Institut Riset dan Publikasi Indonesia (IRPI)
MALCOM: Indonesian Journal of Machine Learning and Computer Science
Journal Homepage: https://journal.irpi.or.id/index.php/malcom
Vol. 4 Iss. 3 July 2024, pp: 753-763
ISSN(P): 2797-2313 | ISSN(E): 2775-8575

Exoplanet Classification Through Machine Learning: A Comparative Analysis of Algorithms Using Kepler Data

Gregorius Airlangga

Information System Study Program, Atma Jaya Catholic University of Indonesia, Indonesia

E-Mail: gregorius.airlangga@atmajaya.ac.id

Received Feb 10th 2024; Revised Mar 15th 2024; Accepted Apr 15th 2024
Corresponding Author: Gregorius Airlangga

Abstract

In our study, we employed a suite of advanced machine learning models to classify exoplanet candidates from the Kepler Space Telescope dataset. The aim was to evaluate and compare the efficacy of different algorithms in distinguishing potentially habitable planets. Our research involved a rigorous preprocessing stage to ensure data integrity, followed by a careful selection of features with significant predictive power. The models included a range of machine learning techniques, from basic algorithms like Logistic Regression to more complex ones such as ensemble and boosting methods like Random Forest, Decision Trees, XGBoost, LightGBM, and CatBoost. Each model was evaluated based on precision, recall, and F1 scores, with 10-fold cross-validation to ensure robustness against overfitting and reliability in predictions. Our results were indicative of a high level of performance among ensemble and boosting models, with LightGBM, AdaBoost, and CatBoost achieving near-perfect mean scores of approximately 0.99 across all metrics, showcasing their strong predictive capabilities and stability. In stark contrast, Gaussian Naive Bayes and Logistic Regression lagged behind, with significantly lower F1 scores of around 0.69 and 0.75, respectively. The research underscores the potential of ensemble and boosting-based models in handling complex classification tasks within the realm of exoplanetary science. It also highlights the necessity of applying rigorous evaluation frameworks to ensure model reliability, particularly in disciplines where the accurate classification of data is paramount. Future work should consider the continuous evolution of data and incorporate new findings into the model training process for enhanced prediction accuracy in practical, real-world scenarios.

Keywords: Comparative Analysis, Exoplanet Classification, Kepler Data, Machine Learning, ML Algorithms

1. INTRODUCTION

The cosmos, vast and brimming with mysteries, has captivated humanity's imagination for millennia. The endeavor to understand our place in the universe has led us from the naked eye observation of constellations to the deployment of sophisticated space telescopes like Kepler [1]–[3]. The discovery of exoplanets, planets that orbit stars beyond our solar system, represents a quantum leap in this quest [4]–[6]. These celestial bodies, ranging from gas giants many times the size of Jupiter to rocky worlds that might bear similarities to Earth, are key to unlocking mysteries about the formation of planetary systems and the potential for life elsewhere in the universe [7]. The Kepler Space Telescope's mission, which has identified thousands of exoplanet candidates, serves as a cornerstone for this research, offering unparalleled data for scientific exploration [8]. The exponential growth in exoplanet discoveries has been paralleled by an explosion in data complexity and volume. This has necessitated a shift towards more sophisticated analytical methods. Machine learning (ML) has emerged as a pivotal technology in this context, offering innovative tools to sift through and analyze the vast datasets generated by telescopes like Kepler [9]–[11]. The literature in this domain showcases a progression from simpler ML models to complex ensemble methods and deep learning algorithms [12]. Each model brings specific strengths to the table; for instance, Decision Trees offer simplicity and interpretability, while algorithms like XGBoost and LightGBM are celebrated for their predictive accuracy and efficiency [13]–[15]. However, the literature also reveals a fragmented approach, with many studies focusing narrowly on individual models or specific aspects of the data, rather than embracing a holistic, comparative analysis across a broad spectrum of ML techniques [16].

The search for habitable exoplanets is more than a scientific pursuit; it's a race against time as we seek alternatives or complements to Earth's dwindling resources and escalating environmental challenges [17]. The urgency of this quest is met with state-of-the-art ML algorithms, which have significantly advanced in recent years [18]. These developments are not just technical triumphs but represent a paradigm shift in how we

approach astronomical data [19]. The advent of deep learning, for example, offers the potential to uncover patterns and correlations in data that were previously beyond our reach [20]. Despite these advancements, the state-of-the-art is constantly evolving, driven by breakthroughs in algorithmic strategies, computational power, and our ever-deepening understanding of the cosmos [21]. This research aims to capitalize on the advancements in ML to significantly enhance the classification and understanding of exoplanets [22]. By employing a comprehensive dataset from the Kepler mission, the study seeks to refine the classification of exoplanets into confirmed planets, candidates, and false positives with unprecedented accuracy [23]. More than a mere exercise in classification, this research endeavors to peel back the layers of the dataset to uncover the underlying astrophysical characteristics that dictate these classifications [24]. The goal is to provide not only a more efficient tool for exoplanet discovery but also to contribute to the foundational knowledge about these distant worlds, offering insights into their nature, formation, and potential habitability [25].

