Exoplanet Detection with Decision Trees

By Sriram Loganathan
Affiliation: Cupertino High School

Abstract

Exoplanets can be detected through the observations of brightness and movement of the stars they orbit. In the past, machine learning algorithms have been able to classify possible candidates using specific techniques such as analyzing large samples of data and automating the otherwise tedious process of classification. In our research, we train a decision tree algorithm on datasets containing confirmed exoplanets, candidates, and false positives from the Kepler Mission in the NASA Exoplanet Archive. From this training, we build a decision tree classification model with a 94.12% accuracy at classifying exoplanets when training on confirmed exoplanets, candidates, and false positives, and a 99.78% accuracy when training only on confirmed exoplanets and false positives. Alternatively, when training a decision tree regression model to predict Kepler Object of Interest KOI) scores, we obtain a loss of 0.04. The decision tree algorithm is a viable option in classifying and detecting exoplanets, as displayed by its effectiveness.

Introduction

Machine learning has a large potential in revolutionizing exoplanet detection; it could possibly be the first step to discovering alien life in our universe. As the search for such life continues, the integration of machine learning into astronomy serves to increase our capabilities by offering a new path to navigate data. The decision tree algorithm can provide astronomers with a more systematic approach to analyze data and extract relevant key details to identify potential exoplanet candidates. The objective of this paper is to demonstrate the effectiveness of machine learning and decision tree models as valuable tools, utilizing their ability to analyze extensive datasets, detect patterns, and make predictions, to enhance the discovery of exoplanets.

Exoplanets, also known as extrasolar planets, are planets located outside our solar system. The proposal of exoplanets has a rich history that dates back to 400 BCE. The concept was first proposed by the Greek philosopher Democritus, who suggested that there might be worlds similar to ours in the universe[1]. However, it wasn't until the 20th century that technology was advanced enough to make such statements. In 1992, astronomers at the Arecibo Observatory in Puerto Rico made a groundbreaking discovery, detecting and confirming the existence of exoplanets[2]. This discovery provided direct proof that planets could exist outside of our solar system and opened new opportunities for research and advancement.

In the following years, the field of exoplanets experienced a surge of advancements and discoveries. The development of more sophisticated telescopes, space missions, and analytical techniques enabled astronomers to detect exoplanets with increasing precision and to uncover a wider range of planetary systems. Most notably, the Kepler space telescope, launched in 2009 by NASA, played a pivotal role in this progress. Kepler's mission was committed to observing an extensive number of stars in a fixed field of view, monitoring their brightness for subtle dips caused by transiting exoplanets[3]. Prolonged observations on the same field allowed Kepler to capture multiple exoplanet transits, especially those with longer orbital periods. Its highly sensitive detectors and sophisticated data processing algorithms ensured reliable candidates. And with such precision, Kepler led to the identification of thousands of exoplanet candidates.

As astronomers continue to refine their methods and explore newer opportunities, the use of machine learning in exoplanet research holds the key to uncovering more profound insights into the diversity of exoplanetary systems, and their habitability potential, hidden within the collected data. This application represents a large stride towards uncovering new worlds, their enigmatic compositions, and potentially answering one of humanity's most profound questions: Are we alone in the universe?

Exoplanet Detection Methods

Exploring the universe to identify exoplanets involves many methods, each serving to detect distinct signatures left by these celestial objects, spanning from direct imaging to the observation of light curves. In this section, we examine the techniques used by astronomers to

discover exoplanets and learn more about their properties. From the radial velocity method, which detects the wobble induced by orbiting exoplanets, to the transit method, which observes the periodic dimming of a star caused by a passing planet, we unravel the physics behind exoplanet detection. Additionally, we examine the phenomenon of gravitational microlensing, where the gravity of a star or planet acts as a cosmic lens, magnifying the light of background objects. By understanding these methods, we lay the foundation for comprehending the data that machine learning algorithms will process, opening the door to a deeper exploration of exoplanetary systems.

At the forefront of exoplanet detection lies the radial velocity method, which unveils the otherwise nearly hidden movement between stars and their orbiting planets. This method converts light into data, revealing the presence and properties of exoplanets through the electromagnetic spectrum. When an exoplanet orbits its host star, both celestial bodies are bound by mutual gravitational forces. As the planet exerts its subtle tug, the star reciprocates with a slight shift in its movement, oscillating around their common center of mass. This leaves behind a Doppler shift to the star's emitted light[4]. The formula for radial velocity using the Doppler Shift effect is given by:

$$\frac{\Delta\lambda}{\lambda} = \frac{v}{c},$$

where

$\Delta\lambda$ represents the change in wavelength of the star's spectrum,
λ is the original wavelength of the star's light,
v is the radial velocity of the star caused by the presence of an exoplanet, and
c is the speed of light.

