Machine Learning and Neural Networks mid-term assignment report

Abstract

This project explores the application of k-Nearest Neighbors (kNN) and Decision Tree algorithms for the classification of exoplanets, aiming to identify which model provides more accurate predictions. By analyzing exoplanetary data to classify celestial bodies into the categories Gas Giants, Super Earths, Neptune-like and Terrestrial planets. The findings reveal significant insights into the strengths and limitations of each model, and propose which algorithm better suits the intricate nature of exoplanet classification.

Introduction

Machine learning plays a pivotal role in the astronomical classification of exoplanets. This study utilizes a dataset containing a variety of exoplanetary attributes to compare two distinct classification algorithms: k-Nearest Neighbors and Decision Trees. The k-nearest neighbours was implemented using libraries and the decision trees was self implemented.The project's goal is to determine which algorithm more effectively discerns between different types of exoplanets, thereby providing a robust tool for ammature astronomers in the field.

Background

Over 5,000 confirmed exoplanets paint a staggering picture of our universe's diversity. However, simply discovering these celestial bodies is just the first step. Unveiling their true nature, classifying them into categories remains a complex endeavor.

Machine learning offers a powerful tool for navigating the intricacies of exoplanet data. One such algorithm, k-Nearest Neighbors (kNN), approaches this challenge with surprising simplicity. Imagine a multidimensional space where each known planet occupies a unique location based on its various characteristics, such as mass, radius, and orbital elements. When a new planet is discovered, kNN analyzes its features and identifies the k closest "neighbors" in this space. These neighbors aren't physically near, but they share the most similar characteristics with the new planet.

Based on the types of its k closest neighbors, kNN then predicts the most likely category for the new planet. The underlying assumption is that planets with similar features likely belong to the same category. It's akin to grouping objects based on their shared properties, not their physical arrangement.

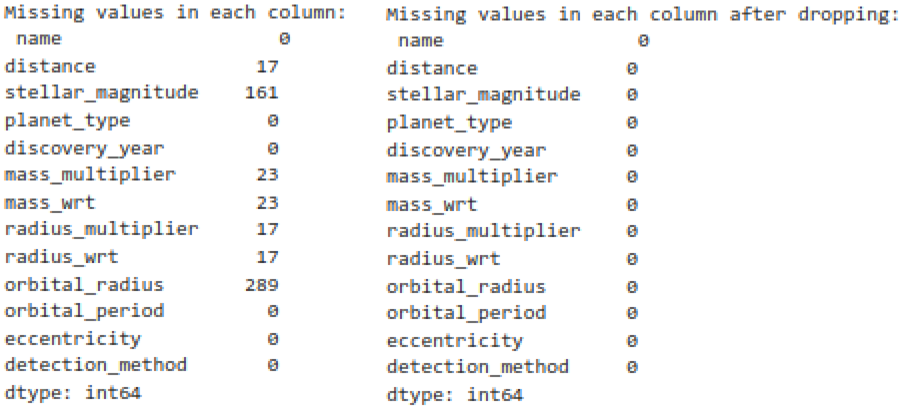
While kNN excels in its straightforward approach, it's essential to acknowledge its limitations. Exoplanet classification rarely has clear-cut solutions, and often we must rely on extrapolating the properties of familiar planets to understand newly discovered ones. This is where Decision Trees offer a complementary perspective.

Decision Trees construct a hierarchical structure of decisions based on exoplanet attributes. However, it's crucial to note that this process isn't arbitrary. Each split in the tree is guided by a quantifiable decision criterion, such as Gini impurity or information gain. These criteria measure the purity of resulting subsets, ensuring the tree's structure reflects statistical properties of the dataset rather than subjective choices.

By systematically evaluating these criteria, Decision Trees create a series of branching rules that ultimately lead to a classification. While not without limitations, they provide a structured approach to navigate uncertainties in exoplanet identification.

