Habitability Prediction of Exoplanets Using Statistical Machine Learning Techniques

Rishab Kashyap (1225438201), Adrija Nag (1225464032), Monalisa Dokania (1225344640), Thejas A Naik (1222309959), Raviteja Reddy Guntaka (1225497663)

December 7, 2022

Abstract

Exoplanet exploration and habitability categorization is a massive field of research that attracts a significant amount of interest from experts to this day. Finding extraterrestrial life has been a goal for centuries, and the process of detecting such systems requires a plethora of data as well as procuring useful information from it. We have said data at our disposal from previous space exploration missions, and now, through our work, we aim to analyze this data and predict the habitability of an exoplanet using a collection of machine learning models. Upon performing the preprocessing on the data, these models result in an average accuracy of 97%. Additionally, we use the F1-score, Precision, and Recall as our metrics to create a comparative study from our models and deduce conclusions based on the outcomes of each of them.

1 Introduction

Finding signs of life in outer space has been one of the most important questions in the field of space research that still goes unanswered to this day. Scientists and astronomers alike have spent decades obtaining and understanding the various attributes of celestial bodies to understand what constitutes signs of life. Through the Kepler mission and its descendants like K2 and the James Webb Telescope project, NASA has achieved tremendous success in finding numerous confirmed exoplanets similar in size to earth, with plenty of candidates yet to be given a planetary status. To date, 5220 candidates have been confirmed as exoplanets by NASA. A major motivation behind these missions was to look for planets supporting life. But how do we declare a planet to be habitable without having discovered one? Astrobiologists examine the planets for biosignatures, which are chemical signatures associated with life and biological processes, the most important of which is water. However, the process is rather complex. Existing work on categorizing exoplanets is based on assigning habitability scores to each planet which allows for a quantitative comparison with Earth. The Planetary Habitability Index is a distance-based metric to compare the similarity of a planet to that of Earth. Given the accumulation of large amounts of data, it is the need of the hour to explore the advanced methods of data analysis based on machine learning techniques to rapidly classify planets into their relevant category and hence automate the process. The Planetary Habitability Laboratory has published a list of potentially habitable planets based on the Kepler data. The goal of this project is to use this data as training data, and use the Planetary and Stellar features to develop a reasonably accurate machine learning model to classify these exoplanets into 3 categories, which could predict more new potentially habitable planets as and when more exoplanets candidates are confirmed by NASA. In addition, we also explore the efficacy of the different models and create a comparative study between them to understand the differences in their performances as well as their efficiency in working with a similar dataset.

2 Literature Survey

There are a couple of literary works that look into the classification of celestial bodies as exoplanets even before detecting whether they are habitable. Some of the earlier works look into the process of classification of a valid exoplanet [5] and storing these as a part of the Kepler planetary database [7]. Recent years have seen the introduction of a variety of methods used to classify the planets, and an introduction to the habitability of each of these exoplanets using the data [1]. This allowed programmers to use this data to automate the classification process. Some of the literature work look into supervised machine learning models [12] that allow the models to create precise predictions based on the values since the labels are provided for each outcome. It is important to consider the correct features prior to performing the classification by finding feature importances [2]. This is improved by the introduction of unsupervised learning as well as deep learning models [**8**] [**9**] where deep neural networks and convolution networks are used to classify as well as analyze the reason behind the classification. In our work, we limit ourselves to the usage of a multi-layer perceptron for the comparative analysis between model performances.

3 Proposed Methodology

In the following project, we have performed the classification of habitable exoplanets using an array of machine-learning models. Furthermore, we analyze the performance of each of these models to create a comparative study of their efficiencies. The process followed through the project is shown in Figure 1, where we first perform the data preprocessing step where we have our feature extraction taking place. Once we have the usable features, we use all or a subset of the features to build each machine learning model based on the type of features required to train them. Upon training and testing the models, we perform a comparative classification on the basic metrics as we shall see in the sections that follow.

