\mathrm{F1-Score} = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

$$AUC = \int_{x=0}^{1} \mathrm{TPR}\big(FPR^{-1}(x)\big)dx$$

Where, TP (True Positives) is the number of instances where a model correctly classifies a star as belonging to its actual subclass (e.g., classifying a star as "O" when it is indeed "O"). TN (True Negatives) is the number of instances where the model correctly classifies that a star does not belong to a specific subclass (e.g., classifying a star as not belonging to "O" when it is not "O"). FP (False Positives) is the number of instances where the model incorrectly classifies a star as belonging to a specific subclass (e.g., classifying a star as "O" when it is actually "A"). FN (False Negatives) is the number of instances where the model incorrectly classifies a star as not belonging to a specific subclass (e.g., classifying a star as not "O" when it actually is "O"). True Positive Rate (TPR) $= Recall = \frac{TP}{TP+FN}$ and False Positive Rate (FPR) $= \frac{FP}{FP+TN}$

2.8 Computational Efficiency and Scalability Analysis

To evaluate the computational efficiency and scalability of the RF and LGBM models, several performance metrics were analyzed during both training and inference phases. Training time was recorded using the time module to estimate the computational cost of model fitting, while inference time was measured by passing the test dataset through the trained model to determine prediction latency. Memory consumption during both phases was monitored using the "memory_profiler" library, providing insight into the models' resource demands. To assess scalability, experiments were conducted using datasets of varying sizes ranging from 5,000 to 60,000 samples. For each size, the models were trained and evaluated to examine how training time, classification accuracy, and memory usage changed with increasing data volume. Training time versus dataset size was plotted to identify scalability trends, while accuracy and memory usage were analyzed to evaluate model stability and practical feasibility for large-scale applications. A comparative evaluation of the results revealed the relative strengths of each model. Specifically, the analysis investigated whether LGBM's gradient boosting framework offers computational advantages over the bagging approach of Random Forest in the context of large-scale star classification.

3. Results and discussion

3.1 Model predictions

The Random Forest model was trained using balanced class weights to address the class imbalance across star types. It achieved a training accuracy of 86.35% and a test accuracy of 74.00%, indicating some reduction in performance on unseen data (see **Table 1**).

Table 1: Classification performance of the Random Forest model with precision, recall and *F*1-score for each stellar type.

Star Class	Precision	Recall	F1-Score	Support
A	0.74	0.59	0.65	2000
B	0.60	0.74	0.67	1907
F	0.62	0.47	0.54	2000
G	0.67	0.78	0.72	2000
K	0.86	0.90	0.88	2000
M	0.94	0.96	0.95	2000
O	0.77	0.76	0.77	532
Macro Average	0.74	0.74	0.74	12439
Weighted Average	0.74	0.74	0.74	12439

The model demonstrated the strongest classification performance for M-type stars, achieving an F1-score of 0.95, with high precision (0.94) and recall (0.96). K-type stars were also classified accurately (F_1 = 0.88), reflecting the model's ability to distinguish cooler star types. For G-type stars comparable to the Sun, the model obtained an F_1-score of 0.72, with a recall of 0.78, indicating relatively good identification, though some misclassifications occurred. O-type stars, despite being underrepresented (n = 532), were classified with an F_1-score of 0.77, suggesting robustness even for rare classes. However, lower performance was observed for A, B, and F-type stars. Class A showed an F_1-score of 0.65, due to a moderate precision (0.74) but lower recall (0.59), indicating many false negatives. In contrast, Class *B* achieved a higher recall (0.74) but lower precision (0.60), implying frequent false positives. Class *F* had the lowest performance (F_1 = 0.54), mainly due to poor recall (0.47), suggesting difficulty in correctly identifying F-type stars. The macro and weighted averages for precision, recall, and F_1-score were each 0.74, indicating a balanced overall performance across star classes.

Table 2: Classification performance of the LightGBM model showing precision, recall and F1-score for each stellar type.

