On Planetary Habitability: Using various machine learning methods to predict habitability of ExoPlanets

Christian Justine G. Clemente*

*University of Santo Tomas, Institute of Information and Computing Sciences

Abstract

Finding life on planets beyond our solar system has a critical impact in our species. Whether we are alone or not, finding us a new home is critical for the survival of humanity. There are many criteria that scientists use to detect life on classified Exo Planets. In this project, various machine learning methods are explored and analyzed in order to help scientists detect whether an exoplanet is habitable depending on the planetary features and stellar features.

Keywords: AstroPhysics; Machine Learning; Planetary Habitability; *ExoPlanets*

1. Introduction

According to Planetary Habitability Laboratory[1],there are currently 3875 confirmed exoplanets, 55 of which are considered habitable. Predicting planetary habitability is considered difficult because we only have one real data point, *Earth*. Those considered as potentially habitable are all just theoretical computations because we still don't have the technology for interstellar travel and explore on those worlds detected as potentially habitable.

Planetary Habitability is the measure of a planet's potential to develop and maintain environments hospitable to life. Currently, the criteria for the planetary habitability is compared to Earth-like features, if the planet has similar features compared to Earth, then it can be considered as habitable.

Another criterion is if the planet is within the circumstellar *habitable zone (CHZ)* of the system. It is used as a criterion for measuring if the planetary surface can support liquid water given sufficient atmospheric pressure. The bounds of the CHZ is based on the Earth's position in the solar system and the amount of radiant energy it receives from the sun.

However, there are still planets that are not considered habitable even if it has earth-like features and within the habitable zone. This is because there are also other features that must be considered to detect habitability of planets.

K-Nearest Neighbour is an unsupervised machine learning algorithm that is commonly used for classification. It a distance metric to determine distances of a new data from the training data, the training labels who has the most numbers within *k neighbours* is the predicted value of the algorithm. One problem of the KNN algorithm is the curse of dimensionality, this is because data at higher dimensions is perceived. As the dimensionality increases, the volume of the space increases so fast that the data becomes sparse. This is the main reasons why K-Nearest Neighbour doesn't work at higher dimensions.

Decision Trees are a supervised machine learning algorithm that produces a decision tree after training. It can be used for classification. The goal of a decision tree is to create a model that predicts the value of a new data given several input variables. A tree can be *learned* by using several decision tree algorithms such as ID3 and C4.5

Support Vector Machines is another supervised machine learning algorithm. Unlike K-Nearest Neighbour, which uses a low dimensional data, Support Vector Machines plots the data into higher dimensions. It constructs a hyperplane, which is used to split the labelled data. The mapping used by SVM to project data into higher dimensions is defined by using a *kernel function*. There are 3 common kernel functions in SVM namely Polynomial Kernel, Linear Kernel, and Radial Basis Function (RBF).

Currently, there are 5 categories of habitable planets, ranging from size and temperatures of planets illustrated in Figure 1-

A

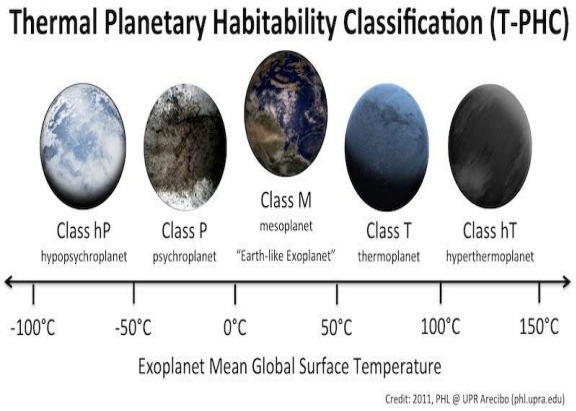


Figure 1-A Habitability Categories

2. Data Set and Features

The Dataset of this project came from Planetary Habitability Laboratory [1], which is compiled from the NASA ExoPlanet Archive [2].

