Kepler Mission: Exoplanet Classification Using Machine Learning

Arop Kuol

24 November 2025

Abstract

This report details the implementation of a machine learning pipeline for classifying objects from the NASA Kepler Objects of Interest (KOI) dataset. The primary goal was to accurately distinguish between Confirmed Exoplanets, Candidate Exoplanets, and False Positives. We utilized several supervised learning models, including Logistic Regression, Decision Tree, Random Forest, and a Single Hidden Layer Neural Network. The **Random Forest** model achieved the highest performance, with an accuracy of **91.85%** and an F1-score of 0.8818. These results demonstrate the superior capability of ensemble methods in identifying non-linear patterns essential for exoplanet vetting.

Acknowledgements

We extend our deep gratitude to the NASA Exoplanet Archive and the Kepler Mission Team for providing the fundamental data for this analysis. This project was initiated through the direct influence and guidance provided by the educational content of **Dr. Thomas Albin's YouTube channel, Astroniz Albin** on Space Science and Python. Following months of studying and applying the techniques demonstrated there, the search for a suitable implementation project led to the discovery and replication of the robust classification methodology established by Monaco and Gervasoni Monaco [2020], which provided the foundational structure for this work.

1 Introduction

The discovery of planets orbiting stars beyond our Sun, known as **exoplanets**, has revolutionized astronomy and as of now, 6052 exoplanets has been discovered NASA Exoplanet Archive [a]. The NASA **Kepler Mission** was a pioneering space telescope launched in 2009 with the primary objective of surveying a massive region of the Milky Way galaxy to determine the frequency of Earth-size and smaller planets in the habitable zone of sun-like stars.

Kepler accomplished this by using the **transit method** of detection. This method measures the slight, periodic dimming of a star's light that occurs when an orbiting planet passes in front of it from our viewpoint. This temporary decrease in brightness is recorded as a transit signal.

These observed transit signals are initially cataloged as **Kepler Objects of Interest (KOIs)**. While the mission was wildly successful, many KOIs originate from astrophysical false positives (e.g., background eclipsing binary stars) or noise, meaning not every KOI is a true exoplanet. Therefore, a critical step in exoplanet research is **classification**: vetting these candidates to distinguish true **Confirmed Exoplanets** from strong **Candidates** and definite **False Positives**.

The goal of this report is to implement a comprehensive **machine learning pipeline** to automate this vetting process. By analyzing the physical properties of both the transit signal and the host star (e.g., period, radius, stellar temperature), we aim to build highly accurate classification models.

2 Related Work

This report replicates and evaluates the structure and methodology of published work McCauliff and Jenkins [2015] and a corresponding data mining project Monaco [2020], using the techniques of **Logistic Regression**,

Decision Trees, **Random Forest**, and a **Single Hidden Layer Neural Network**. The subsequent sections detail the data processing, model training, comparative performance, and interpretation of the final results.

3 Dataset Overview

The aim of our project is to train an algorithm that is capable to classify a celestial object as an exoplanet or not. The source of the dataset is the offical NASA Exoplanet Archive then selected the KOI cumulative list NASA Exoplanet Archive [b]. The dataset consists of 9564 detected cases, which include confirmed exoplanets, false positives or candidate but not confirmed objects.

3.1 Description

The analysis is based on a refined subset of the NASA Kepler Objects of Interest (KOI) dataset, comprising various physical and statistical attributes for each candidate. While identification columns (like `kepid` and `kepler_name`) are excluded from the modeling process, a core set of features related to the transit signal, stellar properties, and vetting statistics are retained.

The **target variable** for classification is `koi_disposition`, a categorical attribute taking the values **CONFIRMED**, **FALSE POSITIVE**, and **CANDIDATE**. This designation indicates the final status of the observed object after the full vetting process. The related variable, `koi_pdisposition`, represents the initial disposition determined solely using Kepler data.

Statistical indicators related to the vetting process include the `koi_score`, which quantifies the **level of confidence** in the KOI's disposition, ranging from 0 to 1. Additionally, several **binary flag attributes** are essential for explaining false positive scenarios:

- `koi_fpflag_nt`: Non-Transit-Like signal flag.
- `koi_fpflag_ss`: Stellar Eclipse False Positive flag.
- `koi_fpflag_co`: Centroid Offset False Positive flag (indicating the signal is offset from the target star).
- `koi_fpflag_ec`: Ephemeris Match Contamination flag.

These flags explain phenomena other than a planet transit that could have caused the stellar dimming.

Other vital attributes characterize the observed **transit event** and the **planetary candidate**:

- `koi_period`: The orbital period, or the time interval between transits.
- `koi_duration`: The total duration of the transit event.
- `koi_depth`: The fractional dimming of the star's light during transit.
- `koi_prad`: The calculated planetary radius (in Earth radii).
- `koi_teq`: The planet's equilibrium temperature (in Kelvin).
- `koi_model_snr`: The **Transit Signal-to-Noise Ratio** (SNR), measuring the strength of the detection.