When light is analyzed, it appears as a spectrum of lines, each corresponding to specific wavelengths. The Doppler effect, a fundamental principle in the field of optics in physics, is introduced. As a star wobbles due to the gravitational influence of an exoplanet, its light shifts either toward the blue or red end of the spectrum. This phenomenon, known as blueshifting and

redshifting, signifies motion approaching or receding, respectively[5]. These shifts convey a wealth of information, enabling astronomers to determine the velocity and other details of a star's motion.

A method that has been more recently popularized is the transit method. This method involves observing the change in brightness of a star as an exoplanet transits in front of it[6]. The depth of this transit (ΔF) can be described using this equation:

$$\Delta F = (R_{planet}/R_{star})^2,$$

where

R_{planet} is the radius of the exoplanet and

R_{star} is the radius of the star

As the exoplanet orbits its star, these recurring transits generate a series of fluctuations in the star's luminosity, forming a light curve. A light curve is a set of data that can hold the information of an exoplanet such as its orbital period and distance from its star[7].

Decision Trees

In the discovery of exoplanets, machine learning emerged as a compass, helping astronomers process and analyze massive amounts of data. These algorithmic techniques methodically dissect data, mimicking behavior similar to that of a human.

A decision tree is a supervised machine learning algorithm. Beginning at the root node, the algorithm selects the most informative feature and splits the data into subsets. This process is repeated recursively for each resulting subset, creating a branching structure that resembles a tree[8]. The algorithm determines the optimal splitting points using criteria such as the cross-entropy loss for classification tasks[9], given by

$$H(P,Q) = -\sum_{x \in X} P(x) \log(Q(x))$$

where

$H(P,Q)$ represents the cross-entropy between two probability distributions, P and Q,

$P(x)$ represents the probability of an event x occurring according to the distribution P,

$Q(x)$ represents the probability of the same event x occurring according to the distribution Q.

$\sum_{x \in X}$ indicates that the calculation is performed over all possible events x in the set X.

or mean squared error for regression tasks[11], given by:

$$\text{MSE} = \frac{1}{|D|} \sum_{i=1}^{|D|} (y_i - \hat{y}_i)^2$$

where

$|D|$ denotes the number of samples in the dataset D

y_i represents the actual target value (also called the ground truth) for the ith sample in the dataset D.

$\hat{y}_i$ represents the corresponding predicted value to y_i, given the current state of the model. (We convert words or string values into numeric values using the one-hot encoding method[10])

As the tree grows, it captures hierarchical relationships within the data, with each internal node representing a decision based on a specific feature, and each leaf node containing the final prediction or decision. To make predictions for new data, the algorithm follows the decision

path from the root to a leaf, based on the feature values, and outputs the corresponding prediction or class label.

The interpretability of decision trees is one of their major strengths. The decision path from the root to a leaf node represents a clear and easily understandable set of rules, making it straightforward to interpret and explain the reasoning behind predictions. However, this transparency can also lead to overfitting, where the tree captures noise in the training data[8]. To mitigate this, various strategies are employed, such as limiting the depth of the tree, setting a minimum number of samples required for a node to be split further, and pruning branches that contribute little to predictive accuracy.

Gradient Boosting Classifier/Regressor

Gradient boosting is a powerful and widely-used ensemble learning technique for both classification and regression tasks. It combines the predictive power of multiple weak learners, usually decision trees, to create a strong and accurate model. The core principle behind gradient boosting is the iterative construction of a sequence of models, each correcting the errors of the previous ones. The process begins by creating a simple base model, typically a shallow decision tree, which serves as the initial prediction. Subsequent models are then built in a stepwise manner, where each new model focuses on reducing the errors made by the combined predictions of the existing models[12].

The key innovation of gradient boosting lies in its optimization strategy. It minimizes a specific loss function, which quantifies the difference between the predicted values of the combined ensemble and the actual target values. Instead of just approximating the target directly, each subsequent tree aims to correct the errors made by the preceding trees[12]. In each iteration, the algorithm identifies the direction (gradient) in which the loss function has the steepest descent, and constructs a new model to fit this residual error which involves the formula given by

$$\text{minimize} \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + \gamma h_m(x_i))$$

where

L is the loss function such as the Mean Squared Error(MSE),

$F_{m-1}(x_i)$ represents the ensemble's predictions up to the $(m-1)$ iteration for the ith sample,
γ is the learning rate which determines the contribution of the new weak learner,
$h_m(x_i)$ is the prediction of the mth weak learner for the ith sample.