The application of ML in the realm of exoplanet research has certainly advanced, yet a closer examination of existing literature reveals significant gaps. The integration of diverse ML models, particularly through ensemble techniques or the application of cutting-edge neural networks, remains underexplored [26]. Additionally, there is a conspicuous absence of systematic, comparative studies of ML models applied to uniform datasets, which is critical for establishing benchmarks and understanding model efficacy [27]. Moreover, the potential of feature importance analysis in revealing the astrophysical significance behind model predictions is often overlooked, leaving a gap in our understanding of the data's deeper, scientific implications [28]. This study aims to bridge these gaps by conducting an exhaustive comparative analysis of a wide range of ML models, including Logistic Regression, Decision Trees, Random Forest, and more advanced algorithms like XGBoost, LightGBM, and CatBoost, applied to the Kepler dataset. This analysis is designed to identify the most effective models for exoplanet classification and to benchmark their performance. Furthermore, by employing feature importance analysis, this research seeks to illuminate the specific astrophysical features that are most predictive of exoplanet classification, thereby contributing to a deeper understanding of the data and the phenomena it represents [29].

The article is meticulously organized to guide readers through the research journey in a coherent and comprehensive manner. After this detailed introduction, the methodology section will outline the data preprocessing steps, feature selection rationale, and model training and evaluation procedures, providing clarity on the research design and execution. The results section will not only present the comparative performance of the ML models but will also interpret these findings within the astrophysical context, enhancing the scientific value of the study. The discussion section will explore the implications of these findings for the broader field of exoplanet research, contemplating their relevance for future missions and the search for habitable worlds. Finally, the conclusion will summarize the study's contributions, reflect on its limitations, and propose directions for future research, ensuring a comprehensive closure to the article.

2. MATERIALS AND METHOD

2.1. Dataset Description

The foundation of this study is a rich dataset sourced from the cumulative files of the Kepler space telescope mission, hosted by the NASA Exoplanet Archive. This dataset encompasses a broad array of observations and measurements pertaining to over 4,000 exoplanet candidates identified by Kepler, encompassing both planetary and stellar characteristics, along with observational metrics crucial for the classification task. The variables included, such as planetary radius, orbital period, and equilibrium temperature for planets, alongside temperature, radius, and mass for stars, are instrumental in distinguishing between confirmed exoplanets, candidates, and false positives, thereby serving as the backbone for the ensuing machine learning analysis. Data can be downloaded from [30]. All codes are open source and can be accessed by accessing *https://github.com/techmentalist/exoplanet*.

2.2. Data Preprocessing

The initial stage of data preprocessing focused on cleansing the dataset to uphold the integrity of the analysis. This involved a meticulous examination to identify and exclude incomplete or missing entries, particularly in variables critical to the classification of exoplanets. Outliers, defined by their significant deviation from the dataset's overall statistical distribution, were also carefully removed to prevent potential skewing of the analysis results. Further refining the dataset, columns were renamed for consistency and clarity, aligning with established astronomical nomenclature, while those deemed irrelevant or redundant to the classification objectives were eliminated. To address the challenge of missing values within numerical features, imputation techniques were employed, substituting absent entries with the mean value of the respective column, thereby preserving the dataset's comprehensiveness without compromising the quality of the data.

2.3. Feature Selection

An exploratory data analysis (EDA) was conducted to scrutinize the relationship between various features and their impact on the classification outcome. This critical process aimed to distill the dataset to a

core set of features with significant predictive power or scientific relevance to the characteristics of exoplanets. By assessing correlations and leveraging domain knowledge, the study ensured that the selected features were both statistically and scientifically grounded, enhancing the reliability and interpretability of the machine learning models' predictions.

2.4. Model Selection and Training

In an endeavor to identify the most effective machine learning classifier for exoplanet data, the study encompassed a broad spectrum of models, ranging from simpler algorithms like Logistic Regression, known for its baseline performance in classification tasks, to more complex models such as Decision Trees and Random Forest Classifiers, chosen for their interpretability and robustness against overfitting. Additionally, advanced gradient boosting machines, including XGBoost, LightGBM, and CatBoost, were selected for their proficiency in handling tabular data and optimizing prediction accuracy. The inclusion of Neural Networks aimed to explore the potential of deep learning in capturing intricate, nonlinear relationships within the data. Each model underwent rigorous training, with parameter optimization and cross-validation to mitigate overfitting and validate their generalizability across unseen data.

2.5. Evaluation Metrics

In the evaluation segment of this research, a detailed and rigorous approach was adopted to assess the performance of various machine learning models in classifying data from the Kepler Space Telescope. The evaluation framework hinges on a comprehensive set of metrics, each offering unique insights into the models' abilities to predict the classification of exoplanets accurately and reliably. This multifaceted evaluation strategy is pivotal for dissecting the nuanced capabilities of each model across different aspects of the classification challenge. Accuracy served as the initial gauge of performance, offering a broad-strokes assessment of how often models correctly identified the classification labels. However, the utility of accuracy as a sole metric is limited, especially in datasets where the distribution of classes is imbalanced. To counteract potential distortions in performance interpretation, precision was employed to measure the quality of positive identifications made by the models, ensuring that the rate of false positives was kept to a minimum.

