Project Report: A Machine Learning Model for Classifying Potential Exoplanets from Kepler Data

Abstract

For this data mining project, I have developed, tested, and evaluated a machine learning model to classify objects identified by the Kepler Space Telescope as either Confirmed Exoplanets or False Positives. The data for this project comes from the NASA Exoplanet Archive, which has tabulated thousands of Kepler Objects of Interest (KOI), each one described by dozens of features including properties of the stars and planets being observed as well as measurement parameters. The initial phase of this project involved much time spent on understanding the features of the data and identifying any irrelevant or redundant features. Exploratory data analysis was conducted to determine which features to include in the model and how to transform the data in the preprocessing stage. The final feature space turned out to be much smaller than initially expected. Multiple classifying methods were tested and optimized, including Decision Tree, Random Forest, Agglomerative Clustering, and a dense neural network. The three supervised learning methods achieved similar results of around 92% accuracy, while the unsupervised clustering method only managed 68% accuracy. The decision tree model accomplished the project goal of having an efficient and highly interpretable model for identifying the presence of exoplanets.

1. Introduction

Since its launch in 2009, NASA's Kepler Space Telescope (named for 17th century German astronomer Johannes Kepler) has dramatically improved our understanding of the universe through the discovery of thousands of exoplanets - planets orbiting stars beyond our solar system. Once thought to be rare, exoplanets are now understood to be extremely common, with the Milky Way galaxy likely containing more planets than stars [1]. The Kepler team discovers exoplanets by continuously monitoring thousands of stars and looking for stellar occultations - dips in brightness that indicate a possible object passing in front of the star. Some of these observations are not true exoplanets but are instead false positives with causes such as fluctuations in stellar activity and measurement noise.

Distinguishing genuine exoplanets from false positives is crucial for astronomers' ability to continue building accurate knowledge in this domain. This typically requires extensive, time-consuming follow-up observations of candidate objects - or KOI's. The role of machine learning is to accelerate the process by determining - based on initial observations - how likely a candidate is to be a true exoplanet, and therefore telling researchers which candidates to prioritize for subsequent observations. The NASA Exoplanet Archive has accumulated thousands of records of KOIs including both confirmed exoplanets and false positives [2]. Within this archive is the KOI Cumulative Table, containing dozens of features for each KOI entry along with a target label of CONFIRMED or FALSE POSITIVE. This project aims to take advantage of this data to build an efficient, automated, interpretable classifier that can be deployed for this purpose.

2. Related Work

Several past studies have applied statistical modeling and machine learning techniques to solving the problem of automated KOI classification. Morton et al. (2016) calculated false positive probabilities for KOIs using vespa, a publicly available Python package for statistical modeling [3].

Shallue & Vanderburg (2018) identified exoplanets with high accuracy (>98%) using Convolutional Neural Networks (CNNs) on light curves from Kepler data [4]. These approaches are computationally intensive and utilize time-series data rather than tabulated data. For my project, I aim to build a simpler, more interpretable classifier using only the KOI Cumulative Table. My goal is to have a model that not only produces accurate results, but provides an easily understood decision-making process.

Past studies have also identified some of the most common conditions leading to false positives. Based on work from the Kepler False Positive Working Group [5], four of these conditions are included in the KOI Cumulative table. One of these is a description of the event as “not transit-like”, indicating that it doesn’t have the characteristics of an object passing in front of a star. Another is “stellar eclipse”, meaning that another star in a binary orbit, not a planet, caused the observed change in brightness. Another flag is for “centroid offset”, which indicates that the change in brightness came from another nearby star and not from the star being observed. Finally, there is a flag for “ephemeris match indicates contamination”, which means that the KOI is judged to be the result of “flux contamination in the aperture or electronic crosstalk”. Understanding these false positive conditions and how they are correlated with parameters in the dataset has been very helpful in informing the feature selection process in the early stages of this project.