Methodology

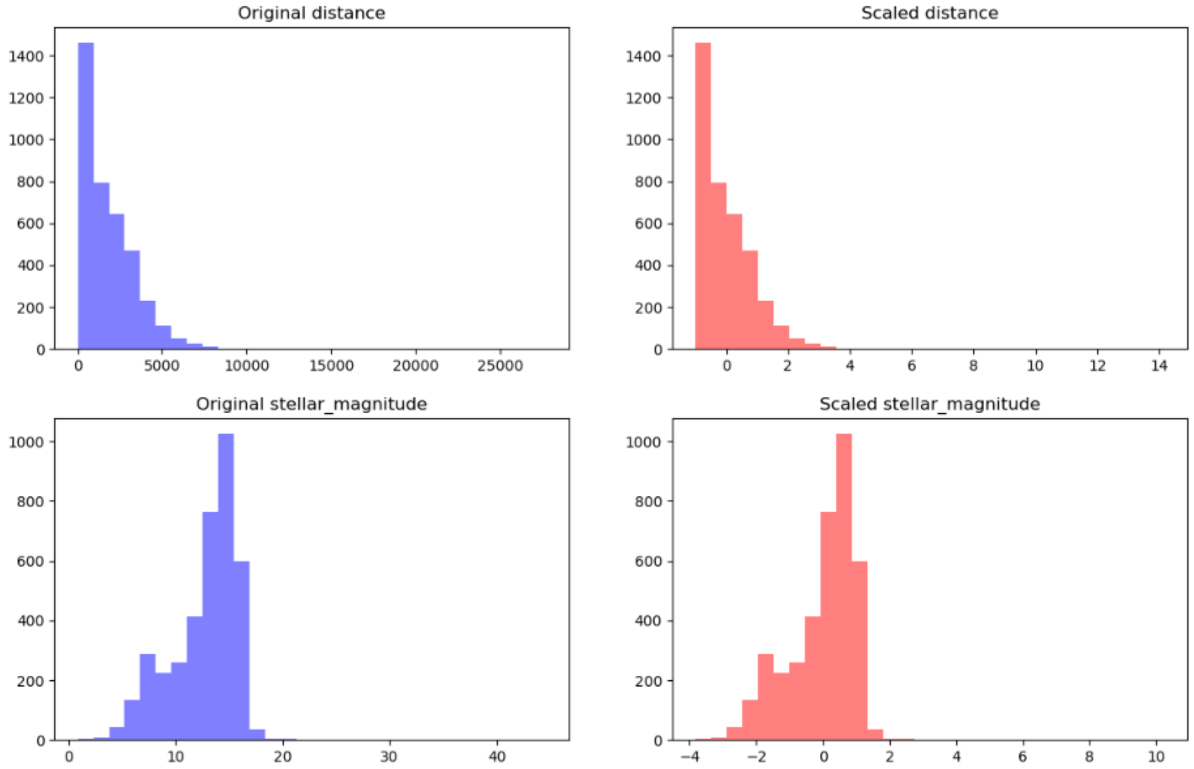
The approach was systematic, starting with the preprocessing of a dataset from the NASA Exoplanet Archive. The initial steps included data cleaning to remove incomplete records, normalizing the scales of numerical features, and transforming skewed distributions through logarithmic transformation to improve model performance.