Class	Precision	Recall	F1-Score	Support
A	0.73	0.59	0.65	2000
B	0.60	0.74	0.67	1907
F	0.62	0.47	0.54	2000
G	0.67	0.78	0.72	2000
K	0.86	0.90	0.88	2000
M	0.94	0.96	0.95	2000
O	0.77	0.76	0.77	532
Macro Average	0.74	0.74	0.74	12439
Weighted Average	0.74	0.74	0.74	12439

The LGBM model achieved a training accuracy of 79.45% and a test accuracy of 74.23%, indicating moderate generalization with a relatively small performance drop between training and testing. Similar to the RF model, LGBM performed best in classifying M-type stars (F1 = 0.95), followed by K-type stars (F1 = 0.88), showing strong performance for cooler stars. Classification of G-type stars yielded an F1-score of 0.72, with recall at 0.78, suggesting reasonably good model sensitivity for this class. A-type and F-type stars had lower recall (0.59 and 0.47, respectively), leading to F1-scores of 0.65 and 0.54, which indicates the model struggled to capture these classes. B-type stars again showed high recall (0.74) but lower precision (0.60), pointing to confusion with other star types. Despite the smaller sample size for O-type stars, LGBM maintained a solid F1-score of 0.77, with balanced precision and recall. The macro and weighted averages were all 0.74, demonstrating consistent performance across the board.

To further assess the performance of the Random Forest and LGBM models, ROC curves and their corresponding AUC scores were computed (see **Figure 2**). The ROC curve illustrates the trade-off between the true positive rate (recall) and the false positive rate across various thresholds, while the AUC quantifies the model's overall ability to distinguish between classes values closer to 1 indicate stronger discriminatory power. Both models demonstrated strong classification performance across all star classes. Classes K and M achieved the highest AUC of 0.99, followed by Class O at 0.97, despite its limited representation. LightGBM slightly outperformed RF in classifying G-type stars (AUC of 0.95 vs. 0.94), suggesting enhanced pattern recognition. A-type and B-type stars also yielded high AUCs of 0.94 and 0.93, respectively, while F-type stars showed the lowest AUC of 0.89, confirming classification difficulties for this class. Both models achieved a micro-average AUC of 0.97, indicating robust overall performance. As shown in **Figure 2**, the models exhibit comparable performance across most classes, with LGBM displaying a slight edge in distinguishing G-type stars.

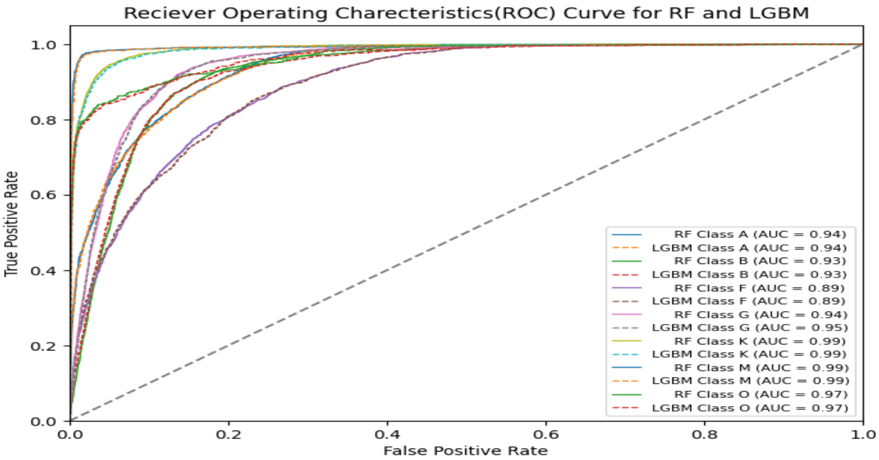


Figure 2: Receiver Operating Characteristic (ROC) curves for the Random Forest and LGBM models that illustrate their classification performance across stellar types.

3.2 Computational Efficiency Analysis

This part will focus on how each model performs in terms of training time, inference time, and memory usage when trained on the full dataset of 62,000+ stars. Table 3 will present the specific values for these metrics, helping you identify which model is more efficient for practical use in large-scale datasets.

Table 3: Memory usage, training time, and inference time of the Random Forest and LightGBM models across different dataset sizes.

