International Journal of Scientific Research in Science and Technology
Available online at : www.ijsrst.com
OPEN ACCESS
Print ISSN: 2395-6011 | Online ISSN: 2395-602X
doi : https://doi.org/10.32628/IJSRST2411426

Exoplanet Detection Using Machine Learning : A Comparative Study Using Kepler Mission Data

Pruthviraj Sunil Rajput
Symbiosis International School, Pune, Maharashtra, India

ARTICLEINFO

Article History:
Accepted : 20 Aug 2024
Published : 06 Sep 2024

Publication Issue :
Volume 11, Issue 5
Sep-Oct-2024

Page Number :
43-48

ABSTRACT

The search for habitable planets outside our solar system has captivated scientists throughout the centuries. Discovery and characterization of exoplanets have been one of the most important endeavors of modern astronomy. With various space missions, we have significantly expanded our observational capacity, resulting in an abundance of information about the universe. The influx of more data necessitates the development of techniques that can aid astronomers in processing all the information more efficiently and in an automated manner. Machine learning in recent years has become an indispensable paradigm to automate complex tasks that are possible only by humans. This work explores the application of machine learning to detect exoplanets from NASA's Kepler mission. Our dataset comprises Kepler Objects of Interest (KOIs), encompassing their characteristic features and confirmed exoplanet status. We experiment with multiple supervised classification techniques including classical, tree-based, and neural methods. The best-performing model Histogram Gradient Boosting achieves a strong performance of 94.6% precision and 94.1% recall on a held-out dataset demonstrating the strong potential of integrating machine learning techniques into astronomy, potentially leading to new insights into planetary systems outside the solar system.

Keywords : Exoplanet Detection, Supervised Classification, Machine Learning

I. INTRODUCTION

An exoplanet is a planet that orbits a star outside of our solar system. Exoplanet discovery has been an important endeavour for astronomers [1] across generations. The first two of the exoplanets were discovered in the 1992, and since then a total of around 5000 have been discovered. Moreover, our exoplanets detection systems also have evolved since then leading to a greater chance of finding more exoplanets. Exoplanet detection helps us to understand the different star systems present in the universe and how they behave. Furthermore, there's a slight chance of discovering an exoplanet with

habitable conditions required for the survival of humanity or even finding an alien civilization.

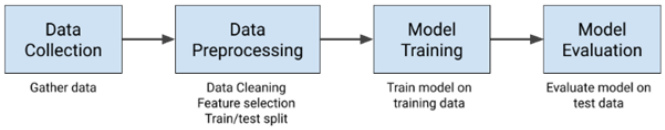

Figure 1: Pipeline of our approach

The search for exoplanets has evolved rapidly in the last few decades. To aid the earth telescopes the National Aeronautics and Space Administration (NASA) has launched telescopes into space. One such telescope dedicated to exoplanet discovery i.e. Kepler telescope was launched in 2009. Apart from the traditional signals such as the wobble in the light frequencies caused by orbiting planets, the Kepler telescope provided more powerful signals such as the distortion in stellar brightness caused by the eclipsing of orbiting planets, etc. From all the data the astronomers carefully analyse the light curves, consider various parameters, and determine exoplanets while ruling out false positives resulting from events such as eclipsing binary stars or instrumental artifacts. With more telescopes being launched and more information being gathered, the volume of data is increasing tremendously. While exciting this also poses a challenge for astronomers to sift through such a large volume of information. There has been a significant interest in developing automated technology to aid the astronomers. Machine learning (ML) provides a promising solution to this challenge. Machine learning algorithms can help scientists by automating many steps of the process. This will allow the astronomers to focus their efforts on the most promising candidates with more extensive analysis. The work shows the potential of machine learning models to automate the initial screening process. In this paper, we propose a machine learning algorithm that can take as input data from the telescope such as, the light curve parameters and the stellar parameters, and learn a decision system from labelled data. We experiment with multiple classification models on the data retrieved from NASA exoplanet archive.