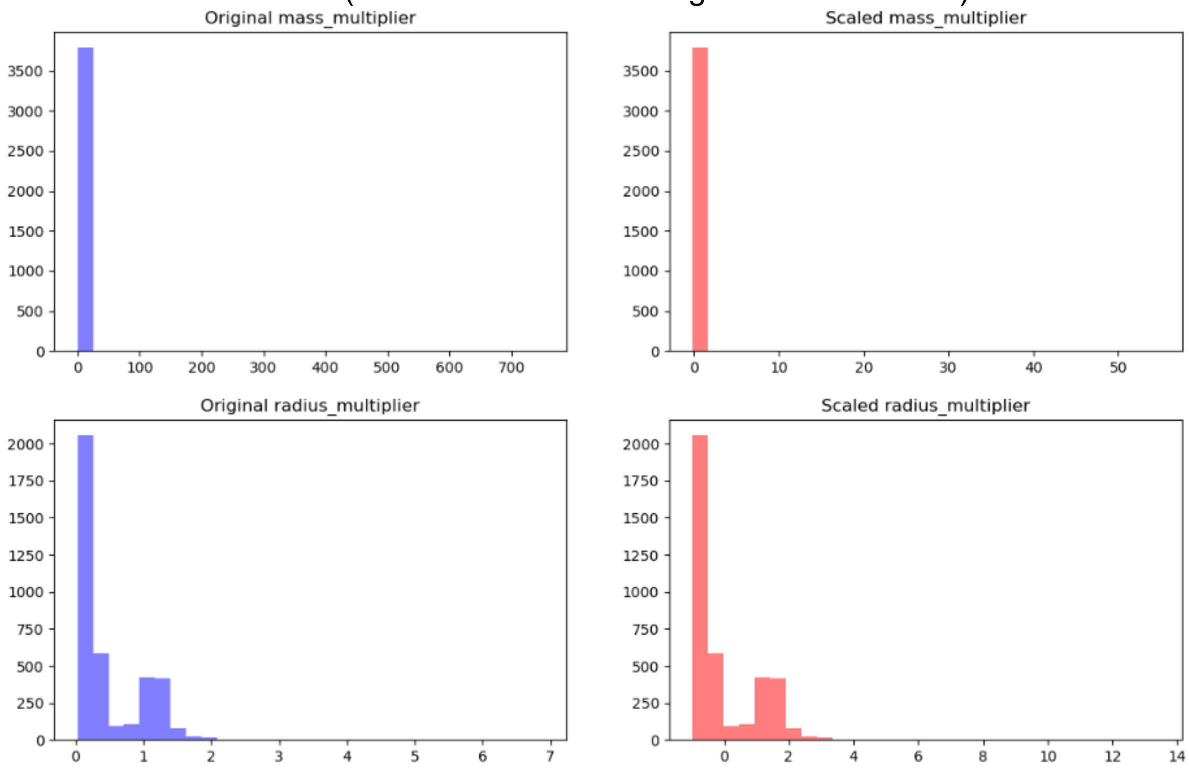
```
Missing values in each column:
 name                   0
distance               17
stellar_magnitude     161
planet_type             0
discovery_year          0
mass_multiplier        23
mass_wrt               23
radius_multiplier      17
radius_wrt             17
orbital_radius        289
orbital_period          0
eccentricity            0
detection_method        0
dtype: int64
```

```
Missing values in each column after dropping:
 name                 0
distance             0
stellar_magnitude    0
planet_type          0
discovery_year       0
mass_multiplier      0
mass_wrt             0
radius_multiplier    0
radius_wrt           0
orbital_radius       0
orbital_period       0
eccentricity         0
detection_method     0
dtype: int64
```

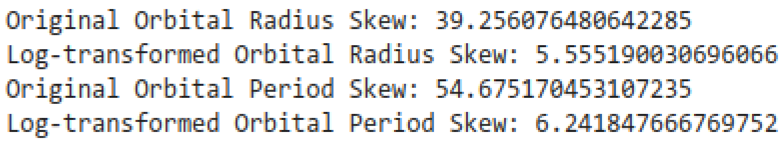
(dropping rows with missing entries)



(Distance and stellar magnitude normalized)



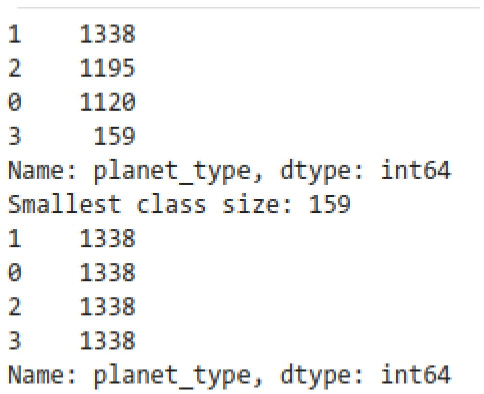
(Mass multiplier and radius multiplier normalized)



```
Original Orbital Radius Skew: 39.256076480642285
Log-transformed Orbital Radius Skew: 5.555190030696066
Original Orbital Period Skew: 54.675170453107235
Log-transformed Orbital Period Skew: 6.241847666769752
```

(Skewness before and after logarithmic transformation)