Complementing precision, recall was used to evaluate the models' sensitivity in capturing all relevant instances of a class. This metric is particularly critical in scenarios were failing to detect a positive instance can carry significant consequences. To reconcile the tension between precision and recall, the F1 score was introduced as a harmonized measure that encapsulates both metrics, rewarding models that maintain a balance between accurately identifying positive instances and minimizing false positives. Beyond these metrics, the study integrated 10-fold cross validation as a pivotal component of the evaluation strategy. This technique, which splits the dataset into ten parts to conduct a series of train-and-test cycles, not only helps in reducing the risk of overfitting but also ensures a more robust estimation of the models' performance on unseen data. By leveraging this method, the research offers a more reliable and generalized understanding of how each model performs, fostering confidence in their predictive accuracy and consistency.

2.6. Software and Tools

The analytical framework for this study was built upon Python, leveraging its rich ecosystem of libraries such as Pandas for data manipulation, Scikit-learn for machine learning algorithms and evaluation, and Matplotlib and Seaborn for visualization. The selection of Python and these specific libraries was driven by their widespread acceptance within the data science community, comprehensive documentation, and support for an extensive range of machine learning techniques, ensuring both the efficiency and reproducibility of the research.

2.7. Logistic Regression

Logistic regression is a statistical model that in its basic form uses a logistic function to model a binary dependent variable, although more complex extensions exist for multiclass classification. Unlike linear regression which predicts a continuous outcome, logistic regression is used for binary classification tasks where the output is discrete, typically representing the presence or absence of an event or class (e.g., positive or negative). In logistic regression, the probability that a given input point belongs to the 'positive' class is modeled as the logistic function of a linear combination of the predictors. This model can be represented mathematically as equation 1.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n)}} \quad (1)$$

Here, $(p(x))$ denotes the probability of the positive class, (β_0) is the intercept term, $(\beta_1, \ldots, \beta_n)$ are the coefficients for the predictors $(x_1, \ldots, x_n)$, and (e) is the base of the natural logarithm. The equation essentially defines a decision boundary that is the set of points where $(p(x) = 0.5)$, above which we predict the positive

class and below which we predict the negative class. To estimate the parameters (β), logistic regression uses the method of maximum likelihood. The likelihood function for a dataset is given in the equation 2.

$$L(beta) = \prod_{i=1}^{m} p\left(x^{(i)}\right)^{y^{(i)}} \left(1 - p\left(x^{(i)}\right)\right)^{1-y^{(i)}} \quad (2)$$

Where (m) is the number of samples in the dataset, ($x^{(i)}$) is the vector of predictors for the (i)-th sample, and ($y^{(i)}$) is the corresponding binary class label. The goal is to find the values of (β) that maximize this likelihood function. In practice, it is more common to work with the log of the likelihood function, known as the log-likelihood as presented in equation 3.

$$l(\beta) = \sum_{i=1}^{m} \left[y^{(i)} \log\left(p\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - p\left(x^{(i)}\right)\right)\right] \quad (3)$$

Maximizing the log-likelihood is computationally more stable and often more convenient than maximizing the likelihood function itself. Furthermore, Logistic regression can be regularized to prevent overfitting. This involves introducing a penalty term to the optimization problem that constrains the size of the coefficients. Regularization methods such as L1 (Lasso) and L2 (Ridge) are commonly used, with the former being useful for feature selection due to its property of shrinking some coefficients to zero. The model can then be used to make predictions for new data points by plugging in the feature values into the logistic function with the estimated coefficients.

2.8. Decision Tree

Decision Trees are a type of non-parametric supervised learning algorithm used for both classification and regression tasks. They allow for predictive modeling of decisions and their possible consequences, resembling a flowchart-like tree structure. In a Decision Tree, each internal node represents a "test" on an attribute, each branch corresponds to the outcome of the test, and each leaf node signifies a class label or regression value. The paths from the root to the leaf represent classification rules or regression predictions. The core challenge in building a Decision Tree is determining how to split the data at each node. This is typically done using measures such as Gini impurity or information entropy. The Gini impurity is a measure that quantifies the frequency at which any element of the dataset will be mislabeled if it were randomly labeled according to the distribution of labels in the subset. Mathematically, the Gini impurity for a set can be calculated using equation 4.

$$Gini(t) = 1 - \sum_{j} [p(j|t)]^2 \quad (4)$$

Where (p(j|t)) is the relative frequency of class (j) at node (t). A Gini score gives an idea of how good a split is by measuring how mixed the classes are in the two groups created by the split. The best split is chosen by the model by selecting the split with the lowest Gini impurity compared to other splits. Building a Decision Tree involves selecting the best attribute using an Attribute Selection Measure (ASM) to split the records and then breaking the dataset into smaller subsets. The process starts at the root node and is repeated recursively for each child node, continuing until one of the following conditions is met: all tuples at a node belong to the same attribute value, there are no more attributes left to be selected for splitting, or there are no more instances. To avoid the common pitfall of overfitting, where the tree performs well on training data but poorly on unseen data, Decision Trees can be pruned. Pruning involves the removal of branches that have little power in predicting the target variable. This can be done by setting a minimum threshold on the number of samples that a leaf can have or the maximum depth of the tree. In the context of regression, Decision Trees predict numerical values instead of class labels. In such cases, the Mean Squared Error (MSE) is often used to measure the quality of a split, with the goal of minimizing the MSE at each leaf of the tree as presented in equation 5.