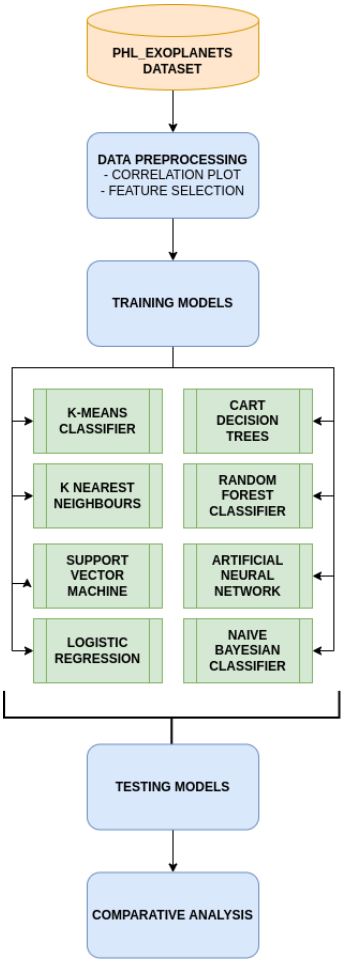

Figure 1: Proposed Methodology

4 Data Preprocessing

Before feeding the raw data as input to our machine learning models, we will first pre-process the data in order to identify some essential features on which our model can be trained and tested.

4.1 Datasets used

We have used the University of Puerto Rico's Habitable Exoplanets Catalog by the Planetary Habitability Laboratory (PHL).

4.2 Cleaning and Preprocessing

The following steps were used for data preprocessing:

1) Since it is usual in astronomical datasets to have lots of missing values, we start by cleaning these empty values by removing the columns with more than 12% of null values.
2) Next we convert the Categorical Values to Numerical Values using the Label encoder function of the Python sklearn library. There were four such columns, for example, the column specifying temperature "P_TYPE_TEMP" with values "Cold", "Hot", and "Warm" was converted into 0, 1, and 2 respectively.
3) Next we identify columns that have a weak relationship between the numerical variables, that is, it does not affect the output to a large extent. A variable can be positively as well as negatively correlated. In order to measure this correlation, we first visualize the data using a heatmap (as shown in Figure 2), with the help of the Python library Seaborn. The rows represent

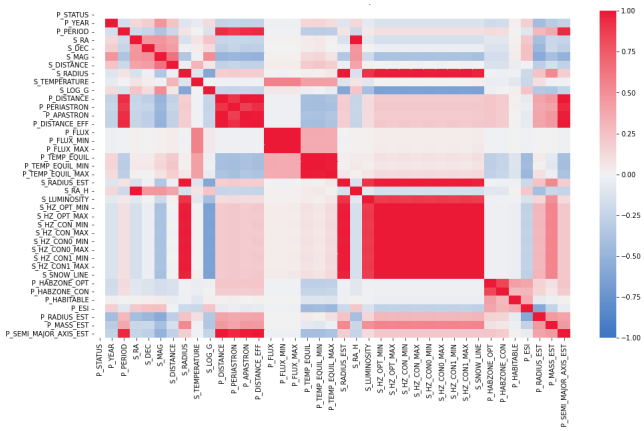

Figure 2: Correlation heatmap

the relationship between each pair of variables. If we observe the upper triangular part (leaving the diagonals), we observe that the target column "P_HABITABLE" with 0 (Nonhabitable), 1 (Conservatively habitable), and 2 (Optimistically habitable) values, is strongly correlated to "P_ESI", "P_HABZONE_OPT", and "P_HABZONE_CON", as expected. The red zone represents a duplication of data and the grey and white zone are of interest to us. On the other hand, some features have correlation 1 (shown in red near the diagonal line) which points toward duplication of information. Also, some columns like star coordinates are unrelated features to the habitability prediction of planets. Hence, we remove all these unnecessary features.
After processing our data, we are left with 30 columns, out of which we only consider 20 numerical columns as our predictor set to input into our models and one column as the target label.