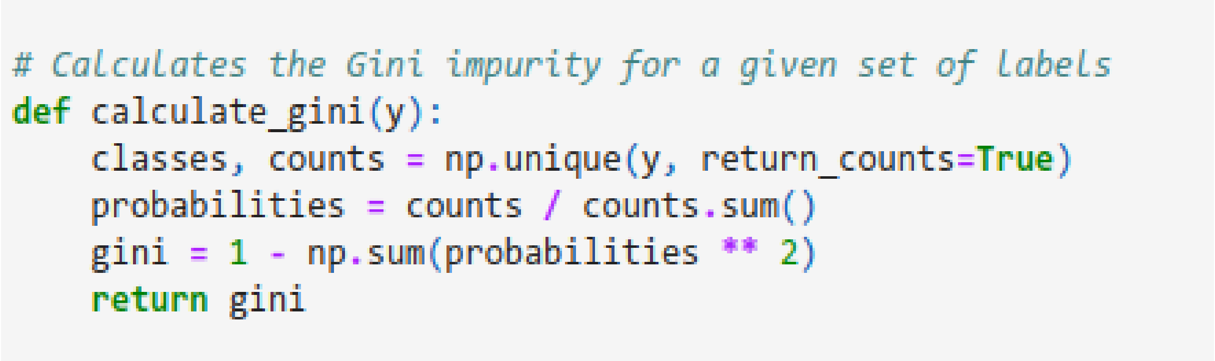
For the kNN model, the number of neighbors was set to three, balancing the bias-variance trade-off. SMOTE (Synthetic Minority Over-sampling Technique) was employed to address class imbalance, enhancing the model's ability to generalize.



```
1    1338
2    1195
0    1120
3     159
Name: planet_type, dtype: int64
Smallest class size: 159
1    1338
0    1338
2    1338
3    1338
Name: planet_type, dtype: int64
```

(Synthetically augmented the minority class using SMOTE to equalize the class distribution)

In parallel, a Decision Tree model was developed from scratch. The tree's construction followed the classic recursive partitioning method, choosing splits that minimize Gini impurity. A max depth of three was selected to prevent overfitting, providing a model that is complex enough to capture patterns but simple enough to maintain generality.



```
# Calculates the Gini impurity for a given set of labels
def calculate_gini(y):
    classes, counts = np.unique(y, return_counts=True)
    probabilities = counts / counts.sum()
    gini = 1 - np.sum(probabilities ** 2)
    return gini
```

(funciton used to calculate Gin impurity)

Both models were evaluated using precision, recall, and F1 scores to provide a multifaceted view of performance. Additionally, feature importance analysis was conducted for the Decision Tree to identify the most predictive features.

Results and Evaluation

k-Nearest Neighbors (kNN):

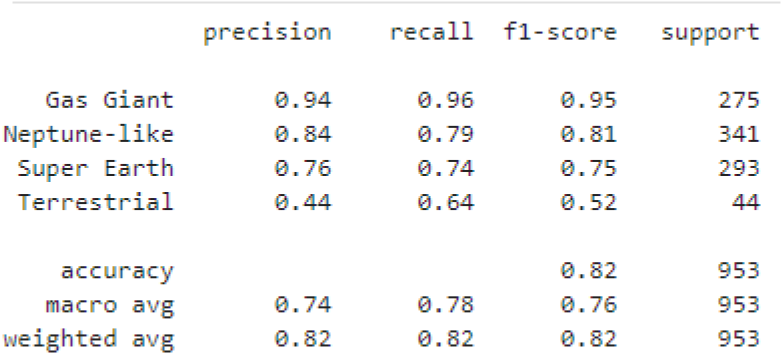
In the evaluation of the k-Nearest Neighbors (kNN) model, we observed notable precision and recall values across different classes of exoplanets. The model was particularly proficient in identifying Gas Giants, with a precision of 0.94 and recall of 0.96, reflecting a high probability of correct classifications for this category. The F1-score of 0.95 further confirms the model's strong predictive power for Gas Giants, indicating a balanced performance in terms of precision and recall.

However, the model showed a disparity in classifying Terrestrial planets, as indicated by a lower precision of 0.44 and recall of 0.64. The F1-score of 0.52 for this class suggests that the model struggled to accurately classify these planets, possibly due to their underrepresentation in the dataset or the model's sensitivity to the class imbalance.