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 \quad (5)$$

Where (N) is the number of samples, (y_i) is the observed value, and ($\hat{y}_i$) is the predicted value. Decision Trees are popular due to their interpretability, simplicity, and ability to handle both numerical and categorical data. However, they are prone to overfitting if not properly tuned or pruned.

2.9. Random Forest

Random Forest is a sophisticated ensemble learning technique used for both classification and regression tasks that enhances decision-making capabilities by leveraging the collective power of multiple decision trees. Unlike traditional decision tree methods that rely on a single tree and may suffer from overfitting, Random Forest builds a forest of trees to improve prediction accuracy and control overfitting. The fundamental principle behind Random Forest is to create an ensemble of decision trees during the training phase and then use these trees to make predictions. Each tree in the ensemble is constructed using a bootstrap sample of the data, meaning a sample drawn with replacement from the original dataset. This approach ensures diversity among the trees in the model, which is a key factor in the algorithm's robustness.

When constructing individual trees, Random Forest introduces randomness by selecting a random subset of features at each split. This randomness helps in making the ensemble of trees more diverse, thereby reducing the variance without significantly increasing the bias. For a classification task, the prediction by the Random Forest model is determined by the mode of the predictions from all the trees within the forest. Mathematically, this can be expressed as presented in equation 6.

$$\hat{y} = \text{mode}\{y_b(x)\}_{b=1}^{B} \tag{6}$$

Where $(\hat{y})$ is the final prediction, (B) represents the total number of trees in the forest, and $(y_b(x))$ denotes the prediction made by the (b)-th decision tree. For regression tasks, the final output is usually the average of the outputs from all the trees. One of the notable advantages of Random Forest is its ability to evaluate the importance of different features in the prediction. By measuring how much each feature decreases the impurity in the splits it creates, the algorithm can rank the features according to their contribution to the model's predictive power.

2.10. XGBoost

XGBoost, short for Extreme Gradient Boosting, is distinguished by its efficiency, scalability, and ability to achieve leading performance across a wide array of machine learning tasks. It represents a refined and optimized implementation of the gradient boosting algorithm, incorporating advanced techniques to tackle the common challenges of overfitting and computational inefficiency. XGBoost is particularly noted for its application of ensemble learning, where it sequentially constructs multiple decision trees, with each new tree correcting the errors of its predecessors. This approach is underpinned by a robust model update rule as presented in equation 7.

$$\widehat{y_i^{(t)}} = \widehat{y_i^{(t-1)}} + \eta \cdot f_t(x_i) \tag{7}$$

In this equation, $(\widehat{y_i^{(t)}})$ signifies the prediction for instance (i) at iteration (t), (η) is the learning rate that controls the step size in the direction of the gradient, and $(f_t(x_i))$ represents the output of the new tree at iteration (t). This iterative process of refinement, where predictions are successively improved by adding new trees, is central to XGBoost's effectiveness. The algorithm distinguishes itself with several key optimization strategies that enhance its performance. Among these is the Gradient-based One-Side Sampling (GOSS), which selectively focuses on instances with larger gradients, effectively reducing the data size without substantially altering the distribution.

Additionally, XGBoost's ability to efficiently process missing values and exploit sparse data structures through its sparsity-aware split finding technique significantly accelerates computation. Moreover, it incorporates both L1 and L2 regularization terms in the objective function, adding a level of regularization that mitigates the risk of overfitting.

2.11. LightGBM

LightGBM, standing for Light Gradient Boosting Machine, marks a significant advancement in the field of machine learning by providing a fast, distributed, and high-performance gradient boosting framework based on decision tree algorithms. Engineered to excel in speed and efficiency, LightGBM is particularly adept at handling large-scale data, making it a preferred choice for many data scientists facing the challenge of processing vast datasets. At the heart of LightGBM's approach is the principle of gradient boosting, where the model iteratively refines its predictions through a series of decision trees, each correcting the predecessor's mistakes. The core mechanism that facilitates this process can be succinctly captured by the model update equation as presented in equation 8.

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F \tag{8}$$

Here, $(\hat{y}_i)$ represents the prediction for the (i)-th instance, (K) denotes the total number of trees, $(f_k(x_i))$ is the prediction contribution from the (k)-th tree, and (F) encompasses the ensemble of all decision trees constructed during the learning process. This equation encapsulates the additive strategy employed by LightGBM, where the collective predictions of the ensemble yield the final output. LightGBM distinguishes itself through innovative techniques that enhance both its efficiency and effectiveness. One of its notable features is the Gradient-based One-Side Sampling (GOSS), which prioritizes instances with larger gradients for training, thereby reducing the number of data points needed without compromising the learning accuracy. Additionally, LightGBM employs the Exclusive Feature Bundling (EFB) technique, which aggregates features with minimal overlap into bundles, significantly reducing the dimensionality of the data and, consequently, the computational complexity.