The planets have been classified based on the following target feature - 'P_HABITABLE', with values:
0 - Non-Habitable
1 - Conservatively Habitable
2 - Optimistically Habitable

5 Models Used

5.1 Support Vector Machines

This is a multiclass classification problem, and SVM using Kernels is a good model for that. The idea is to map the data to a high dimensional space and then classify it using a "One-to-one" approach. This model is defined by the Primal and the Dual Problems. The SVM function that we used will employ a binary classifier for each pair of the three classes that we have. We have used our SVM classifier with an RBF Kernel to incorporate a classifier that is non-linear. Also, the slack penalty value considered here is 10. A low value of 'C' tries to make the decision surface smooth. Using a random_state=42, we got the results with the confusion matrix as shown in Figure 3. The error was 0.84, and the F-1 score was 98.56%.

We also used linear kernel in our experiments, with class_weight parameter value 'balanced' in order to compensate for the imbalanced nature of the dataset. The confusion matrix is shown in Figure 4. The error was 0.24, and the average F-1 score was 99.66%.

5.2 Logistic Regression

The logistic model, often known as the logit model, is a statistical model that estimates the likelihood of an event occurring by making the event's log

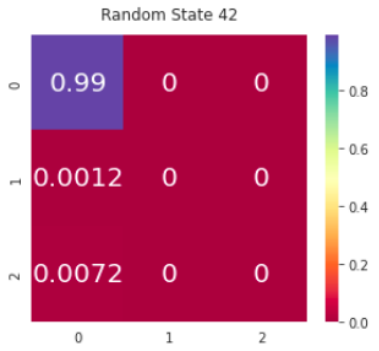

Figure 3: SVM Classifier (rbf Kernel)

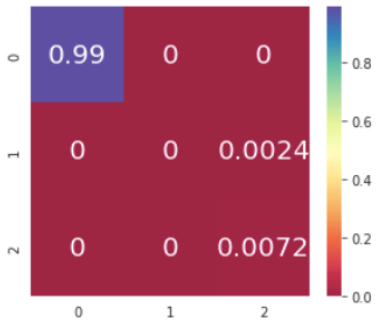

Figure 4: SVM Classifier (Linear Kernel)

odds a linear combination of one or more independent variables. Formally, binary logistic regression has a single binary dependent variable (two classes, coded by an indicator variable) with the values ”0” and ”1,” whereas the independent variables can either be continuous variables or binary variables (two classes, coded by an indicator variable) (any real value). The value labeled ”1”’s related probability can range from 0 (definitely the value ”0”) to 1.

5.3 Naive Bayes Classifier

This model is built on the consideration that the features are causal to the target vector and attributes are independent classes while predicting habitability. This assumes the gaussian distribution of our input dataset and draws the output classes based on the posterior probabilities of the classes assuming a prior, as shown below.

Output metrics of the Gaussian Naive Bayes classifier:
Train Accuracy: 98.23%
Test Accuracy: 97.95%
Precision: 75.7561%
Recall: 82.684%
F-1 Score: 69.413%

5.4 K Nearest Neighbors

When categorizing data, the k-nearest neighbors (KNN) method determines the likelihood that a data point will belong to one group or another based on the group to which the data points closest to it belong. An important property of K Nearest Neighbors is that the model converges better, increasing the value of K to a certain point. Also, the worst fit of the model is achieved at K=1. In our case, we had the highest precision at K=3. The graph received for KNN at k=3 is shown in Figure 5. If we observe the graph close enough, we can notice that three different clusters have been formed. The yellow cluster represents the planets that are confirmed to be habitable, the orange cluster represents planets that are known to be conservatively habitable, and the blue cluster represents the uninhabitable planets. We assessed the performance of our KNN model using a Confusion matrix. The metrics obtained were as follows:
The errors for the three clusters obtained:(0.96,0.84,0.84)Mean Error: 0.88

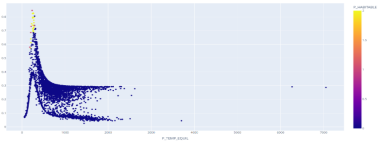

Figure 5: K Nearest Neighbors

5.5 K Means Clustering

K-means clear objective is to discover underlying patterns by combining comparable data points. K-means searches a data set for a predetermined number (k) of clusters to accomplish this goal. To find this K before the start of model training, we generally depend on analysis like plotting the

Elbow curve based on K-means convergence as shown in Figure 6. In this, we keep on increasing the value of K in every iteration and plot the loss or error function (distortion score in our case) against the K value. The Elbow curve indicates at what rate the distortion score is getting updated and further gives a clear Elbow structure at K=3.