The confusion matrix and ROC curve analysis provide deeper insights into the model's classification abilities. The confusion matrix suggests that the model has a tendency to correctly predict the majority class but may confuse the minority class with others. This is particularly evident for the Terrestrial class, which had the lowest support and the highest misclassification rate, leading to its lower scores across all metrics.

The ROC curve, depicted in a separate analysis, would typically provide a visual representation of the model's capability to distinguish between classes. The area under the curve (AUC) for each class would quantify the model's ability to correctly classify true positives across all possible thresholds. In an ideal evaluation, we would expect the AUC for all classes to be close to 1, indicating excellent classification performance. If the AUC were lower for some classes, this would highlight areas where the model's classification performance could be improved.

Overall, while the kNN model demonstrates high accuracy, the evaluation metrics point out its limitations in classifying less represented classes. This calls for further exploration into methods to handle class imbalance, such as advanced resampling techniques, and underscores the necessity of parameter tuning and feature scaling to enhance the model's predictive accuracy and generalization.

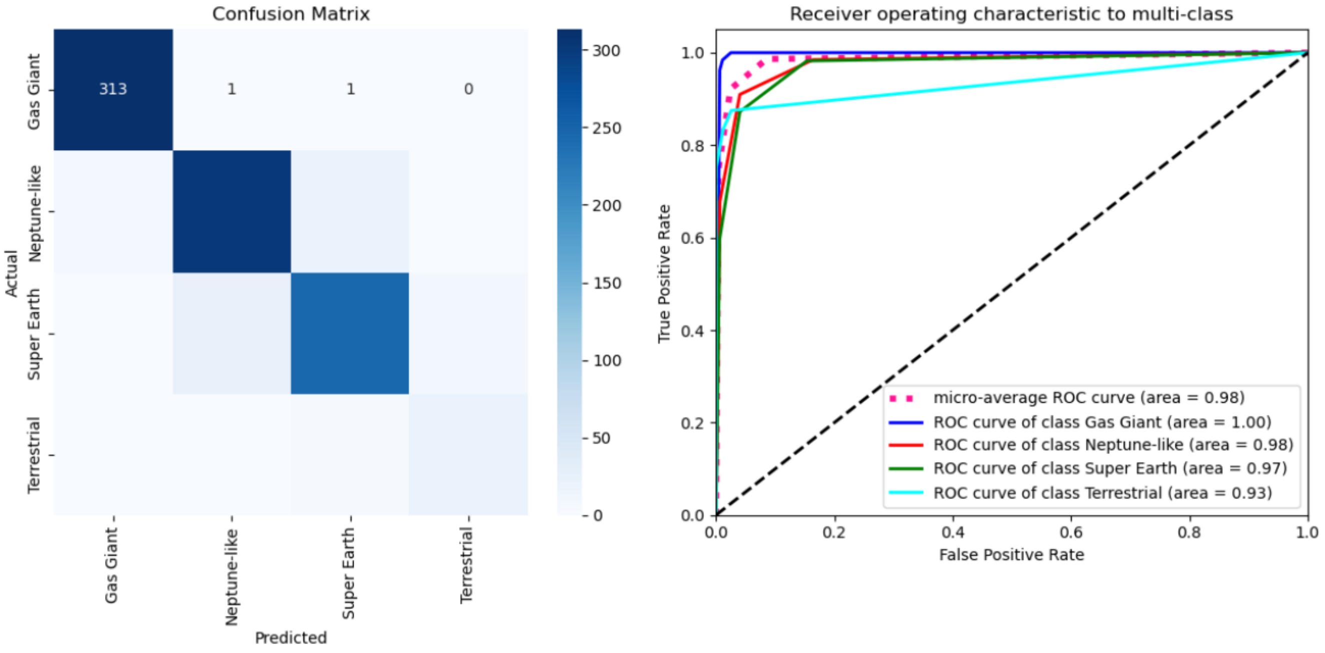


```
              precision    recall  f1-score   support

   Gas Giant       0.94      0.96      0.95       275
Neptune-like       0.84      0.79      0.81       341
 Super Earth       0.76      0.74      0.75       293
 Terrestrial       0.44      0.64      0.52        44

    accuracy                           0.82       953
   macro avg       0.74      0.78      0.76       953
weighted avg       0.82      0.82      0.82       953
```