Finally, the properties of the **host star** are fundamental for characterizing the candidate. These include `koi_steff` (stellar effective temperature), `koi_slogg` (surface gravity), and `koi_srad` (stellar radius). The celestial location and brightness are captured by `ra`, `dec`, and `koi_kepmag` IPAC Caltech.

4 Data Preprocessing and Engineering

The preprocessing phase was essential for preparing and organizing the raw data to ensure robust and efficient training of the machine learning algorithms.

4.1 Feature Selection and Data Cleaning

Initially, a rigorous **feature selection** was performed on the original 141 variables. We systematically removed columns deemed irrelevant to the classification task, including identifying attributes (e.g., `kepler_name` and `koi_tce_plnt.nmbr`), and all columns representing measurement errors, as these were considered non-predictive noise. This initial step yielded a streamlined dataset consisting of 22 relevant columns.

4.2 Handling Missing and Anomalous Data

Next, we addressed the problem of missing data. We first identified and removed approximately 364 observations that contained common missing values across multiple features. Following this step, the `koi_score` attribute still exhibited a significant number of missing entries (1206 observations). Lacking the necessary domain expertise to justify complex imputation methods that could arbitrarily alter the data distribution, we conservatively chose to **remove these remaining observations** to maintain data integrity, reducing the dataset to 7944 rows.

Finally, an **anomaly detection** step was executed. We identified one record where the binary feature `koi_fpflag_nt` had an invalid value of 465 (which should only be 0 or 1). Given the impossibility of confidently determining the correct flag value, this anomalous observation was removed, resulting in a final dataset size of 7943 rows.

4.3 Target Variable Binarization

As a final preprocessing step, we prepared the target variable from the nominal categorical feature `koi_disposition` by performing a **binary class binarization**. We consolidated the classes as follows:

- **Positive Class (Value 1):** All objects designated as **CONFIRMED** exoplanets.
- **Negative Class (Value 0):** All objects designated as **FALSE POSITIVE** or **CANDIDATE**.

This binarization converts the multi-class problem into a binary classification task focused on confirming true exoplanets.

5 Machine Learning Modeling

5.1 Training and Scaling Setup

The initiation of the modeling phase required careful preparation of the feature matrix and target variable. First, the full dataset was partitioned into the feature matrix (X), containing the 20 predictive attributes, and the target vector (y), containing the binarized `koi_disposition` status. Next, the data was divided into training and testing sets using a standard 75/25 split. Critically, the stratify=y parameter was applied to this split to ensure that the resultant training and testing sets maintain the same proportion of the minority and majority classes as the original dataset. This step is vital for avoiding bias, given the existing class imbalance in the exoplanet data. Finally, the feature matrices (`X_train` and `X_test`) were subjected to Standard Scaling. This process transforms the features to have a mean of zero and a unit variance, which is a critical step for algorithms sensitive to feature magnitude, particularly Logistic Regression and the Neural Network, promoting faster and more stable convergence.

5.2 Logistic Regression

The modeling phase commenced with the **Logistic Regression** classifier, serving as the essential linear benchmark against which all subsequent non-linear models would be evaluated. Initially, a baseline model was trained using all 20 available features after scaling.

Following the methodology of the reference paper, which employed the stepAIC procedure in R Monaco [2020] to select an optimal feature subset, we attempted to replicate this approach in Python. Standard `scikit-learn`'s Logistic Regression does not compute statistical quantities like the log-likelihood or AIC, making classical stepwise selection impossible.

An attempt to use `statsmodels` (which supports likelihood-based metrics) failed due to the numerical instability of Logistic Regression on this dataset. This instability is attributed to strong **multicollinearity** among Kepler stellar and planetary parameters, resulting in singular matrices.

To obtain a stable alternative, we utilized **L1-regularized (LASSO) Logistic Regression**, a standard modern technique for automatic feature selection. However, the LASSO model selected all available features, yielding a result identical to the initial baseline model.

Given the numerical challenges and the need to maintain consistency with the replication effort, the final Logistic Regression model was evaluated using a reduced set of 18 features, excluding `koi_insol` and `koi_fpflag_ec`, as dictated by the original stepAIC results reported in the reference paper.

The **plain Logistic Regression model (without class balancing)** emerged as the strongest linear performer. This model, or the nearly identical **L1 (LASSO)** model, serves as our robust **linear benchmark**.

5.3 Decision Tree

The **Decision Tree Classifier** was employed to confirm the presence of non-linear patterns within the data, which the Logistic Regression model could not capture, and to establish the interpretability necessary for feature selection.