With each iteration, a new tree m is constructed, not with the objective of directly predicting the target, but to compute the residual or error of the preceding tree $m-1$'s prediction. This means that if the first model made specific prediction errors, the subsequent model aims to 'correct' these inaccuracies. The final prediction is obtained by summing up the predictions of all the individual models, weighted by a learning rate that determines the contribution of each model to the final result. This refinement and technique results in robust and accurate model making gradient boosting a powerful tool in machine learning.

Results

In our model training and evaluation utilizing the scikit-learn library[13], our objective is to optimize a GradientBoostingClassifier by accessing a range of random hyperparameter combinations for tuning. The hyperparameters considered include:

- N_estimators: The number of boosting stages to be run (number of trees used).
- learning_rate: The step size at each iteration while moving towards a minimum of the loss function.
- max_depth: The maximum depth of the individual estimators.
- max_features: The number of features to consider when looking for the best split.

The optimal hyperparameters are selected based on their performance on a validation set.

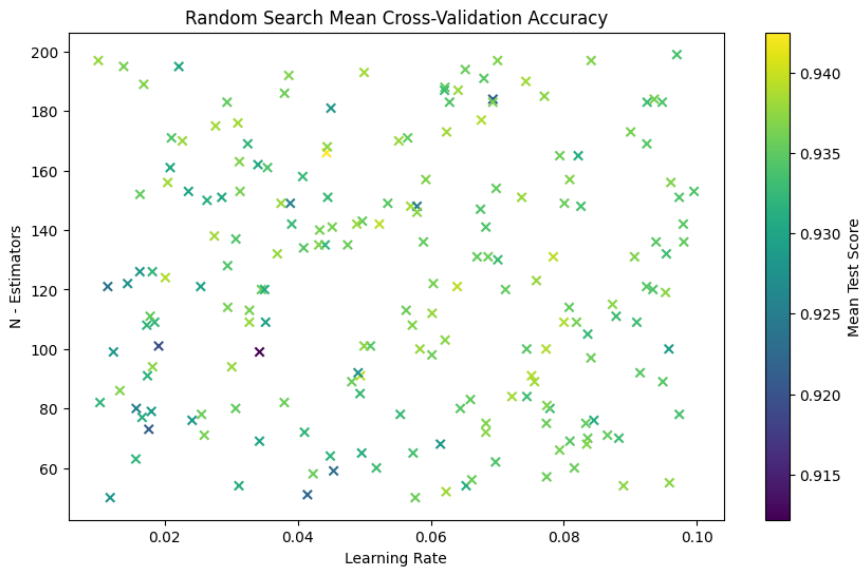

Fig 1: Random search mean cross-validation accuracy chart, varying the hyperparameters n-estimators (number of trees) and learning rate.

Our GradientBoostingClassifier achieves an accuracy of 94.12% on the testing dataset containing confirmed exoplanets, candidates, and false positives. Despite the low variance between the training and testing datasets, there is avoidable bias, in that exoplanet candidates are all either exoplanets or false positives in reality, but have just not been determined as either one yet. When training only on confirmed exoplanets and false positives, the accuracy is 99.78% .

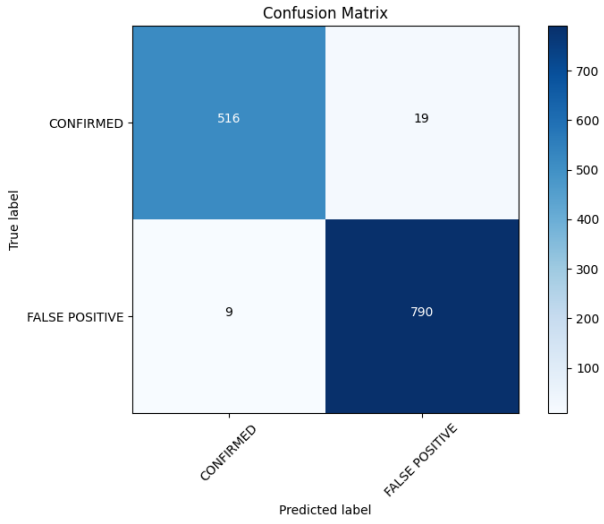

Fig 2: Confusion Matrix Distinguishing Confirmed Exoplanets From False Positives

In addition to classifying exoplanets and false positives, we seek to predict the Kepler Object of Interest (KOI) score. The KOI score, calculated from the Kepler space telescope's observations, quantifies the likelihood that a given celestial signal hints at the existence of an exoplanet. Scores range from 0 to 1, with values closer to 1 suggesting a higher probability that the observed signal corresponds to an exoplanet, while values closer to 0 indicate a higher likelihood of the signal being a false positive or stemming from other cosmic phenomena.