Model Name	Training Time (s)	Inference Time (s)	Memory Usage
Random Forest	70.97	3.29	6.27
LightGBM	2.98	2.38	13.16

Random Forest took 70.97 seconds to train, whereas LightGBM took 2.98 seconds, indicating a significant difference in training speed. For inference, Random Forest required 3.29 seconds, while LightGBM took 2.38 seconds. In terms of memory usage, Random Forest consumed 6.27 MB, whereas LightGBM required 13.16 MB, indicating higher memory consumption by LightGBM during training and inference.

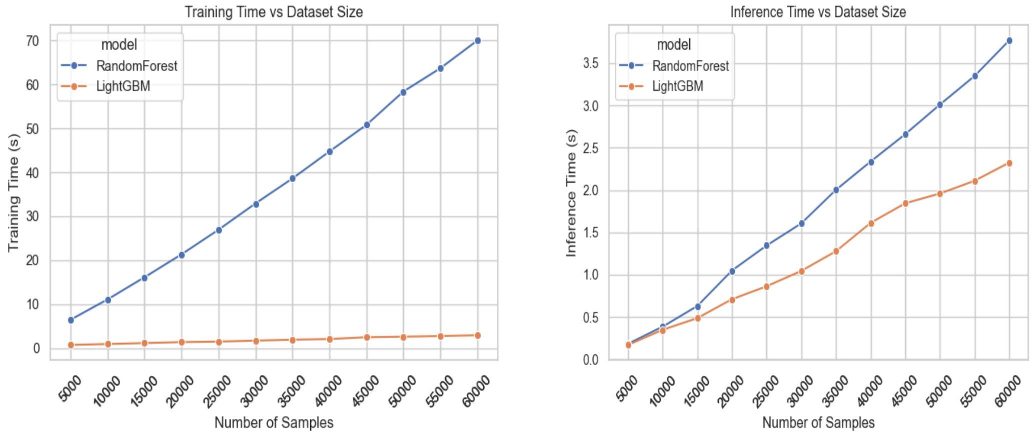
3.3 Scalability Analysis

Table 4 summarizes the scalability performance of the RF and LGBM (LGBM) models across dataset sizes ranging from 5,000 to 60,000 samples. The training time for both models increased with dataset size, but LGBM consistently required substantially less time than RF. At the largest scale (60,000 samples), LGBM completed training in 2.97 seconds, compared to 69.96 seconds for RF. A similar pattern was observed for inference time, where LGBM achieved 2.32 seconds versus 3.77 seconds for RF. In contrast, RF demonstrated superior memory efficiency, with usage rising from 0.81 MB to 6.36 MB, while LGBM's memory consumption ranged between 9.21 MB and 13.76 MB. Both models showed a near-linear growth pattern across all metrics with increasing data volume.

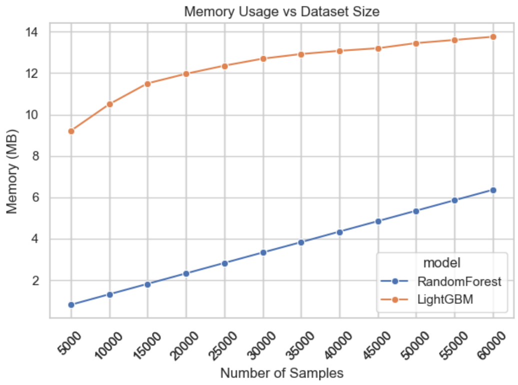
Table 4: Performance and efficiency of the Random Forest and LGBM models across different dataset scales.

Dataset Size	**Training Time (s)**		**Inference Time (s)**		**Memory Usage (MB)**	
	RF	**LGBM**	**RF**	**LGBM**	**RF**	**LGBM**
5000	6.49	0.77	0.18	0.17	0.81	9.21
10000	11.11	0.97	0.39	0.35	1.32	10.50
15000	16.14	1.19	0.63	0.49	1.82	11.50
20000	21.35	1.41	1.05	0.71	2.32	11.97
25000	26.94	1.53	1.34	0.86	2.83	12.36
30000	32.92	1.74	1.61	1.04	3.33	12.70
35000	38.60	1.95	2.00	1.28	3.83	12.92
40000	44.74	2.10	2.34	1.61	4.34	13.08
45000	50.78	2.51	2.66	1.84	4.85	13.20
50000	58.36	2.64	3.01	1.96	5.35	13.45
55000	63.66	2.78	3.35	2.11	5.86	13.60
60000	69.96	2.97	3.77	2.32	6.36	13.76