Another pivotal aspect of LightGBM is its use of the leaf-wise growth strategy for trees, as opposed to the more traditional level-wise growth. This approach allows LightGBM to achieve lower loss compared to level-wise growth, albeit at the potential cost of increased complexity. However, LightGBM mitigates this by incorporating mechanisms to control over-fitting, thereby ensuring that the model remains robust even as it leverages the efficiency of the leaf-wise strategy. In practice, LightGBM has demonstrated remarkable success in a myriad of applications, ranging from risk management and fraud detection to web search ranking and real-time bidding for online advertising. Its ability to deliver high performance on large datasets, coupled with its lower memory usage and faster execution than many of its counterparts, positions LightGBM as a leading solution in the arsenal of modern data science.

2.12. CatBoost

CatBoost, an acronym for *Categorical Boosting*, is a state-of-the-art machine learning algorithm that belongs to the family of gradient boosting decision trees (GBDT). Developed by Yandex, CatBoost is specifically optimized to deal with categorical variables directly, which sets it apart from other gradient boosting frameworks that typically require extensive preprocessing to convert categorical variables into numerical format. CatBoost enhances the gradient boosting technique by introducing several innovative features designed to increase model accuracy while reducing the tendency for overfitting. One of the core components of CatBoost is its novel approach to processing categorical data. Unlike traditional methods that use one-hot encoding or label encoding, CatBoost utilizes an ordered boosting scheme that is less prone to overfitting and does not require extensive data preprocessing. The model update rule in CatBoost is articulated through equation 9.

$$\hat{y}_i = \sum_{t=1}^{T} \gamma_t h_t(x_i) \tag{9}$$

Where $(\hat{y}_i)$ represents the predicted value for the (i)-th instance, (T) is the total number of trees, (γ_t) is the learning rate for the (t)-th tree, and $(h_t(x_i))$ signifies the output of the (t)-th tree for the (i)-th instance. This equation encapsulates the essence of CatBoost, where the model iteratively refines its predictions by summing the contributions of each tree, adjusted by the corresponding learning rate. Furthermore, CatBoost employs a sophisticated algorithm for calculating feature importances, enabling users to understand the contribution of each feature to the model's decision-making process. This aspect is particularly useful in applications where interpretability is as crucial as predictive performance.

2.13. AdaBoost

AdaBoost stands as a cornerstone in the field of ensemble learning, where the core principle revolves around the sequential combination of multiple weak learners to form a robust, strong learner. Introduced in the 1990s, AdaBoost has since been instrumental in addressing classification problems across various domains due to its simplicity, efficiency, and effectiveness. The operational mechanism of AdaBoost is fundamentally iterative, where at each step, it focuses on the instances that were incorrectly classified by the previous models, applying higher weights to these instances. Consequently, subsequent learners are forced to concentrate on the harder cases in the dataset, thereby adaptively improving the model's overall accuracy.The aggregation of these weak learners is mathematically represented in equation 10.

$$haty(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right) \tag{10}$$

In this equation, $(\hat{y}(x))$ denotes the final prediction made by the ensemble, based on the weighted sum of the predictions $((h_t(x)))$ made by each weak learner. The weight $((\alpha_t))$ assigned to each learner's prediction is indicative of its accuracy, with more accurate learners receiving higher weights. The term (T) represents the total number of weak learners in the ensemble. A distinctive feature of AdaBoost is its adaptive learning of both the ensemble weights $((\alpha_t))$ and the data instance weights, which together ensure that subsequent learners focus more on the examples that previous learners misclassified. This adaptive process continues until a predefined number of iterations is reached, or the model achieves a desired level of accuracy. AdaBoost's effectiveness has been demonstrated in a wide range of applications, from binary classification tasks to multi-class problems. Despite its simplicity, AdaBoost can achieve high accuracy, often comparable to more complex models, making it a valuable tool for both theoretical and practical applications in machine learning.

2.14. HistGradient Boosting

HistGradientBoosting represents a modern take on gradient boosting, introducing an efficient strategy to handle continuous variables through the use of histograms. By discretizing the feature space into finite intervals, HistGradientBoosting reduces the complexity and computational overhead associated with finding the optimal split points in the decision trees, which is a fundamental component of the gradient boosting algorithm. The core idea behind this method is to approximate the continuous feature values with a fixed number of bins, thereby transforming the data into a more manageable form without significantly sacrificing model performance. This histogram-based technique not only speeds up the training process but also reduces memory usage, making it particularly advantageous for large datasets. The model's prediction for an instance is determined according to equation 11.

$$\hat{y}_i = \text{argmax}_k \left(\sum_{t=1}^{T} h_{t,k}(x_i) \right) \tag{11}$$

In this formulation, $(\hat{y}_i)$ signifies the predicted output for the (i)-th instance, derived from aggregating the contributions $\left((h_{t,k}(x_i))\right)$ of each tree (t) towards class (k). The function (argmax_k) selects the class with the highest cumulative score across all trees, thus determining the final prediction. HistGradientBoosting also incorporates several advanced features typical of gradient boosting algorithms, such as handling missing values and supporting categorical features directly, further enhancing its applicability and ease of use. The methodology offers a compelling balance between speed and accuracy, making it an attractive choice for a wide range of predictive modeling tasks. Its capability to efficiently process large volumes of data without extensive preprocessing or feature engineering underscores its practical value in real-world applications.