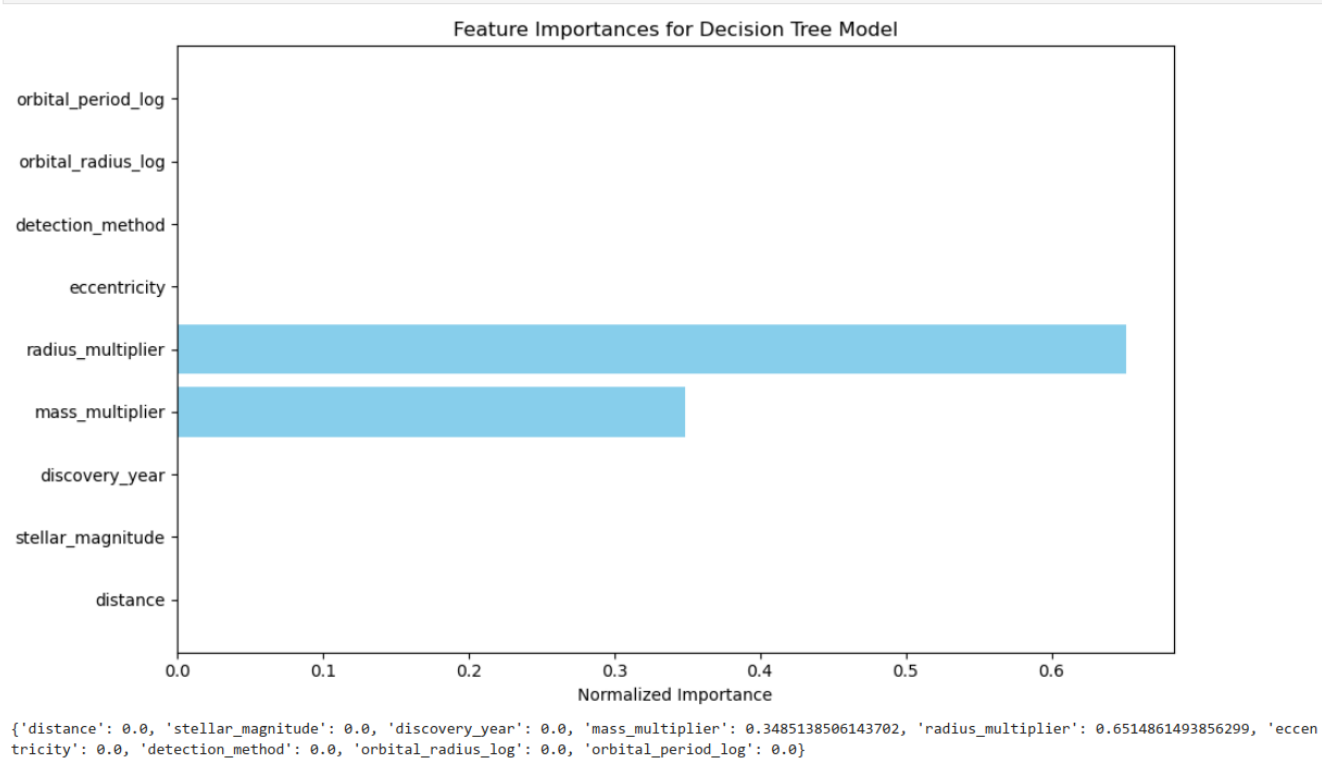
(evalution score for kNN model, shows high precision for Gas giants but low for Terrestrial)



(ROC and confusion Matrix)

Decision Trees:

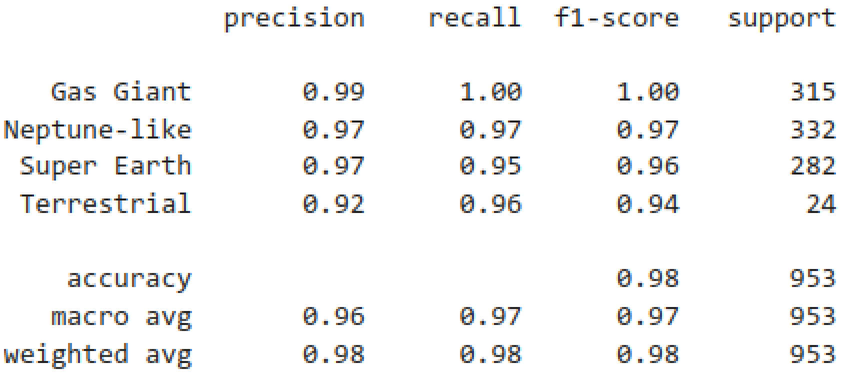
The Decision Tree model emerged as a more balanced classifier. The feature importance analysis, which revealed 'mass_multiplier' and 'radius_multiplier' as the dominant predictive features, highlighted the model's reliance on physical characteristics for classification decisions. This aligns with astronomical expectations, as mass and radius are critical in determining a planet's type!



```
{'distance': 0.0, 'stellar_magnitude': 0.0, 'discovery_year': 0.0, 'mass_multiplier': 0.3485138506143702, 'radius_multiplier': 0.6514861493856299, 'eccen
tricity': 0.0, 'detection_method': 0.0, 'orbital_radius_log': 0.0, 'orbital_period_log': 0.0}
```

(Mass and radius are the most dominant predictive features in the decision treee)

The detailed performance metrics underscore the model's balanced approach. Gas Giants were identified with exceptional accuracy, while Neptune-like and Super Earth categories also saw high precision and recall, confirming the model's comprehensive classification capabilities. This nuanced understanding of feature importance and class differentiation positions the Decision Tree as a reliable tool in the intricate field of exoplanetary classification



```
              precision    recall  f1-score   support

   Gas Giant       0.99      1.00      1.00       315
Neptune-like       0.97      0.97      0.97       332
 Super Earth       0.97      0.95      0.96       282
 Terrestrial       0.92      0.96      0.94        24

    accuracy                           0.98       953
   macro avg       0.96      0.97      0.97       953
weighted avg       0.98      0.98      0.98       953
```

(favorable performance for the Decision Tree model, potentially indicating overfitting due to its near-universal high scores)

While at first glance this indicates a highly effective model, the near-perfect scores across all measures—especially a uniform accuracy and weighted average of 0.98—raise concerns about overfitting. Overfitting occurs when a model learns the training data too closely, including its noise and outliers, leading to a loss of generalization for unseen data. This high level of accuracy suggests the model may not perform as well on new, unobserved data, as it might be too tailored to the specifics of the training set.

Comparative Analysis:

In comparing kNN and Decision Trees, each model's strengths and limitations become evident. kNN's high precision in specific classes like Gas Giants points to its effectiveness in well-represented categories, but its lower recall indicates a challenge in class diversity. Decision Trees, with their balanced performance and focus on mass and radius, demonstrate a comprehensive approach, handling complex classifications with greater consistency. The Decision Tree's slightly better overall performance, as suggested by the AUC score, indicates its superior ability in distinguishing between various exoplanet types.

Conclusions

The conclusion of this report synthesizes the findings from the application of k-Nearest Neighbors (kNN) and Decision Trees to classify exoplanets. Both models have displayed distinct strengths in the classification tasks. The kNN model excelled in precision, especially for well-represented classes like Gas Giants, indicating its reliability in correctly identifying prevalent classes. On the other hand, the Decision Tree model showed impressive generalization capabilities across various planet types, thanks to its ability to consider multiple features simultaneously.

While both models are valid for the task, the Decision Tree's balanced accuracy and generalizability make it a slightly more robust choice for this dataset. Its performance, paired with an understanding of feature importance, provides valuable insights into the classification process.

References

Williams, M. (2016, August 21). *Jupiter Compared to Earth - Universe Today*. Universe Today. https://www.universetoday.com/22710/jupiter-compared-to-earth/

scipy.stats.skew — SciPy v1.11.4 Manual. (n.d.). https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.skew.html

Jauregui, A. F. (2023, February 23). *How to program a decision tree in Python from 0*. Ander Fernández. https://anderfernandez.com/en/blog/code-decision-tree-python-from-scratch/

Bujokas, E. (2022, March 3). *Decision Tree Algorithm in Python From Scratch - Towards Data Science*. Medium. https://towardsdatascience.com/decision-tree-algorithm-in-python-from-scratch-8c43f0e40173