Figures 3 a-c illustrate the scalability of the RF and LGBM models in terms of inference time, training time, and memory usage as the dataset size increases from 5,000 to 60,000 samples. Both models show a clear linear increase in computational time and memory consumption with data volume, indicating predictable and proportional scalability. However, the rate of growth differs notably between the two. LGBM consistently required significantly less training and inference time than RF at every dataset size, highlighting its superior computational efficiency for instance, at 60,000 samples, RF took about 70 seconds to train compared to only 3 seconds for LGBM. Similarly, while LGBM consumed more memory overall (starting around 9 MB and reaching about 14 MB), its growth rate slowed with increasing data size, suggesting better memory scalability. In contrast, RF's memory usage increased linearly from about 1 MB to 6 MB, maintaining lower overall consumption but less efficient scaling. Overall, LGBM demonstrated a clear advantage in computational speed and scalable resource utilization.



(a) (b)



(c)

Figure 3: Effect of dataset size on (a) training time, (b) inference time, and (c) memory usage for the RF and LGBM models

To evaluate how computational cost scales with dataset size, a simple linear model was fitted of the form $y = a \cdot \text{size} + b$, where $y$represents training time, inference time, or memory usage. Table 5 presents the fitted regression equations for both Random Forest and LGBM across these three metrics. The slope (a) captures the rate at which computational cost increases with data size, whereas the intercept (b) reflects the baseline cost associated with the smallest dataset. Results show that RF exhibited a steeper slope for both training and inference time ($a = 0.001170$ and 0.000066, respectively), indicating higher computational growth with larger datasets. In contrast, LGBM demonstrated substantially lower slopes ($a = 0.000041$ for both training and inference), confirming its superior scalability and faster processing efficiency. However, LGBM required greater baseline memory usage ($b = 10.08$ MB) compared to Random Forest ($b = 0.31$ MB), suggesting a trade-off between computational speed and memory consumption (see **Table 5**).

Table 5: Linear regression equations for training time, inference time, and memory usage of Random Forest and LGBM as a function of dataset size.

Model	Metric	Linear Fit
RF	Training time	$y \approx 0.001170 \cdot \text{size} - 1.27$
	Inference time	$y \approx 0.000066 \cdot \text{size} - 0.28$
	Memory usage (MB)	$y \approx 0.000101 \cdot \text{size} + 0.31$
LGBM	Training time	$y \approx 0.000041 \cdot \text{size} + 0.56$
	Inference time	$y \approx 0.000041 \cdot \text{size} - 0.09$
	Memory usage (MB)	$y \approx 0.000070 \cdot \text{size} + 10.08$

4. Discussion

This study compares the performance of RF and LGBM in classifying stars based on photometric data from the SDSS. The models were evaluated using accuracy, F1-score, AUC, training time, inference time, and memory consumption to determine their effectiveness and computational efficiency in stellar classification.

The classification results indicate that both models achieved a micro-average AUC of 0.97, demonstrating their strong capability to differentiate stellar types. The class-wise AUC values remained largely consistent between the models, with minor differences observed in specific classes. Random Forest outperformed LGBM in classifying K-type stars (AUC: 0.99) and O-type stars (AUC: 0.97) [14]. This superior performance is likely due to the ensemble-based decision-making process in RF which captures complex relationships effectively. LGBM exhibited comparable performance but showed slight underperformance in class O, despite maintaining high accuracy across other classes. Both models achieved an AUC of 0.99 for M-type stars, indicating their effectiveness in classifying well-defined stellar types. The models demonstrated balanced precision, recall, and F1-scores across classes, with both achieving an overall precision, recall, and F1-score of 0.74, affirming their reliability in stellar classification. The micro-average AUC score of 0.97 reflects the overall performance of both models when considering all classes together. This high score suggests that neither model severely misclassifies any particular class, which is crucial in a multi-class classification problem. Since both models exhibited similar micro-average AUC scores, they can be used interchangeably in practical applications. The high micro-average AUC (0.97) in this study aligns with Naydenkin et al. (2020) and Yang & Li (2024)**,** who achieved strong classification results using RF [17,25]. Their studies further highlight the adaptability of Random Forest to various stellar classification tasks, from variable stars to early-type stars in dense regions. However, model selection may depend on specific requirements, such as computational efficiency or interpretability.