2.15. ExtraTrees

The Extra Trees algorithm, or Extremely Randomized Trees, offers a unique approach to ensemble learning by creating a collection of decision or regression trees. Unlike other ensemble methods that apply careful optimization at each split, Extra Trees introduces a higher degree of randomness in the way splits are chosen, aiming to reduce variance while maintaining a low bias. In the construction of the trees, rather than searching for the most discriminative thresholds, splits are determined by randomly selecting cut-points and choosing the best one among them. This strategy not only speeds up the training process significantly compared to more conventional methods like Random Forests but also helps in achieving a more diversified ensemble, which can be beneficial for reducing overfitting.

The ensemble's prediction for a given input is aggregated from the outputs of all individual trees, as denoted by equation 12.

$$\hat{y} = \frac{1}{B} \sum_{b=1}^{B} h_b(x) \tag{12}$$

Here, $(\hat{y})$ represents the ensemble's prediction, $(h_b(x))$ is the prediction made by the (b)-th tree, and (B) signifies the total number of trees in the ensemble. For classification problems, the final prediction is typically the class that receives the majority vote from all trees, whereas for regression problems, it is the average of the outputs from all trees.

2.16. Gaussian Naïve Bayes

Gaussian Naive Bayes classifier stands as a pillar among probabilistic classifiers, particularly distinguished for its application to continuous data. Underpinning this approach is the assumption that the continuous values associated with each class are distributed according to a Gaussian distribution. This

assumption enables the model to apply the Naive Bayes theorem to real-valued features by estimating the parameters of the Gaussian distribution (mean and variance) for each class. The foundation of Gaussian Naive Bayes lies in Bayes' theorem, which describes the probability of a class given set of features, mathematically represented as equation 13.

$$P(c|x) = \frac{P(c)P(x|c)}{P(x)} \tag{13}$$

In this equation, $(P(c|x))$ denotes the posterior probability of class (c) given predictors (x), $(P(c))$ is the prior probability of class (c), $(P(x|c))$ represents the likelihood which is the probability of predictor given class, and $(P(x))$ is the prior probability of the predictor. For Gaussian Naive Bayes, the likelihood $(P(x|c))$ is computed assuming a Gaussian distribution for the feature values in each class as presented in equation 14.

$$P(x_i|c) = \frac{1}{\sqrt{2\pi\sigma_c^2}} \exp\left(-\frac{(x_i - \mu_c)^2}{2\sigma_c^2}\right) \tag{14}$$

Where (x_i) is a feature, (μ_c) and (σ_c^2) are the mean and variance of the feature for class (c), respectively. This probabilistic framework enables the Gaussian Naive Bayes model to make predictions by calculating the class that maximizes the posterior probability given the feature values of a data point. Despite its simplicity, the Gaussian Naive Bayes classifier has shown remarkable effectiveness in various applications, including spam detection, document classification, and medical diagnosis. Its efficiency, stemming from the straightforward computation of class probabilities, and its robust performance, even with the assumption of feature independence, make it a valuable tool for tasks requiring quick and reliable classification of continuous data.

2.17. XGBoost RF

XGBoost RF marries the strengths of XGBoost, a leading gradient boosting framework, with the ensemble philosophy of Random Forest, creating a hybrid model that capitalizes on the advantages of both approaches. By incorporating the Random Forest methodology of building a multitude of decision trees and averaging their predictions, XGBoost RF aims to boost predictive accuracy and robustness while mitigating the risk of overfitting—a challenge often encountered in standard gradient boosting methods. The predictive mechanism of XGBoost RF adheres to the principle of ensemble learning, where the collective wisdom of multiple models is harnessed to make more accurate predictions than any single model could on its own. This is mathematically represented by equation 15.

$$\hat{y} = \frac{1}{B}\sum_{b=1}^{B} f_b(x) \tag{15}$$

Here, $(\hat{y})$ denotes the aggregated prediction for a given input (x), (B) signifies the total number of trees in the ensemble, and $(f_b(x))$ represents the prediction output by the (b)-th XGBoost tree. This formulation emphasizes the averaging process across all trees, which is central to the Random Forest strategy, thereby ensuring that the final prediction benefits from the diverse perspectives of multiple models.

In the context of XGBoost RF, each tree is constructed in a slightly different manner compared to traditional Random Forests, employing the sophisticated optimization and regularization techniques that XGBoost is known for. This includes the use of gradient-based optimization to fit trees and the incorporation of regularization terms to control model complexity, which altogether enhance the model's generalization capabilities.

3. RESULTS AND DISCUSSION

As presented in the table 1, the ensemble and boosting-based models, including LightGBM, AdaBoost, CatBoost, XGBoost, Random Forest, Decision Tree, HistGradientBoosting, and XGBoostRF, exhibited remarkable performance, with mean precision, recall, and F1 scores nearing or surpassing the 0.99 mark. Notably, LightGBM, AdaBoost, and CatBoost demonstrated exceptional accuracy and consistency in prediction, as indicated by their high mean scores and low standard deviations across the board. LightGBM stood out with mean scores of precision, recall, and F1 all at 0.999335 and with remarkably low standard deviations, suggesting very consistent performance across different folds in the data. This consistency indicates robustness in the model's ability to generalize well to unseen data, which is critical for reliable predictions in varying real-world conditions.