The remainder of the paper is organized as follows. Section II introduces a few works related to ours, Section III provides details on the dataset, the data preparation strategy, the models we experiment with, and the evaluation metrics we use. Section IV provides the results of our experimentation and discusses the implications. We conclude with Section V summarizing our findings and suggesting directions for future research.

II. RELATED WORKS

Malik et al. [2] uses machine learning methods to detect exoplanets. They take a different approach to using the transit method for detection. They analyse the light curves using a time series library and extract features from the curve. The works most similar to ours are [3,4] which uses machine learning to determine exoplanets from Kepler data. They use various ML classifiers for their model. Unlike ours, [3,4] uses the analysis from the astronomers on false positives. For example, 'koi_fpflag_co' denotes whether the source of the signal is from a nearby star and is determined by astronomers. We on the other hand do not use those columns and only use the light curve, transit properties, and stellar parameters to detect exoplanets.

III. METHODS AND MATERIAL

For our experiments we use the pipeline shown in Fig.

1. We first gather our data in the first stage. The data is cleaned and prepared for the next stage which involves training the machine learning models. In the final stage, we evaluate the performance of our model on unseen data held out for evaluation purposes. We discuss each stage in the following.

A. Dataset

In 2009 NASA launched the Kepler space telescope with the goal of discovering Earth-sized exoplanets hoping to find places promising for life [5]. We make use of the data collected by Kepler.

1) Data preparation: We retrieve our dataset from the NASA exoplanet archive [6] on 2nd August 2024. Specifically, we download the "KOI Table (Cumulative list)" from their data page. The table contains 9564 rows of various extraterrestrial objects. The dataset contains 141 columns of features.

TABLE I
THE FINAL LIST OF COLUMNS USED BY OUR MODEL

Model	Accuracy
Exoplanet Information	koi_disposition
Transit Properties	koi_period, koi_period_err1, koi_period_err2, koi_time0bk, koi_time0bk_err1, koi_time0bk_err2, koi_time0, koi_time0_err1, koi_time0_err2, koi_eccen, koi_eccen_err1, koi_eccen_err2, koi_longp, koi_longp_err1, koi_longp_err2, koi_impact, koi_impact_err1, koi_impact_err2, koi_duration, koi_duration_err1, koi_duration_err2, koi_ingress, koi_ingress_err1, koi_ingress_err2, koi_depth, koi_depth_err1, koi_depth_err2, koi_ror, koi_ror_err1, koi_ror_err2, koi_srho, koi_srho_err1, koi_srho_err2, koi_prad, koi_prad_err1, koi_prad_err2, koi_sma, koi_sma_err1, koi_sma_err2, koi_incl, koi_incl_err1, koi_incl_err2, koi_teq, koi_teq_err1, koi_teq_err2, koi_insol, koi_dor, koi_dor_err1, koi_dor_err2, koi_ldm_coeff1, koi_ldm_coeff2, koi_ldm_coeff3, koi_ldm_coeff4,
Threshold Crossing Event (TCE) Information	koi_model_snr, koi_count, koi_tce_plnt_num, koi_model_dof, koi_model_chisq,
Stellar Parameters	koi_steff, koi_steff_err1, koi_steff_err2, koi_slogg, koi_slogg_err1, koi_slogg_err2, koi_smet, koi_smet_err1, koi_smet_err2, koi_srad, koi_srad_err1, koi_srad_err2, koi_smass, koi_smass_err1, koi_smass_err2, koi_sage, koi_sage_err1, koi_sage_err2