The dataset from Planetary Habitability Laboratory consists of two data, Kepler Data and Confirmed ExoPlanets data. The difference is that the planets from the Kepler Data has some unconfirmed planets, but habitable ones, meaning those planets has a chance of being a satellite rather than a planet. In this project, only the data from the confirmed planets are used.

For the features, it is derived from the paper of [3], which produces a criterion for exoplanets derived from habitable zones around M and late K stars. The features cleaned from the dataset and based from [3] are:

- Orbital Period(days)
- Planet Mass
- Planet Gravity
- Fitted Stellar Density
- Planetary Radius
- Planet Density
- Planet Distance from the Star
- Orbital Semi Major Axis
- Planet's Equilibrium Temperature
- Escape Velocity
- Planet Surface Pressure
- Stellar Flux
- Stellar Effective Temperature
- Stellar Metallicity
- Stellar Radius
- Stellar Mass
- Stellar Luminosity
- Minimum Distance of Habitable Zone
- Maximum Distance of Habitable Zone

With these features, classification of habitability of exoplanets is now possible. Since currently, there are 100+ classified habitable planets and almost 3000+ of non-habitable planets, we can now use the 3 mentioned machine learning algorithms in the Introduction.

After removing the data which has missing values, the dataset consists of the following:

Label	Number of Data
Mesoplanet	20
Psychroplanet	14
Thermoplanet	2
Hyposychroplanet	3
Not Habitable	2065

3. Research Methods

The main programming language used in this project is *Python*. Along with the Pandas, Matplotlib, Scikit-Learn libraries. Accuracy is the only measure used in this project.

First, a visualization between different categories of label is programmed, this can be considered as a guide for later use when the data is applied into the machine learning algorithms. From the given features, Principal Component Analysis is used to convert features into a 2-dimensional array that is used to plot the data in the Figure 3-A.

The data is split by 80%-20% Training-Test split, the split is stratified so that each label is equally represented.

First, The K-Nearest Neighbor algorithm is used for the data. The optimal number of neighbors is first programmed, then after obtaining the optimal number of neighbors, KNN is tried at different dimensions of data, ranging from 1-20.

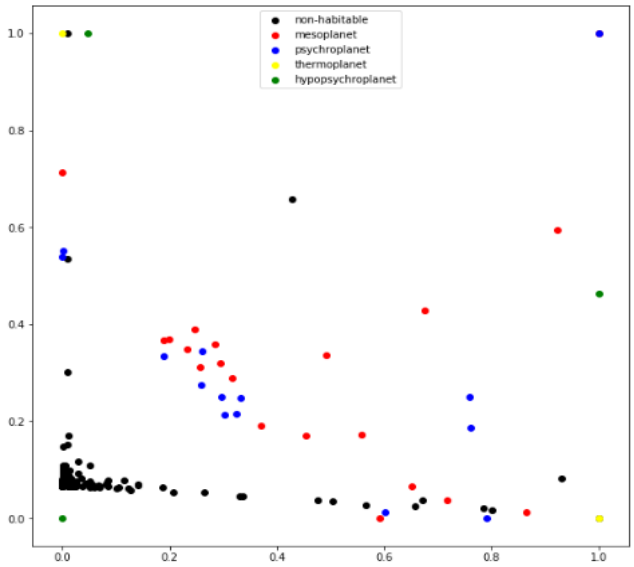


Figure 3-A. Scatterplot of the data

After testing with the KNN, the Support Vector Machines algorithm is now used. The data is tested at 3 different kernels of the SVM namely Polynomial, Linear, and RBF.

Then for the last machine learning algorithm, the decision tree. After training and testing, the decision tree will be visualized.

4. Results and Analysis

K Nearest Neighbor

For the K-Nearest Neighbor, normalized data and not normalized data is tested. It shows that normalized data is far greater because of the difference in values. The optimal number of neighbors is 15 neighbors, and after that, the accuracy is always the same. The neighbor vs accuracy is illustrated below in Figure 4-A. The best accuracy is at 98%.