Given the continuous nature of the KOI score, which ranges between 0 and 1, a classification approach would be unsuitable. Instead, we opt for a regression tree, a machine learning model adept at predicting continuous outputs. Upon training our regression tree on the dataset, we achieved a Mean Squared Error (MSE) of 0.04. This means that, on average, our predictions deviate from the actual values by 0.2 units, representing a 20% error with respect to the full scale.

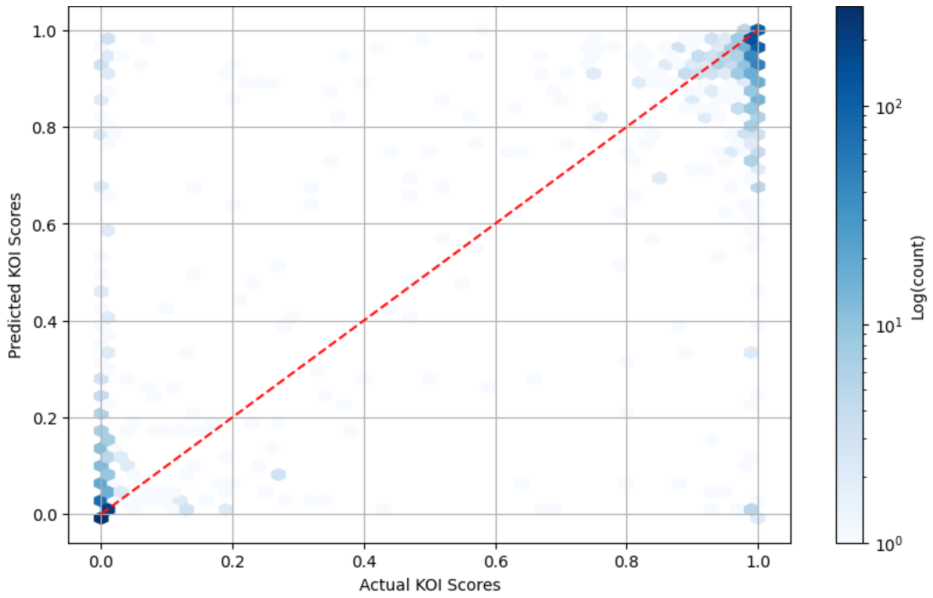

Fig 3: Hexbin Density Plot of Actual vs. Predicted KOI Scores

References

1. Gagnon, Jean, et al. “IAL 18: Exoplanets & General Planetary Systems.” UNLV Physics, 1997, https://www.physics.unlv.edu/~jeffery/astro/ial/ial_018.html.

2. Britannica, The Editors of Encyclopaedia. "Arecibo Observatory". Encyclopedia Britannica, 1 Aug. 2023, https://www.britannica.com/topic/Arecibo-Observatory.
3. Dooling, Dave. “Kepler.” *Encyclopædia Britannica*, Encyclopædia Britannica, inc., 2009, www.britannica.com/topic/Kepler-satellite.
4. Richmond, M. (2001). A connection between radial velocity and distance. http://spiff.rit.edu/classes/phys240/lectures/expand/expand.html
5. Dobrijevic, D., & Howell, E. (2022, January 14). Redshift and blueshift: What do they mean? https://www.space.com/25732-redshift-blueshift.html
6. Rauf, J. (2021). Looking for Exoplanets. https://www.uc.edu/content/dam/refresh/cont-ed-62/olli/21-fall/exoplanets4.pdf
7. Richmond, M. (2014). Important parameters of an eclipsing system. What can we learn from light curves? http://spiff.rit.edu/classes/phys373/lectures/light_curves/light_curves.html
8. Stanford Online. (2020, April 17). Lecture 10 - Decision Trees and Ensemble Methods | Stanford CS229: Machine Learning (Autumn 2018) [Video]. YouTube. https://www.youtube.com/watch?v=wr9gUr-eWdA
9. Koech, K. 2020, August 20. Cross-Entropy Loss Function. Towards Data Science. https://towardsdatascience.com/cross-entropy-loss-function-f38c4ec8643e
10. Yadav, D. 2019, December 9. Categorical encoding using Label-Encoding and One-Hot-Encoder. Towards Data Science. https://towardsdatascience.com/categorical-encoding-using-label-encoding-and-one-hot-encoder-911ef77fb5bd
11. 2008. Mean Squared Error. In: The Concise Encyclopedia of Statistics. Springer, New York, NY. https://doi.org/10.1007/978-0-387-32833-1_251
12. Kurama, V. 2020, March 29. Gradient Boosting for Classification. https://blog.paperspace.com/gradient-boosting-for-classification/
13. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.