AdaBoost and CatBoost also performed exceptionally well, closely trailing LightGBM in terms of their mean scores. However, AdaBoost showed a slightly higher variance in recall, evidenced by its standard deviation, which suggests a minor fluctuation in performance across different validation folds compared to LightGBM and CatBoost. Conversely, the Gaussian Naive Bayes and Logistic Regression models showed substantially lower performance metrics, with F1 scores of 0.697408 and 0.748153 respectively. The considerably lower mean precision and recall scores for Gaussian NB highlight its challenges in correctly identifying positive instances and differentiating between classes, which is further corroborated by its higher standard deviations for these metrics. Logistic Regression, while performing better than Gaussian NB, still falls short of the ensemble and boosting methods, indicating that it might not capture the complexities of the data as effectively.

Table 1. Results of Comparison of Machine Learning Models.

Methods	Accuracy Mean	Precision Mean	Recall Mean	F1 Score Mean	Precision Std	Recall Std	F1 Std
Decision Tree	0.99	0.998677	0.998443	0.998557	0.003297	0.002	0.001984
Random Forest	0.99	0.998459	0.998889	0.998667	0.003414	0.0033	0.002211
Hist Gradient Boosting	0.99	0.998233	0.998889	0.998556	0.002754	0.002676	0.001652
CatBoost	0.99	0.998242	0.999333	0.998782	0.004022	0.002	0.002124
AdaBoost	0.99	0.998239	1	0.999115	0.003655	0	0.001837
LightGBM	0.99	0.999335	0.999333	0.999333	0.001418	0.002	0.001135
XGBoost	0.99	0.998454	0.998888	0.998667	0.001418	0.002	0.001135
Extra Trees	0.99	0.995586	0.995548	0.995553	0.0055	0.00498	0.00368
Logistic Regression	0.795	0.795036	0.789602	0.748153	0.133	0.241349	0.151231
XGBoost RF	0.99	0.997581	0.998221	0.997892	0.004534	0.002777	0.002342
Gaussian NB	0.544	0.544920	0.975968	0.697408	0.057874	0.017586	0.046012

Furthermore, the superior performance of ensemble methods can be attributed to their ability to aggregate decisions from multiple models or iterations, thereby reducing variance and bias. Boosting methods, which iteratively learn from the misclassifications of previous models, seem particularly adept at handling the intricacies of exoplanet classification, as evidenced by the high F1 scores. The gradient boosting machines, such as XGBoost, LightGBM, and CatBoost, utilize sophisticated algorithms to optimize both the loss function and the model complexity, leading to their outstanding performance. The low variability in performance of these models, as indicated by the low standard deviations, confirms their robustness. These models not only achieved high accuracy but also maintained this level of performance across different subsets of data, which is essential for a model that would be deployed in a dynamic and complex astronomical environment.

The relatively poor performance of Gaussian Naive Bayes could be due to its assumption of independence between features, which is often violated in complex datasets like those involving exoplanet characteristics. Logistic Regression's underperformance might be related to its linear nature, which could limit its ability to capture more complex, non-linear relationships in the data. The findings suggest that for tasks involving high-dimensional and complex datasets like exoplanet classification, ensemble and boosting-based machine learning models are more suitable due to their higher predictive power and stability. However, it is crucial to note that these results are contingent upon the quality and nature of the dataset. As the dataset consists of carefully preprocessed and feature-selected inputs, the performance of these models is optimal within the scope of the current study.

4. CONCLUSION

In conclusion, our research has presented a comprehensive evaluation of machine learning models for the classification of exoplanet candidates from the Kepler Space Telescope dataset. The analysis revealed that advanced ensemble and boosting models, namely LightGBM, AdaBoost, CatBoost, and XGBoost, significantly outperformed simpler models like Gaussian Naive Bayes and Logistic Regression. The high precision, recall, and F1 scores close to unity indicate these models' exceptional ability to accurately classify potential exoplanets with minimal error. LightGBM emerged as the leading model, demonstrating unparalleled precision and consistency across different folds of data. This consistency is indicative of the model's robustness and its capability to generalize well to new, unseen datasets, which is critical for the dynamic field of astronomy. AdaBoost and CatBoost also exhibited stellar performance, confirming the efficacy of boosting methods in dealing with complex datasets that require the model to learn from iterative corrections.

The study's findings endorse the use of ensemble and boosting-based machine learning models in astrophysical applications, especially for tasks involving intricate datasets like those of exoplanets. These

models' ability to discern subtle patterns and nuances in the data makes them invaluable tools for advancing our understanding of the universe and potentially identifying habitable worlds beyond our solar system. However, the lower performance of Gaussian Naive Bayes and Logistic Regression models serves as a reminder of the limitations inherent in simpler algorithms when handling high-dimensional and complex datasets. This underscores the importance of model selection in machine learning applications within the astronomical domain.

While the results are promising, we recognize the need for cautious optimism. The high performance metrics could, in part, be a result of the meticulous preprocessing and feature selection processes employed in this study. Therefore, continued validation with new datasets and in varying conditions is essential to ensure the models' resilience and reliability in real-world scenarios. Future work will also benefit from exploring the impact of new variables, refining feature engineering techniques, and experimenting with hybrid models that could offer even greater insights and predictive accuracy. Ultimately, our research contributes to the ongoing quest for understanding exoplanetary systems and supports the broader scientific community's efforts to harness the power of machine learning in the field of astronomy.