3. Project Work

3.1) Data Understanding

Understanding the definitions of the features of The KOI Cumulative Table and how they were derived proved to be a crucial element of the work done for this project. The raw data file contained more than 100 features, but many were redundant or irrelevant to the task of predicting exoplanet status. Importantly, I learned that some features were added to the table after the KOI’s status had already been verified: for example, false positive flags indicating the reason a KOI was classified as false positive, and a “disposition score” indicating the false positive probability given to the event while making the determination. Including these parameters in the model would defeat its purpose.

The NASA Exoplanet Archive website provided a convenient tool for selecting features before downloading the data into a CSV file. I first removed features that were irrelevant to the task, such as the ID of the KOI and the date of observation. There were also columns for which the data was entirely missing. I also removed features that were obviously calculated from other features in the table, such as the ratio of planetary radius to star radius while both radii were included in separate columns. When it came time to download the file, it contained 27 features. This included core measurement features such as the magnitude and duration of the occultation, stellar parameters such as the size and temperature of the star, planetary features, telescope settings, statistical features, and also metadata such as the number of planets previously discovered orbiting the star. After conducting a more thorough investigation of how each feature was derived, I found several more that were redundant. For example, the planetary radius was estimated from the solar radius and the transit depth (magnitude of drop in brightness), both of which are features in the data. In the end, the number of features narrowed to 17, which is much fewer than I expected at the start of the project. All these features are numerical, which is surprising, since I had expected to need to deal with categorical features. The full list of features is shown below:

- Orbital period - time between occultations
- Transit depth - magnitude of drop in brightness
- Transit duration - length of occultation
- Inclination - angle at which the object traverses the star
- Maximum Multi-Event Statistic (MES) - a measure of the statistical significance of repeated observations
- Transit signal-to-noise ratio (SNR)
- Number of planets discovered in this system
- Planet number for this KOI
- Odd-even depth comparison (OEDC) statistic - a measure of the difference between odd and even numbered crossings, intended to help distinguish planets from binary star systems
- Stellar temperature
- Stellar metallicity
- Stellar radius
- Stellar mass
- Telescope angles (2 parameters)
- Kepler-band - telescope brightness setting

3.2) Data Cleaning

After selecting the features, I imported the data to a Pandas dataframe in Python. A basic inspection of the data showed it contained 9564 rows and 18 columns - 17 features plus the target variable. With the value_counts() function I found the following counts for the target variable:

- FALSE POSITIVE 4839
- CONFIRMED 2746
- CANDIDATE 1979

Since I am training a classifier, only the entries that are confirmed exoplanets or false positives are useful. After filtering out the candidates, the dataset contained 7585 entries with a 36%-64% split between confirmed exoplanets and false positives.

The .info() function revealed that most of the columns had a small number of missing values. The feature with the most null values (910) was the OEDC statistic, and it also contained values marked -1 (regular values ranged from 0-1) to indicate that it couldn't be measured (in some cases because only one occultation had been observed). These were also treated as missing values. Before deciding how to handle the missing values, I conducted exploratory data analysis (EDA) by plotting histograms to examine the distribution of each variable. This revealed that many of the variables, such as transit depth and orbital period, had an exponential distribution, while others such as stellar mass had an approximately normal distribution. I chose to impute missing values using the median rather than the mean value to avoid changing the skew of the distributions.

As part of EDA, I also plotted a correlation matrix using the Seaborn library's heatmap() function. I was somewhat surprised at how little correlation was shown between the variables, as part of the initial plan for this project was to use correlations to further reduce the feature space. The map did show a strong correlation between the transit SNR and MES statistics and between solar mass and solar temperature, so perhaps only one variable from each of those pairs is needed. For now, though, I left all 17 features intact.

3.3) Data Preprocessing

The first machine learning model to be tested in this project was a decision tree classifier, and for this model no additional preprocessing was required. Decision trees set thresholds for features individually, so there is no requirement for features to be transformed or normalized to a similar scale. I also planned to test a clustering model, however, and for this normalization is very important. The first step was to use a log-transform method for features that

were exponentially distributed. The log values of these features resembled a normal distribution. Figure 1 below shows the raw distribution of the Transit Duration variable, while Figure 2 shows the log-value distribution:

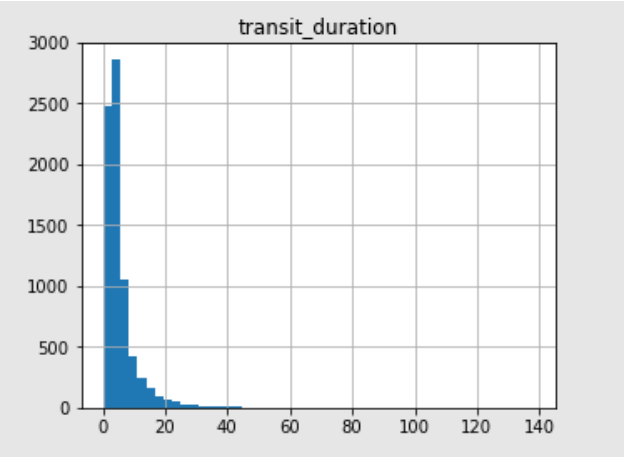

Fig. 1) Histogram of Transit Duration

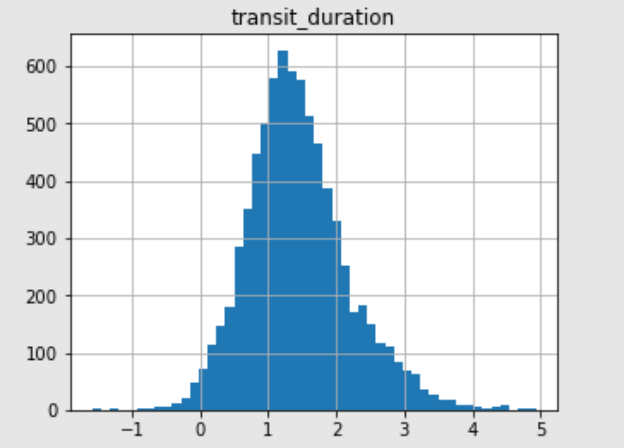

Fig. 2) Histogram of log(Transit Duration)

After applying the log transformations, the remaining normally-distributed variables were assigned values according to their Z-score (number of standard deviations from the mean). The final step in data preprocessing was to further transform the data using Principal Component Analysis (PCA), a dimensionality reduction technique that combines the variables into components that maximize the explained variance of the data. Seven components (out of 17 features) were enough to explain about 80% of the variance.

3.4) Data Modeling and Results

The two machine learning algorithms that I was most interested in testing for this project were Decision Tree and Agglomerative Clustering. Decision trees are a natural choice for a dataset containing a diverse set of features, and the main advantage is that they are highly interpretable. A user can follow the decision tree and gain an understanding of which features are most predictive.

I began the modeling stage by splitting the data into training and testing sets, with 80% of the data in the training set. The target variable was set to 1 for a confirmed exoplanet and 0 for a false positive. I constructed a baseline decision tree model using the SciKit-learn library's DecisionTreeClassifier() function with default parameters; i.e., no limits on number of features or number of leaves in the tree. The tree was constructed by minimizing the Gini index at each split [6]. This baseline model had testing accuracy of 88%, but precision (accuracy of predicted positives) and recall (percentage of true positives identified) were lower, at 81% and 83%, respectively, owing to the imbalance between positive and negative cases. To improve the model, I used SciKit-learn's GridSearchCV function with five cross-validation folds to test many different combinations of hyperparameters. This parameter sweep aimed to optimize ROC_AUC (area under the Receiver Operating Characteristic curve) to provide a better balance between accuracy, precision, and recall. The best performing decision tree minimized Entropy rather than Gini index, limited the depth of the tree to 10 levels, and set a minimal sample size of 4 data points per leaf to reduce overfitting. This improved accuracy, precision, and recall to 92%, 89%, and 91%, respectively. The difference between the baseline and optimized decision trees is shown by the confusion matrices below:

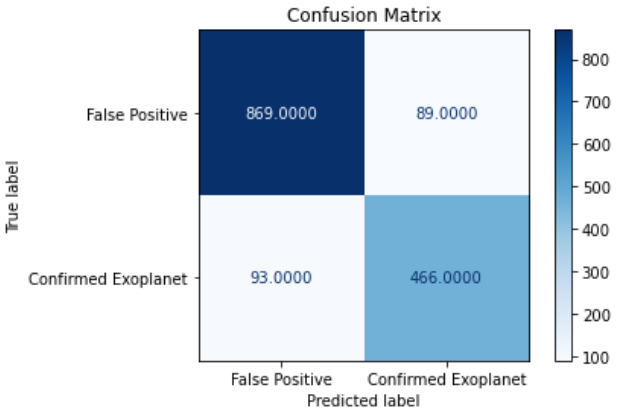

Fig. 3) Confusion matrix for baseline decision tree

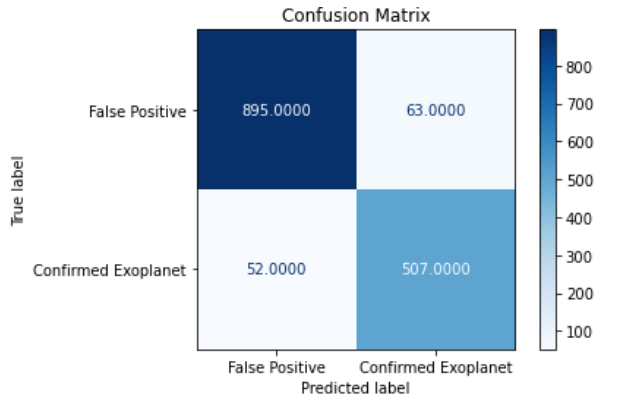

Fig. 4) Confusion matrix for optimized decision tree

While a single decision tree model is ideal for interpretability, I decided to try a Random Forest classifier - an ensemble method involving many decision trees - to see what - if any - improvement could be gained. A similar GridSearchCV procedure produced a Random Forest model with 200 decision trees that were each randomly assigned 4 variables. The testing metrics were only slightly better than those for the optimized decision tree.

The next model that I tested was an Agglomerative Clustering model. This is an unsupervised approach, meaning that the target variable was not used for training the model. Instead, the model sorts the data into two clusters based on the locations of data points within the featurespace. Agglomerative clustering starts with each data point assigned to its own cluster, and then the two nearest clusters are iteratively combined until only two clusters remain. This model used the PCA-transformed data, and I included the number of PCA components defining the featurespace as a hyperparameter to optimize along with metrics for calculating the distances between points and between clusters. The best-performing model of this type managed only 68% accuracy.

The final model tested for this project was a dense neural network classifier using TensorFlow. After some experimentation, I settled on an architecture with three dense layers containing 256, 128, and 64 neurons, followed by a dropout layer to reduce overfitting. Even with 100 epochs of training, the results were not significantly different from the decision tree model.

4. Evaluation

The results from the four classification models tested in this project are summarized in the table below:

Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	92.4%	88.9%	90.7%	0.898
Random Forest	92.9%	91.4%	89.3%	0.903
Agglomer-ative Clustering	67.8%	54.5%	67.0%	0.601
Dense Neural Network	92.5%	87.2%	93.2%	0.901

Fig 5) Model evaluation table

According to the four metrics listed in the results table, the three supervised learning models (Decision Tree, Random Forest, Neural Network) greatly out-performed the unsupervised model (Agglomerative Clustering). Within the three supervised learning models, a single optimized decision tree classifier performed just about

as well as the two more powerful, computation-intensive methods.

Qualitatively, the decision tree offers the best balance of accuracy and interpretability for the purpose of helping astronomers search for exoplanets. The tree allows users to follow along the decision-making process and develop insights into the most important parameters of the dataset. Below is a rendering of the structure of the decision tree developed in this project:

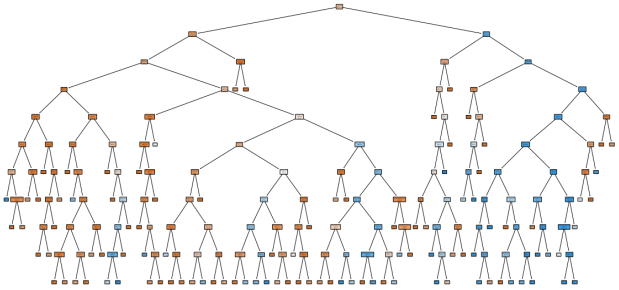

Fig 6) Visualization of the decision tree

5. Discussion

In the project proposal, I defined the following timeline:

Phase 1 - Data Cleaning and Understanding, including EDA Visualizations (2 days)

Phase 2 - Data Preprocessing and Feature Selection (1 day)

Phase 3 - Baseline Model Development (1 day)

Phase 4 - Model Evaluation and Hyperparameter Tuning (2 days)

Phase 5 - Final Presentation of Results (1 day)

In reflection, the data understanding phase was the most research-intensive and important phase of the project. Without taking the time to learn about each feature in the dataset and learn which ones were derived from other ones, I would have a dataset full of redundant features, which would pose problems for model training and interpretability. Instead, I had a relatively small set of features, which worked out well for producing an efficient and interpretable model. A larger featurespace likely would have needed Random Forest or a deep neural network model to get the optimal results, but for this dataset, these powerful methods did not lead to substantial improvement.

The failure of the unsupervised clustering model to produce comparable results to the supervised models is interesting. It suggests that confirmed exoplanets and false positives do not occupy clearly distinct regions of the feature space, and that more intricate reasoning is needed to separate them.

One lesson learned through the process of model development and evaluation was not to rely solely on accuracy as an evaluation metric. When there is a significant class imbalance in the target variable (36-64 split in this project's data), a classification model can achieve a good accuracy score by tilting its results toward one side, but this might not be the most useful outcome. For this project, it might be most important to have a high recall so that few true exoplanets are missed.

One interesting insight from analyzing the structure of the decision tree is that the most predictive variable - the root of the tree - is the number of planets in the star system being observed. This implies that if a certain star already has confirmed exoplanets in its orbit, a new KOI found at that star is also likely to be a planet. If a KOI is found at a star without any known exoplanets, it's likely to be a false positive. Perhaps previous discoveries at a star indicate that the star has the right conditions for finding more planets. A potential limitation of the model is that many of the confirmed exoplanets in the training data come from stars with multiple planets,

and therefore it might not generalize well when applied to data from newly-observed stars.

6. Conclusion

This data mining project has delivered on its goal of developing a machine learning model to classify Kepler Objects of Interest (KOIs) as either confirmed exoplanets or false positives using tabulated data from the NASA Exoplanet Archive. Prior research has shown that automated classification methods can accelerate the KOI vetting process, improving the pace of discovery and the understanding of our place in the universe. The efficient, highly interpretable decision tree classifier developed for this project complements the more computationally-intensive existing methods.

Potential future work for this project should involve developing a classifier that relies less on the number of planets variable. Perhaps a model could be optimized for the subset of the data for which the stars have no other known planets. Another area of future work should be trying to understand what went wrong with the clustering approach. It would be interesting to find out if any different methods of feature selection or data transformation could make this approach work, or if the two classes in this dataset are truly not separable by clustering.

References

1. Cassan, A., Kubas, D., Beaulieu, JP. *et al.* “One or more bound planets per Milky Way star from microlensing observations”. *Nature* 481, 167–169 (2012). https://doi.org/10.1038/nature10684

2. NASA Exoplanet Science Institutue. (2024). *Kepler Objects of Interest (KOI) cumulative table*. NASA Exoplanet Archive. California Institute of Technology. https://exoplanetarchive.ipac.caltech.edu/

3. Morton, T.D. et al. *False positive probabilities for all Kepler objects of interest.* (2016). The Astrophysical Journal. https://arxiv.org/abs/1605.02825

4. Shallue, Christopher J and Vanderburg, Andrew. *Identifying Exoplanets with Deep Learning.* (2018). The Astronomical Journal. https://arxiv.org/abs/1712.05044

5. Bryson, Stephen T. et al. “The Kepler Certified False Positive Table”. (2017). https://ui.adsabs.harvard.edu/abs/2017ksci.rept...12B/abstract

6. *scikit-learn Developers.* **DecisionTreeClassifier** — *scikit-learn 1.7.2 Documentation*. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html