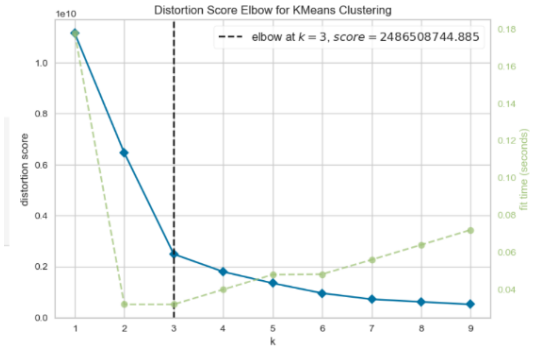

Figure 6: K Means Elbow Curve

A Silhouette score is a good measure of model performance for K Means Clustering. Its value usually ranges from -1 to 1 and is given by:

$$\frac{a-b}{max(a,b)}$$

, where a and b are the average distances of points belonging to the two different clusters. The Silhouette score for our model was 0.6418.

5.6 Decision Trees

A supervised machine learning algorithm that employs a set of principles to make judgments, much like how people do. We make use of a CART decision tree model, where we used a Gini index to determine the probability of classifying a random and an incorrect feature. Here is a diagram that depicts the following, through Figure 7.

By generating confusion matrices for our model, we were able to determine that we ended up with high training and test accuracy, with a minimal loss of 0.2%.

5.7 Random Forest

Similar to the Decision Tree classifier, the reason behind using the CART trees from earlier is to use the GINI index that allows us to implement the Random Forest classifier. This classifier creates multiple decision tree models and chooses the best-performing model out of the list of them. As we can see for the accuracy metric of the random forest classifier in Figure 8, we have a 99% accuracy on it similar to that of the decision tree in the case of testing.

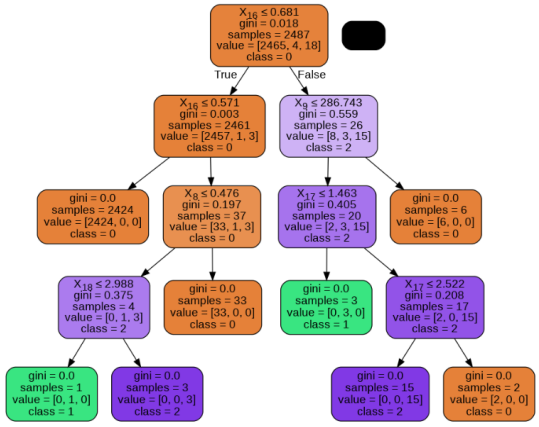

Figure 7: Decision Trees

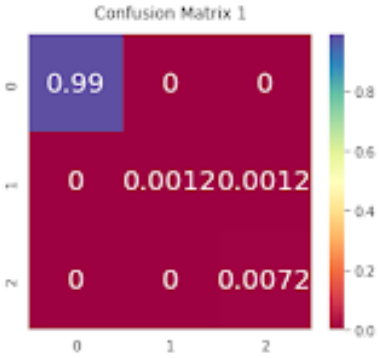

Figure 8: Confusion Matrix for Random Forest Classifier

5.8 Neural Networks

Artificial Neural networks are one of the most powerful models used for classification purposes, specifically in the case of a multi-layer perceptron network. We did not require to look into complex neural networks such as RNNs and CNNs since we were getting high metrics through a simple MLP. The structure of the Neural Network is shown in

Figure 11.

Figure 9: Loss curve for Neural Network

We train the model for over 20 epochs for 3 rounds to gain the required accuracy. Parameter fine-tuning involved changing the number of layers, the number of nodes per layer, the number of epochs to train with, as well as the usage of different activation functions. Through this, we came up with the final structure and achieved an accuracy of 98% on this. The loss curve can be seen as shown in Figure 9.