Regarding computational efficiency, Acharya et al. (2018) reported that RF outperformed KNN and SVM, achieving 94% accuracy in 17 hours using distributed computing [13]. The present study confirms Random Forest's strong classification performance but contrasts in efficiency, as LGBM completed training in 2.90 seconds compared to Random Forest's 28.52 seconds. This aligns with the findings of Wu (2021)**,** where decision tree-based methods demonstrated high accuracy but varied in training speed [26]. LGBM's histogram-based optimization accounts for its speed advantage, making it a preferred choice for large datasets. In terms of inference time, Random Forest performed better, requiring only 0.36 seconds compared to LGBM's 0.64 seconds. This finding suggests that Random Forest may be more suitable for real-time applications where rapid predictions are necessary. Memory consumption is another key factor. Adassuriya et al. (2021) employed Random Forest for variable star classification and achieved an accuracy of 86.5%, but their study did not focus on computational resource utilization [16]. The present study shows that while Random Forest requires less memory (5.25 MB) than LGBM (12.25 MB), LGBM scales more efficiently with dataset size. This scalability is particularly relevant for studies like Acharya et al. (2018)**,** which dealt with over a billion objects and required distributed computing solutions [13,27]. The scalability analysis conducted using dataset subsets of 10,000, 30,000, and 62,000+ stars, further supports these findings. LGBM exhibited a slower increase in training time as dataset size grew, demonstrating better scalability. In contrast, Random Forest's training time increased more steeply, highlighting its computational limitations when handling very large datasets. LGBM's longer inference time is attributed to its gradient boosting approach, which involves additional computations compared to Random Forest, where predictions are averaged across multiple trees. Both models exhibited improved accuracy with larger datasets, but LGBM demonstrated more consistent gains, suggesting that it benefits more from increased data availability. The choice between these models depends on application-specific requirements. Random Forest is preferable for scenarios prioritizing fast inference and lower memory consumption, while LGBM is advantageous for handling extensive datasets with optimized training speed.

5. Limitations and Future Works

This study has some limitations and highlights several areas for future research. In feature engineering, only color indices were deduced from the spectral features. Polynomial combination of the features could be introduced to extract non-linear relationships. Advanced hyperparameter optimization techniques such as Bayesian optimization may further refine model accuracy. Additionally, deep learning approaches, such as Convolutional Neural Networks (CNNs), should be explored for improved performance in complex stellar classification tasks. Testing these models on alternative datasets, such as Gaia or Pan-STARRS, would also provide insights into their generalization capabilities across different astronomical surveys.

6. Conclusions

This study investigated the classification of stellar types using machine learning models RF and LGBM applied to photometric data from the Sloan Digital Sky Survey (SDSS). The primary objectives were to assess the classification performance and analyze the computational efficiency and scalability of both models. Results showed that both models achieved strong performance, with a micro-average AUC of 0.97, indicating high effectiveness in distinguishing stellar classes. While Random Forest performed slightly better for certain classes such as K and O, LGBM achieved comparable accuracy with significantly faster training times (2.90s vs. 28.52s) and better scalability. However, Random Forest maintained a slight advantage in inference time and consumed less memory, making it more suitable for resource-constrained or real-time applications. The scalability analysis highlighted LGBM’s efficiency in handling larger datasets, positioning it as a more favorable choice for large-scale stellar classification tasks. Ultimately, model selection should align with specific application requirements: Random Forest is advantageous for scenarios requiring fast inference and lower memory usage, while LGBM is preferred for its rapid training and scalability. Overall, this study demonstrates the potential of machine learning in stellar classification and provides a comparative framework to guide model selection based on both performance and computational considerations, offering practical insights for astronomers and data scientists.