2) Data preprocessing: We perform comprehensive data cleaning and preprocessing to use for our model. Among the 141 columns as input features, we keep the column related to transit properties, threshold crossing event (TCE) information, and stellar parameters for input features. These columns provide physical information about the planets and hence enables us in creating an ML model that can predict exoplanets from physical properties. In Table 1 we show the columns we used for our model. If any column has empty values in the entire table, we remove that column. Similarly, columns with constant values are also removed. For all the uncertainties values we fill the missing values with 0. We didn't find any other column having missing or NaN values. 'koi_disposition' is used as ground truth labels. The 'koi_disposition' column contains three values 'CONFIRMED', 'FALSE POSITIVE', and 'CANDIDATE'. 'CONFIRMED' constitutes our positive exoplanet examples, 'FALSE POSITIVE' is our negative example. We remove the rows with the value 'CANDIDATE' as they do not provide any signal. After the data cleaning, we are left with 7099 rows from the original 9564 rows and 54 columns (including ground truth label) from the original 141 columns.

3) Dataset split: For model training and evaluation, we split the 7099 rows randomly into train and test datasets. We set aside 30% of the rows for evaluation. After the train-test split, the training dataset contains 4969 rows, and the test dataset contains 2130 rows.

B. Methods

We experiment with multiple classification models with varying degrees of complexity. We consider both classical and neural network-based machine learning models. Following paragraphs provide a brief description of the models used.

1) Naive Bayes: Naive Bayes is a probabilistic classifier that makes a strong assumption of feature independence and applies Bayes' theorem. Despite the strong assumption, the method has been shown to perform well for high-dimensional spaces.

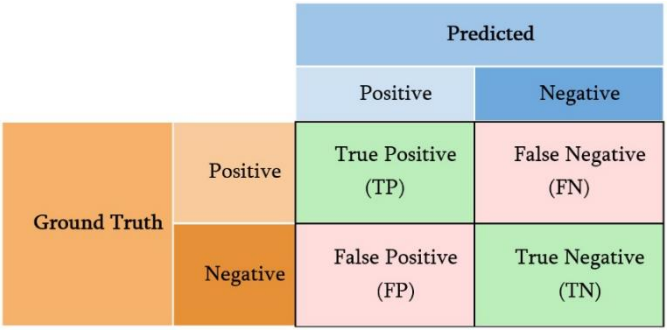

Figure 2 : Definition of True Positive, False Positive, True Negative and False Negative

2) Decision Tree: Decision Trees are a non-parametric method. The method works by recursively splitting the feature space on a threshold derived from feature values. A distinct advantage of the decision tree is the

decision prediction is visually interpretable. The decision tree generally performs poorly as they do not generalize. Extension of decision trees using ensembles has been proposed to tackle that.

3) Logistic Regression: Logistic Regression is a binary classifier that estimates the log-odds probability using a linear model. It is ubiquitously used for its simplicity. In real-world applications, it performs quite well if the decision boundary is linear.

4) Perceptron: The Perceptron is the earliest and simplest neural network-based linear classifier. It is simple in design that updates its weights with backpropagation and gradient descent.

5) Multilayer Perceptron (MLP): Multilayer Perceptron (MLP) is a neural network model that stacks multiple layers of Perceptron units separated by non-linear layers to learn a nonlinear decision boundary. During training, similar to Perceptron it uses backpropagation to calculate the gradients for each weight and updates it during the gradient descent step.

6) Histogram Gradient Boosting: Histogram Gradient Boosting is an ensemble of decision trees that is built as a sequence of decision trees where each tree reduces the error after the previous tree's result. The continuous features are discretised using histograms to increase efficiency. Using an ensemble helps with learning complex decisions from large-scale data and reduces overfitting problems.

C. Evaluation metrics

We use the metrics ubiquitous in classification literature i.e. accuracy, precision, recall, F1 score, and Precision-Recall curve. We use the F1-score to decide the best-performing model.