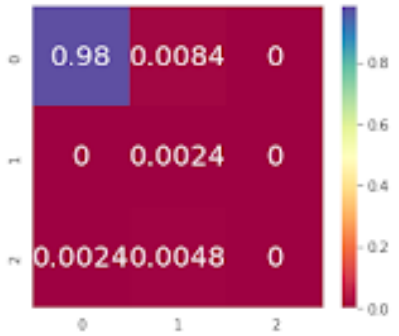

Figure 10: Accuracy for Neural Network

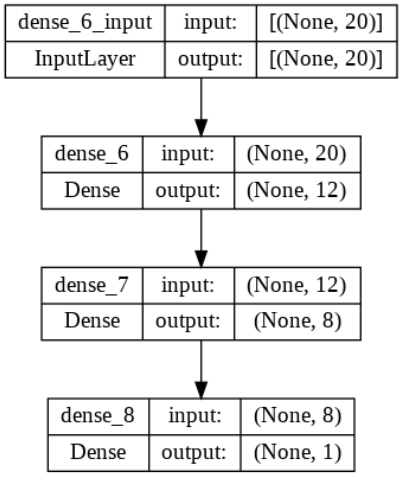

Figure 11: Structure for Neural Network

6 Model Comparison

We used the following metrics to compare our models: **Accuracy**, **Precision**, **Recall**, **F1-Score**. As we can see in Figures 12, 13, 14, and 15, most models performed extremely well on the dataset in the case of accuracy metrics other than K-Means which eliminates one model out of the rest of the models for validity and plotting. The rest of the models give high accuracies, but some models such as the Naive Bayes classifier and K-Means fall short when it comes to the other metrics.

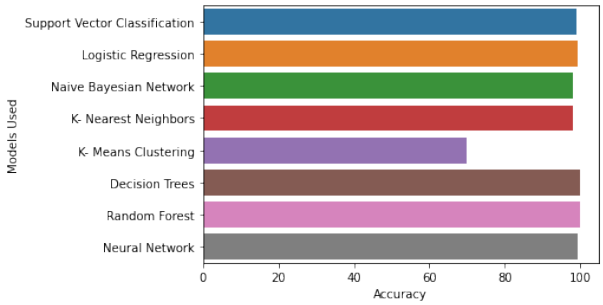

Figure 12: Accuracy Values

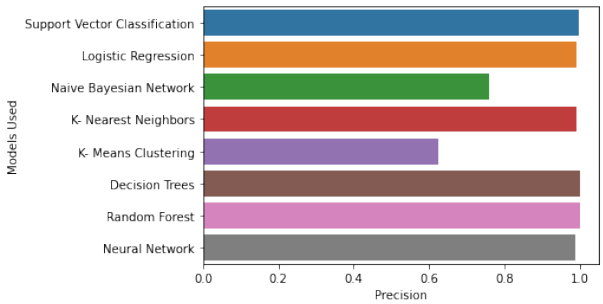

Figure 13: Precision Values

Further, in the case of the KMeans prediction, we calculate the Silhouette score to give us the accuracy metric as mentioned before. Since the value is not as high as expected, as well as very poor performance as seen, KMeans cannot be used for classification. Similarly, Cohen's Kappa value was used on the Neural Network to check the validity and return a value of 0.21, it shows us that there is a moderate agreement between the prediction