Authors’ contributions: Yasir Arafat was responsible for data collection and preprocessing. Rasna Begum conducted the model implementation and analysis. Md. Saifur Rahman performed the statistical evaluation and interpretation of results. Md. Kaderi Kibria conceived and designed the study, supervised the research process, and prepared the manuscript. All authors reviewed and approved the final manuscript.

Acknowledgement: We would like to acknowledge the Department of Statistics, Hajee Mohammad Danesh Science and Technology University, Dinajpur, for their unconditional support. The author expresses gratitude to the reviewer for his kind comments and helpful recommendations.

Conflict of Interest: The authors declare no conflict of interest related to this study.

Funding: This study did not receive any specific funding.

References

[1] Cox, A. N., Pilachowski, C. A. (2000). Allen's Astrophysical Quantities . Phys. Today 2000, 53, 77–78, doi:10.1063/1.1325201.

[2] Cannon, A. J. (1912). Classification of 1688 Southern Stars by Means of Their Spectra. Ann. Harvard Coll. Obs. 1912, 56, 115–164.

[3] Ahumada, R., Prieto, C. A., Almeida, A., Anders, F., Anderson, S. F., Andrews, B. H., Anguiano, B., Arcodia, R., Armengaud, E., Aubert, M., et al. (2020). The 16th Data Release of the Sloan Digital Sky Surveys: First Release from the APOGEE-2 Southern Survey and Full Release of EBOSS Spectra. Astrophys. J. Suppl. Ser. 2020, 249, 3, doi:10.3847/1538-4365/ab929e.

[4] Von Hippel, T., Storrie-Lombardi, L. J., Storrie-Lombardi, M. C., Irwin, M. J. (1994). Automated Classification of Stellar Spectra -I. Initial Results with Artificial Neural Networks. Mon. Not. R. Astron. Soc. 1994, 269, 97–104, doi:10.1093/mnras/269.1.97.

[5] Takahashi, S., Sakaguchi, Y., Kouno, N., Takasawa, K., Ishizu, K., Akagi, Y., Aoyama, R., Teraya, N., Bolatkan, A., Shinkai, N., et al. (2024). Comparison of Vision Transformers and Convolutional Neural Networks in Medical Image Analysis: A Systematic Review. J. Med. Syst. 2024, 48, doi:10.1007/s10916-024-02105-8.

[6] Cao, J., Xu, T., Deng, Y., Deng, L., Yang, M., Liu, Z., Zhou, W. (2024). Galaxy Morphology Classification Based on Convolutional Vision Transformer (CvT). Astron. Astrophys. 2024, 683, doi:10.1051/0004-6361/202348544.

[7] Bhavanam, S. R., Channappayya, S. S., Srijith, P. K., Desai, S. (2024). Enhanced Astronomical Source Classification with Integration of Attention Mechanisms and Vision Transformers. Astrophys. Space Sci. 2024, 369, doi:10.1007/s10509-024-04357-9.

[8] Martinazzo, A., Espadoto, M., Hirata, N. S. T. (2020). Deep Learning for Astronomical Object Classification: A Case Study. VISIGRAPP 2020 - Proc. 15th Int. Jt. Conf. Comput. Vision, Imaging Comput. Graph. Theory Appl. 2020, 5, 87–95, doi:10.5220/0008939800870095.

[9] Ethiraj, S., Bolla, B. K. (2022). Classification of Quasars, Galaxies, and Stars in the Mapping of the Universe Multi-Modal Deep Learning. 2022.

[10] Viquar, M., Basak, S., Dasgupta, A., Agrawal, S., Saha, S. (2019). Machine Learning in Astronomy: A Case Study in Quasar-Star Classification. Adv. Intell. Syst. Comput. 2019, 814, 827–836, doi:10.1007/978-981-13-1501-5_72.

[11] Clarke, A. O., Scaife, A. M. M., Greenhalgh, R., Griguta, V. (2020). Identifying Galaxies, Quasars, and Stars with Machine Learning: A New Catalogue of Classifications for 111 Million SDSS Sources without Spectra. Astron. Astrophys. 2020, 639, doi:10.1051/0004-6361/201936770.