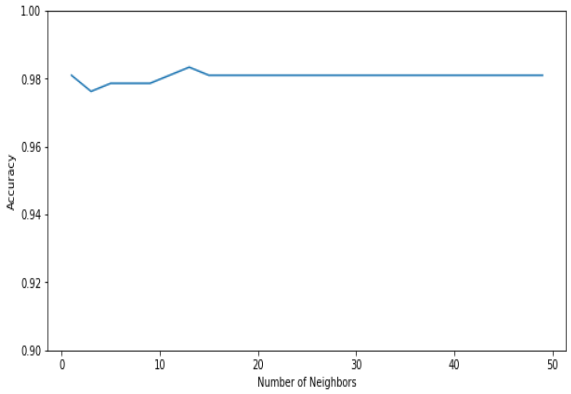


Figure 4-A. Accuracy vs. Neighbors Graph

Moreover, it seems that the data is not significantly affected by the curse of dimensionality, since from 1 dimension up to the length of features is tested, the accuracy has not changed. One of the reasons why this happened is because the data is correlated. A visualization of the accuracy between dimensions is shown in Figure IV-B. And the visualization of information loss between number of dimensions is illustrated at Figure IV-C.

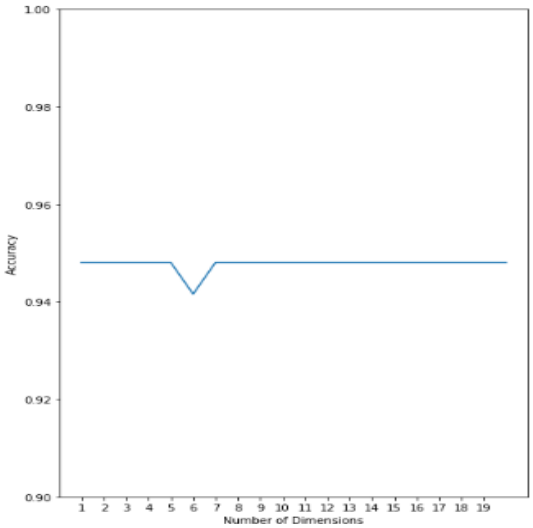


Figure IV-B. Accuracy between Number of Dimensions

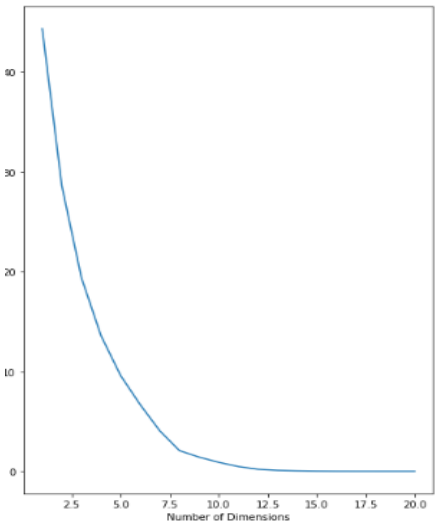


Figure IV-C. Information loss between number of dimensions

Support Vector Machines

For the SVMs, the 3 kernels: polynomial, linear, and RBF is tested in this data. The accuracy is the same between those 3 kernels, one of the reasons why is because the data is linearly separated from the start based from Figure 3-A.

The Accuracy of the SVM is at 95%. The value is the same for all the kernels and is illustrated below at Figure IV-D.