REFERENCES

[1] X. Fan, *Consciousness, Life and the Universe*. Taylor & Francis, 2024.

[2] A. A. Sweet, C. F. Sweet, and F. Jaensch, *THE UNITY OF TRUTH*. Xlibris Corporation, 2024.

[3] A. Loeb, *Extraterrestrial: The first sign of intelligent life beyond earth*. Houghton Mifflin, 2021.

[4] J. Bennett, S. Shostak, N. Schneider, and M. MacGregor, *Life in the Universe*. Princeton University Press, 2022.

[5] C. Impey, "Life beyond Earth: How will it first be detected?," *Acta Astronaut.*, vol. 197, pp. 387–398, 2022.

[6] C. Impey, *Worlds Without End: Exoplanets, Habitability, and the Future of Humanity*. MIT Press, 2023.

[7] I. Vavilova, L. Pakuliak, I. Babyk, A. Elyiv, D. Dobrycheva, and O. Melnyk, "Surveys, catalogues, databases, and archives of astronomical data," in *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, Elsevier, 2020, pp. 57–102.

[8] S. Sen, S. Agarwal, P. Chakraborty, and K. P. Singh, "Astronomical big data processing using machine learning: A comprehensive review," *Exp. Astron.*, vol. 53, no. 1, pp. 1–43, 2022.

[9] S. Baxter, "The Visibility of Big History," in *Expanding Worldviews: Astrobiology, Big History and Cosmic Perspectives*, Springer, 2021, pp. 91–106.

[10] J. N. Winn, *The Little Book of Exoplanets*. Princeton University Press, 2023.

[11] M. J. Smith, "Using deep learning to explore ultra-large scale astronomical datasets," 2022.

[12] S. Abimannan, E.-S. M. El-Alfy, Y.-S. Chang, S. Hussain, S. Shukla, and D. Satheesh, "Ensemble multifeatured deep learning models and applications: A survey," *IEEE Access*, 2023.

[13] M. H. Mobarak *et al.*, "Scope of machine learning in materials research—A review," *Appl. Surf. Sci. Adv.*, vol. 18, p. 100523, 2023.

[14] A. Vultureanu-Albi\csi and C. B\uadic\ua, "Improving students' performance by interpretable explanations using ensemble tree-based approaches," in *2021 IEEE 15th International Symposium on Applied Computational Intelligence and Informatics (SACI)*, 2021, pp. 215–220.

[15] J. G. Ponsam, S. V. J. B. Gracia, G. Geetha, S. Karpaselvi, and K. Nimala, "Credit Risk Analysis using LightGBM and a comparative study of popular algorithms," in *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, 2021, pp. 634–641.

[16] M. D. Agnew, H. Pettifor, and C. Wilson, "Lifestyle, an integrative concept: Cross-disciplinary insights for low-carbon research," *Wiley Interdiscip. Rev. Energy Environ.*, vol. 12, no. 6, p. e490, 2023.

[17] S. J. Pyne, *The Pyrocene: How we created an age of fire, and what happens next*. Univ of California Press, 2021.

[18] S. I. Silva, "Searching for New Worlds in an Era of Massive Data Sets: Planetary Transits, Gravitational Microlensing, and Neural Networks," The Catholic University of America, 2023.

[19] E. A. Brettschneider, "Exploring the Ethics of Human Space Travel: Navigating the Challenges and Implications of Missions Past, Present, and Future," 2023.

[20] T. Patil, G. Patil, and S. Arora, "AI-Powered Expedition: Navigating the Cosmos for Habitable Planets through Advanced ML Techniques," 2023.

[21] A. Kamp, "Navigating the landscape of higher engineering education," *education*, vol. 2, p. 115ce70ecb98, 2020.

[22] S. Bialek, "Skyward AI: Advancing Astronomy with Intelligent Machines," 2023.

[23] L. Y. Temple and others, "Confirmation of exoplanet candidates in the Wide-Angle Search for Planets survey via Doppler tomography," Keele University, 2020.

[24] A. M. Teachey, *On the Detection and Characterization of Exomoons Through Survey and Targeted Observations*. Columbia University, 2020.

[25] M. A. Barstow *et al.*, "The search for living worlds and the connection to our cosmic origins," *Exp. Astron.*, pp. 1–32, 2021.
[26] P. Jha, D. Dembla, and W. Dubey, "Deep learning models for enhancing potato leaf disease prediction: Implementation of transfer learning based stacking ensemble model," *Multimed. Tools Appl.*, pp. 1–20, 2023.
[27] A. Alyaseen *et al.*, "Assessing the compressive and splitting tensile strength of self-compacting recycled coarse aggregate concrete using machine learning and statistical techniques," *Mater. Today Commun.*, vol. 38, p. 107970, 2024.
[28] S. Aigrain and D. Foreman-Mackey, "Gaussian process regression for astronomical time series," *Annu. Rev. Astron. Astrophys.*, vol. 61, pp. 329–371, 2023.
[29] S. K. Meher and G. Panda, "Deep learning in astronomy: a tutorial perspective," *Eur. Phys. J. Spec. Top.*, vol. 230, pp. 2285–2317, 2021.
[30] NASA, "Kepler Exoplanet Search Results." 2017.