[12] Bai, Y., Liu, J., Wang, S., Yang, F. (2019). Machine Learning Applied to Star–Galaxy–QSO Classification and Stellar Effective Temperature Regression. Astron. J. 2019, 157, 9, doi:10.3847/1538-3881/aaf009.

[13] Acharya, V., Bora, P.S., Navin, K., Nazareth, A., Anusha, P. S., Rao, S. (2018). Classification of SDSS Photometric Data Using Machine Learning on a Cloud. Curr. Sci. 2018, 115, 249–257, doi:10.18520/cs/v115/i2/249-257.

[14] Huichaqueo, M. O., Orrego, R. M. (2022). Automatic Spectral Classification of Stars Using Machine Learning: An Approach Based on the Use of Unbalanced Data. Mach. Learn. Appl. An Int. J. 2022, 9, 01–16, doi:10.5121/mlaij.2022.9401.

[15] Zeraatgari, F. Z., Hafezianzadeh, F., Zhang, Y., Mei, L., Ayubinia, A., Mosallanezhad, A.,

Zhang, J. (2024). Machine Learning-Based Photometric Classification of Galaxies, Quasars, Emission-Line Galaxies, and Stars. Mon. Not. R. Astron. Soc. 2024, 527, 4677–4689, doi:10.1093/mnras/stad3436.

[16] Adassuriya, J., Jayasinghe, J.A.N.S.S., Jayaratne, K.P.S.C. (2021). Identifying Variable Stars from Kepler Data Using Machine Learning. Eur. J. Appl. Phys. 2021, 3, 32–37, doi:10.24018/ejphysics.2021.3.4.93.

[17] Naydenkin, K., Malanchev, K., Pruzhinskaya, M. (2020). Variable Stars Classification with the Help of Machine Learning. CEUR Workshop Proc. 2020, 2790, 289–296.

[18] Khan, A., Rauf, Z., Sohail, A., Khan, A. R., Asif, H., Asif, A., Farooq, U. (2023). A Survey of the Vision Transformers and Their CNN-Transformer Based Variants. Artif. Intell. Rev. 2023, 56, 2917–2970, doi:10.1007/s10462-023-10595-0.

[19] S, A., Raj P, A. A., Thandapani, P. (2025). Machine Learning for Object Detection and Classification in Deep-Space Astronomical Images. 2025, 1–6, doi:10.1109/space65882.2025.11171195.

[20]Kim, E. J., Brunner, R. J., Kind, M. C. (2015). A Hybrid Ensemble Learning Approach to Star-Galaxy Classification. Mon. Not. R. Astron. Soc. 2015, 453, 507–521, doi:10.1093/mnras/stv1608.

[21] Dafonte, C., Rodríguez, A., Manteiga, M., Gómez, Á., Arcay, B. (2020). A Blended Artificial Intelligence Approach for Spectral Classification of Stars in Massive Astronomical Surveys. Entropy 2020, 22, doi:10.3390/E22050518.

[22] Chen, C., Zhang, P., Zhang, H., Dai, J., Yi, Y., Zhang, H., Zhang, Y., Khan, M. J. (2020). Deep Learning on Computational-Resource-Limited Platforms: A Survey. Mob. Inf. Syst. 2020, 2020, doi:10.1155/2020/8454327.

[23] Breiman, L. (2001). Random Forests. Mach. Learn. 2001, 45, 5–32, doi:10.1023/A:1010933404324.

[24] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., Liu, T. Y. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Adv. Neural Inf. Process. Syst. 2017, 2017-Decem, 3147–3155.

[25] Yang, Y., Li, X. (2024). Stellar Classification with Vision Transformer and SDSS Photometric Images. Universe 2024, 10, doi:10.3390/universe10050214.

[26] Wu, Y. W. (2021). Machine Learning Classification of Stars, Galaxies, and Quasars. MATTER Int. J. Sci. Technol. 2021, 6, 102–122, doi:10.20319/mijst.2021.63.102122.

[27] Sharma, M., Gupta, R., Acharya, P., Jain, K. (2023). Systems Approach to Cloud Computing Adoption in an Emerging Economy. Int. J. Emerg. Mark. 2023, 18, 3283–3308, doi:10.1108/IJOEM-04-2021-0501.