Hyperparameter Tuning

We conducted a comprehensive hyperparameter search using `GridSearchCV` to identify the optimal configuration for the Decision Tree. The search space focused on controlling the tree's complexity and impurity measure:

- `max_depth`: Explored depths from 5 to 20, and the unlimited option (`None`).
- `min_samples_leaf`: Tested constraints from 1 to 20.
- `criterion`: Evaluated both `gini` and `entropy` for split quality.

The best model configuration, selected by maximizing the F1 score using 5-fold cross-validation, was found to be:

$$\text{Best Parameters:} \quad \begin{cases} \texttt{criterion} : \text{'entropy'}, \\ \texttt{max\_depth} : 5, \\ \texttt{min\_samples\_leaf} : 5 \end{cases}$$

This result indicates that a **shallow, constrained tree** (`max_depth=5`, `min_samples_leaf=5`) utilizing **Entropy** provided the best balance of performance and generalization, significantly improving upon the linear benchmark.

Feature Importance and Selection

A crucial advantage of the Decision Tree is its ability to quantify the relative predictive power of each input feature. This is measured by **Feature Importance**, specifically the Mean Decrease in Impurity (MDI) contributed by each variable across all splits in the tree.

The feature importance analysis identified 7 out of 20 variables as non-contributing ("not important"), having importance scores near zero. This justified a necessary step of **feature selection** to reduce noise, improve training efficiency, and potentially enhance the generalization capabilities of subsequent, more complex models.

Based on this analysis, the final feature set for subsequent modeling (Random Forest and Neural Network) was reduced to the following 13 high-impact features:

```
'koi_score', 'koi_fpflag_nt', 'koi_fpflag_ss', 'koi_fpflag_co', 'koi_period',
'koi_impact', 'koi_duration', 'koi_depth', 'koi_prad', 'koi_model_snr',
'koi_srad', 'ra', 'koi_kepmag'
```

5.4 Random Forest

Following the successful feature reduction achieved by the Decision Tree, the **Random Forest (RF) Classifier** was implemented. This ensemble method uses multiple decision trees to overcome the overfitting issues of a single tree, and was trained using the **reduced feature set**.

Hyperparameter Tuning

We used `GridSearchCV` to find the best settings for the ensemble. The search focused on:

- `n_estimators` (number of trees): Tested 100, 200, and 500.
- `max_features`: Tested 'sqrt' (4 features) and the integer 4.
- `max_depth` and `min_samples_leaf`: Constraints on tree growth.

Best Model

The best model configuration, selected by maximizing the F1 score, was found to be:

$$\text{Best Parameters:} \quad \begin{cases} \texttt{n\_estimators} : 200 \\ \texttt{max\_features} : \text{'sqrt'} \\ \texttt{max\_depth} : \text{None} \\ \texttt{min\_samples\_leaf} : 1 \end{cases}$$

This model uses 200 deep, unconstrained trees (`max_depth=None`) and randomly samples 4 features at each split (`max_features='sqrt'`), leading to a diverse and robust ensemble.

5.5 Single Hidden Layer Neural Network

The final model implemented was the **Single Hidden Layer Neural Network (SHLNN)**, a Multi-Layer Perceptron (MLP) Classifier. This non-linear model was tested to determine if a connectionist architecture could discover complex features that outperformed the best-performing Random Forest ensemble. The model was trained using the same **reduced feature set** and scaled data.

Hyperparameter Tuning

We used `GridSearchCV` to tune the critical architectural parameters of the network. The `adam` solver was utilized for optimization, and the maximum iterations (`max_iter`) was set high (5000) to ensure full convergence. The search focused on two primary factors:

- `hidden_layer_sizes`: Explored the optimal number of neurons in the single hidden layer, testing sizes (5,), (10,), (25,), (50,), and (100,).
- `learning_rate_init`: Tested initial learning rates of 0.001, 0.01, and 0.1.

Best Model

The cross-validation process, maximizing the F1 score, yielded a highly parsimonious (simple) network structure:

$$\text{Best Parameters:} \quad \begin{cases} \texttt{activation} : \text{'relu'} \\ \texttt{hidden\_layer\_sizes} : (10,) \\ \texttt{learning\_rate\_init} : 0.001 \end{cases}$$

The optimal model utilizes only **10 neurons** in the hidden layer and required a very **slow learning rate** (0.001), indicating the optimization landscape was sensitive to large updates. The use of the ReLU activation function is the industry standard for non-linear processing in the hidden layer.

6 Results and Performance Evaluation

6.1 Evaluation Metrics and Confusion Matrix

Model performance was assessed using the standard confusion matrix and derived metrics, which are particularly important in datasets exhibiting class imbalance. The models predict the Positive Class (1) as a CONFIRMED Exoplanet and the Negative Class (0) as either a Candidate or False Positive.