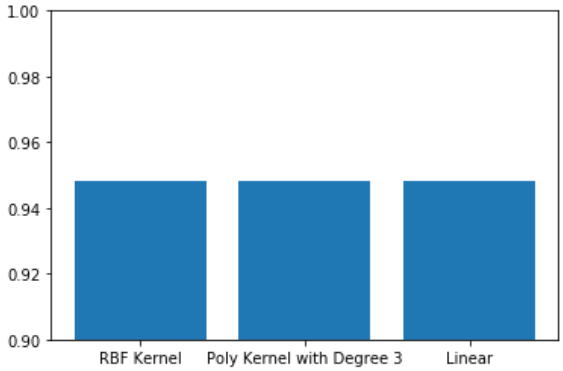
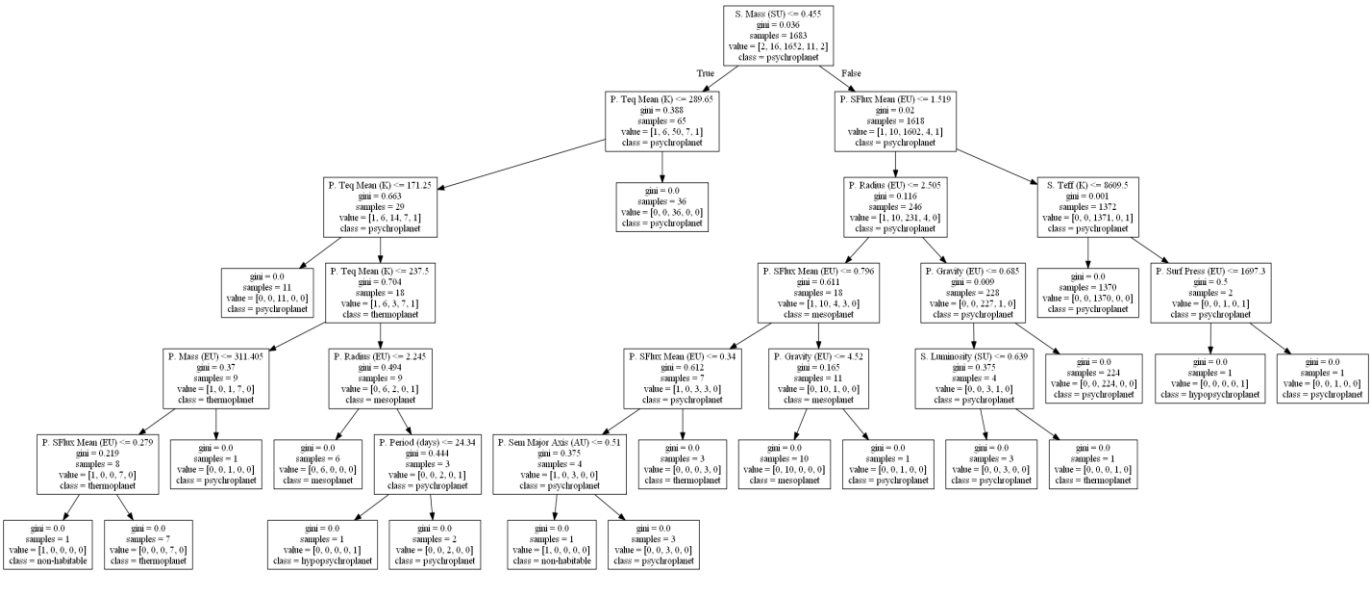


Figure IV-D. Accuracies of different kernels of SVM.

Decision Trees

For the decision trees, the data is not normalized unlike from SVM and KNN, since decision trees are based on values, it is no use to normalize the given data that will be fed into the decision trees. The accuracy of the decision tree is at 99%. The decision tree that is made after training is illustrated at Figure IV-E.

Figure IV-E. Decision Tree



5. Summary and Conclusion

Algorithm	Accuracy
K-Nearest Neighbor	98%
Support Vector Machine	98%
Decision Trees	99%

The SVM and the KNN has the same accuracy. One of the reasons why is because they depend on the linearly separability of the given data.

It seems that the best algorithm for this kind of problem is the decision trees, since it outputs a tree, it can is like a criterion that is used by scientists to detect the habitability of exoplanets.

6. References

[1] Planetary Habitability Laboratory. http://phl.upr.edu.

[2] NASA ExoPlanet Archive. https://exoplanetarchive.ipac.caltech.edu/.

[3] Kasting et al. Remote life-detection criteria, habitable zone boundaries, and the frequency of Earth-like planets around M and late K stars. 2014

Click here to delete instruction text