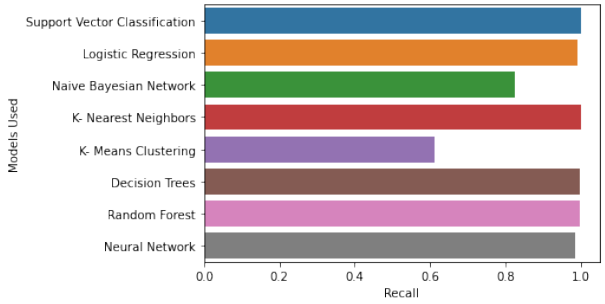

Figure 14: Recall Values

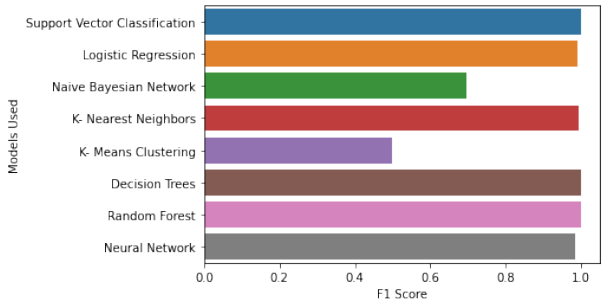

Figure 15: F1 Scores

values as made by the model with respect to the dataset itself. The difference between this metric and the accuracy is the fact that the Kappa value does not show how well the model predicts, but how well we can interrelate the feature classification done in the model with respect to the features that are present with respect to the corresponding labels in the dataset. From the final comparison, we can show that Logistic Regression, Random Forest classifiers, Neural Networks, and Decision Trees prove to be the most efficient in their classification.

7 Discussion and Conclusion

One of the interesting points to ponder would be why this is an important research problem. Classification is already done with labels being generated by experts. But the advantage we have with an automated system comes from not only saving a lot of time with calculations but models such as Neural Nets and RNNs can understand massive amounts of data and can not only classify but can generate similar examples by plugging in randomized values, allowing for future predictions to become a lot easier.

Research on the habitability of planets is an ongoing work, and the threshold of this research is expected to improve as we are bound to discover more habitable planets. Incorporating more relevant features in this discovery can help with this research. Also, experimenting with more models for the purpose of classification also adds credibility to finding more habitable planets as we look forward to discovering what's out there.

References

[1] The Planetary Habitability Laboratory (PHL). "THE HABITABLE EXOPLANETS CATALOG". In: (2021). URL: https://phl.upr.edu/projects/habitable-exoplanets-catalog.

[2] J Adassuriya, JANSS Jayasinghe, and KPSC Jayaratne. "Identifying Variable Stars from Kepler Data Using Machine Learning". In: *UMBC Student Collection* (2021).

[3] Megan Ansdell et al. "Scientific domain knowledge improves exoplanet transit classification with deep learning". In: *The Astrophysical journal letters* 869.1 (2018), p. L7.

[4] Natalie M Batalha. "Exploring exoplanet populations with NASA's Kepler Mission". In: *Proceedings of the National Academy of Sciences* 111.35 (2014), pp. 12647–12654.

[5] William J Borucki et al. "Kepler planet-detection mission: introduction and first results". In: *Science* 327.5968 (2010), pp. 977–980.

[6] Rutuja Jagtap et al. "Habitability of exoplanets using deep learning". In: *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*. IEEE. 2021, pp. 1–6.

[7] NASA and Caltech. "The NASA Exoplanet Archive". In: (2011). URL: https://exoplanetarchive.ipac.caltech.edu/.

[8] PHL. "References for our dataset and codes". In: (2020). URL: https://www.kaggle.com/datasets/chandrimad31/phl-exoplanet-catalog/.

[9] Ishaani Priyadarshini and Vikram Puri. "A convolutional neural network (CNN) based ensemble model for exoplanet detection". In: *Earth Science Informatics* 14.2 (2021), pp. 735–747.

[10] N Schanche et al. "Machine-learning approaches to exoplanet transit detection and candidate validation in wide-field ground-based surveys". In: *Monthly Notices of the Royal Astronomical Society* 483.4 (2019), pp. 5534–5547.

[11] George Clayton Sturrock, Brychan Manry, and Sohail Rafiqi. "Machine learning pipeline for exoplanet classification". In: *SMU Data Science Review* 2.1 (2019), p. 9.

[12] Varad Vishwarupe et al. "Comparative Analysis of Machine Learning Algorithms for Analyzing NASA Kepler Mission Data". In: *Procedia Computer Science* 204 (2022), pp. 945–951.