Confusion Matrix Fundamentals

The Confusion Matrix (Table 1) serves as a foundational tool for describing and comparing the performance of the classification models on the test data by cross-referencing predicted values against actual observed values.

Table 1: Structure of the Binary Confusion Matrix

	Actual: Negative (0)	**Actual: Positive (1)**
Predicted: Negative (0)	TN	FN
Predicted: Positive (1)	FP	TP

- **TP (True Positive):** Correctly classified as CONFIRMED Exoplanet.
- **TN (True Negative):** Correctly classified as non-Exoplanet (Candidate/False Positive).
- **FP (False Positive):** Incorrectly classified as CONFIRMED Exoplanet (a "false alarm").
- **FN (False Negative):** Incorrectly classified as non-Exoplanet (a "missed discovery").

Derived Measures

The following key indicators were derived from the confusion matrix values:

- **Accuracy (A):** The fraction of records rightly classified across all observations.

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

- **Precision (P):** The fraction of positive predictions that are correct. In this context, it measures the model's ability to minimize false alarms among all objects labeled as CONFIRMED.

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

- **Recall (R):** The fraction of actual positive records that were correctly identified. This measures the model's ability to avoid missing actual Exoplanets.

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

- **F1-Score (F_1):** The harmonic mean of Precision and Recall. The F_1-Score is the primary metric for model selection as it provides a single measure that balances the trade-off between the costs of false alarms (P) and missed discoveries (R).

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R}$$

6.2 Model Comparison

Now let's look at the model performance.

Table 2: Comparative Performance of Classification Models

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC Score
Logistic Regression	85.69	0.7176	0.9575	0.8204	0.9344
Decision Tree	90.45	0.8482	0.8768	0.8623	0.9570
Random Forest	**91.85**	**0.8723**	**0.8915**	**0.8818**	**0.9736**
Neural Network	89.94	0.8177	0.9076	0.8603	0.9637

7 Conclusion

This project successfully implemented a machine learning pipeline to classify Kepler Objects of Interest (KOIs), aiming to automate the vetting process for exoplanet discovery. The project successfully navigated challenges in data preparation, including feature selection based on Decision Tree importance and handling numerical instability during the Logistic Regression setup, while adhering to the structure of the reference work. Four distinct classification models were systematically trained and evaluated on the reduced feature set: Logistic Regression, Decision Tree, Random Forest, and a Single Hidden Layer Neural Network.

The Random Forest model emerged as the most effective performer. Its superior performance demonstrates that ensemble methods are highly effective for this type of astrophysical classification, successfully modeling complex, non-linear relationships in the data while maintaining stability. Conversely, the Logistic Regression model provided a stable linear benchmark, and the Neural Network's marginally higher Recall was ultimately outweighed by lower Precision and F1-score. In conclusion, the results confirm that the tuned Random Forest model provides a robust, accurate, and stable classification tool, validating the use of sophisticated machine learning techniques for prioritizing promising planetary discoveries in modern astrophysical research.

8 Future Work

While the Random Forest model achieved high accuracy, future research can enhance the pipeline's capabilities through several avenues. We propose exploring **Deep Learning Architectures**, such as Convolutional Neural Networks (CNNs), to analyze the raw light curve time-series data directly, bypassing the reliance on engineered features. Furthermore, integrating external data, such as precise stellar measurements from the **Gaia DR3 mission**, could refine stellar parameter estimates and improve vetting accuracy. Finally, experimenting with advanced sampling techniques, like **ADASYN**, could be conducted to determine if a different approach to handling the initial class imbalance might further reduce the False Negative rate. Also adding further domain knowledge enhances the scientific context and justification for your modeling choices.

References

T. Albin. Astroniz [youtube channel]. URL `https://www.youtube.com/c/Astroniz`. Accessed: 2024-12-05.

IPAC Caltech. Kepler Candidate Data Columns. `https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html`. Accessed: 2025-11-22.

S. D. McCauliff and J. M. Jenkins. Automatic classification of kepler planetary transit candidates. *The Astrophysical Journal*, 806(2):173, 2015. doi: 10.1088/0004-637X/806/2/173.

A. Monaco. DataMining: Kepler Exoplanet Classification [source code]. `https://github.com/AlbertoMonaco/DataMining`, 2020. Accessed: 2025-10-24.

NASA Exoplanet Archive. Exoplanets 101. https://exoplanets.nasa.gov/the-search-for-life/exoplanets-101/, a. Accessed: 2025-06-13.

NASA Exoplanet Archive. Kepler Cumulative KOI Dataset. https://exoplanetarchive.ipac.caltech.edu/, b. Accessed: 2025-11-22.