1) Accuracy, Precision, Recall, F1 score: These metrics are derived from True Positive, True Negative, False Positive, and False Negative as defined in Fig. 2. The definition of the metrics are as follows

$$\text{Accuracy} = \frac{correct\ classification}{total\ classification} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{correctly\ classified\ as\ positive}{all\ positive\ classification\ prediction} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{correctly\ classified\ as\ positive}{all\ ground\ truth\ positives} = \frac{TP}{TP + FN}$$

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

Gradient Boosting model stands out as the top performer, achieving the highest scores in accuracy (95.5%), precision (94.6%), F1 score (94.4%). This model's strong and balanced performance across these metrics highlights its effectiveness in accurately identifying exoplanets while minimizing false positives.

TABLE III
QUANTITATIVE PERFORMANCE

Model	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Naive Bayes	81.0	68.1	**98.0**	80.4
Decision Tree	90.7	89.5	86.7	88.1
Perceptron	92.1	88.6	91.8	90.2
Logistic Regression	93.0	91.1	91.2	91.2
Multilayer Perceptron (MLP)	95.2	92.9	95.0	94.0
Histogram Gradient Boosting	**95.5**	**94.6**	94.1	**94.4**

2) Precision-recall curve: Precision-recall curve is a graphical metric used to evaluate the performance of a classification model. It is particularly useful when there is a class imbalance in the dataset. It plots precision values in the y-axis and recall values in the x-axis for different classification thresholds. Average precision (AP) gives us a single score to represent a PR curve. It is defined as the weighted mean of precision at each threshold. The increase in recall from the previous threshold constitutes the weights.

IV. RESULTS AND DISCUSSION

A. Experimental setting

We conduct our experiments using NumPy [7] and Scikit-learn [8] libraries. For logistic regression, histogram gradient boosting, and perceptron we set the number of iterations to 50000. For MLP we use a batch size of 32, 2 hidden layers of size 5 each, and 1000000 iterations. The hyperparameters are validated using a held-out set from the training dataset. We set the random state to 42 for all our experiments.

B. Quantitative results

Our evaluations of various ML models are presented in Table II. The result reveals notable differences in their performance across several metrics. The Histogram The Multilayer Perceptron (MLP) also performed well, particularly excelling in recall (95.0%) and showing robust results in other metrics. Conversely, the Naive Bayes model, while demonstrating the highest recall (98.0%), exhibited a lower precision (68.1%), indicating a higher rate of false positives. Simpler models, such as Decision Tree and Perceptron, showed competitive results but were ultimately surpassed by the more sophisticated algorithms. Logistic Regression also performed well with an F1 score of 91.2%. These findings underscore the superior performance of ensemble and neural network-based approaches in capturing the intricate patterns within exoplanet data, while also illustrating the value of simpler models.

C. Precision-recall curve

The precision-recall curve comparison provides further evaluation of the performance of the four best- performing machine learning models used: Logistic Regression, Naive Bayes, Multilayer Perceptron (MLP), and Histogram Gradient Boosting (HGBoost). Fig. 3 illustrates the relationship between precision and recall across varying classification thresholds for each model. Histogram Gradient Boosting stands out with the highest average precision (AP) of 0.98, maintaining strong precision across a broad range of recall values. Logistic Regression closely follows with an AP of 0.97, showing comparable performance characteristics. The MLP model achieves an AP of 0.96 but exhibits greater variability in precision at lower recall levels compared to Histogram Gradient Boosting and Logistic Regression. On the other hand, the Naive Bayes model significantly lags with an AP of 0.70, demonstrating a constant precision across all recall levels and highlighting its limitations in balancing precision and recall effectively for this task. This visualization demonstrates the superior performance of ensemble methods (Histogram Gradient Boosting) and more sophisticated models (Logistic Regression, MLP) over simpler approaches like Naive Bayes in the context of exoplanet detection, highlighting the critical role of model choice in optimizing performance.

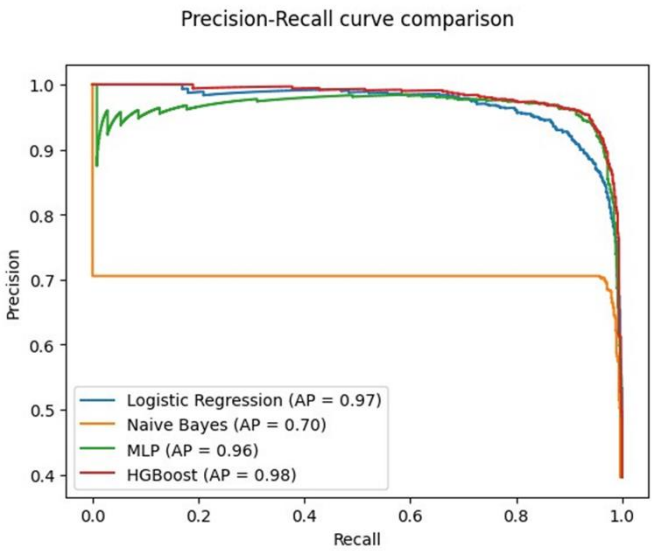

Figure 3: Precision-Recall Curve
(HGBoost = Histogram Gradient Boosting, MLP=Multi-layer Perceptron, AP=Average Precision

V. CONCLUSION

In this work, we present a machine learning-based approach that classifies exoplanets from Kepler cumulative object of interest data obtained from

NASA. Our work uses a comprehensive data preparation and filtering stage followed by experimentation with multiple state-of-the-art classification models. We achieve a performance of 94.6% precision with a recall of 94.1% which solidifies the potential of machine learning to automate exoplanet detection. Notwithstanding the promising results, we have investigated only Kepler object of interest data which contains exoplanets from a limited area of the universe and may not be representative of the diversity of exoplanets from the entire universe. A future avenue can investigate incorporating data from other space missions and ground observatories. More data also unlocks the possibility of using more complex machine learning models such as transformer models [9] which can take advantage of self-attention to discover complex interactions between the input features.

In conclusion, we show the tremendous potential of machine learning in aiding astronomers in exoplanet detection by automating the detection process. They can be a powerful tool for this task with their high performance, thus accelerating the process of understanding exoplanets and by extension the long-standing question of whether we are alone in the universe.

VI. REFERENCES

[1]. Brennan, Pat (2019). “Why Do Scientists Search for Exoplanets? Here Are 7 Reasons”. NASA Website. Online. Retrieved from https://exoplanets.nasa.gov/news/1610/why-do-scientists-search-forexoplanets-here-are-7-reasons/.

[2]. Abhishek Malik, Benjamin P Moster, Christian Obermeier, Exoplanet detection using machine learning, Monthly Notices of the Royal Astronomical Society, Volume 513, Issue 4, July 2022, Pages 5505–5516, https://doi.org/10.1093/mnras/stab3692

[3]. Sturrock, George Clayton; Manry, Brychan; and Rafiqi, Sohail (2019) "Machine Learning Pipeline for Exoplanet Classification," SMU Data Science Review: Vol. 2: No. 1, Article 9. Available at: https://scholar.smu.edu/datasciencereview/vol2/iss 1/9

[4]. Jin, Yucheng, Lanyi Yang, and Chia-En Chiang. "Identifying exoplanets with machine learning methods: a preliminary study." arXiv preprint arXiv:2204.00721 (2022).

[5]. “Kepler/K2”. NASA Official Website. Online. Retrieved from https://astrobiology.nasa.gov/missions/kepler/.

[6]. “Exoplanet Archive”. NASA Official Website.Online. Retrieved from https://exoplanetarchive.ipac.caltech.edu/docs/data.html.

[7]. Harris, Charles R., K. Jarrod Millman, Stéfan J. Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser et al. "Array programming with NumPy." Nature 585, no. 7825 (2020): 357-362.

[8]. Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel et al. "Scikit-learn: Machine learning in Python." the Journal of machine Learning research 12 (2011): 2825-2830.

[9]. Vaswani, Ashish, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser and Illia Polosukhin. “Attention is All you Need.” Neural Information Processing Systems (2017).