Predicting habitable exoplanets from NASA's Kepler mission data using Machine Learning

"A sad spectacle. If they be inhabited, what a scope for misery and folly. If they be not inhabited, what a waste of space." - Thomas Carlyle

RAJEEV MISRA

December 2017

Abstract

We are at unique juncture of human history where within next few years we may be able to confirm earth like planets or earth's twin around other stars where conditions are suitable for life to exists or even find a proof of life on those planets. NASA launched Kepler space telescope in 2009 which looked at a patch of sky and studied over 150,000 stars and found around 2500 confirmed planets around other stars using transit method. Scientist and researchers have identified several of these planets which could be suitable for life based on planet and parent star's features. Purpose of this project is to use data generated from Kepler space telescope and come up with machine learning model which could use planetary and stellar features to classify exoplanets into habitable and non-habitable planets.

1. Introduction

Our Milky Way galaxy has estimated 100 to 400 billion stars, and close to 2 to 3 billion stars similar to our Sun. Observable universe is around 46 billion light years across in radius and contains around 100 billion galaxies. Based on Kepler data, scientists have estimated that there could as much as 40 billion earth size planet in our own Milky Way galaxy alone [1]. Given these astronomically large numbers, there ought to be some stars, which have planets, where conditions are suitable for evolution of life or at least support life in the form we know.

In early 90s, scientist started discovering planets around other stars, called exoplanets, using gravitation pull of planets causing wobble in parent star's revolution axis which caused red or blue shift in parent star's spectrum (called Radial Velocity method), and later on, using other techniques such as Transit method.

Kepler space telescope was launched by NASA in 2009 to find exoplanets in our stellar neighborhood. Kepler finds exoplanets by looking for tiny dips in the brightness of a star when a planet orbiting it crosses in front of it, we say the planet transits the star.

Once detected, the planet's orbital size can be calculated from the period (how long it takes the planet to orbit once around the star) and the mass of the star using Kepler's Third Law of planetary motion. The size of the planet is found from the depth of the transit (how much the brightness of the star drops) and the size of the star. From the orbital size and the temperature of the star, the planet's characteristic temperature can be calculated [2]. These Planetary and Stellar parameters are published using Kepler's public data repository [3]. Kepler probe has found close to 9000 "CANDIDATE" planets and approximately 2237 planets have been "CONFIRMED" as planets after vetting.

Scientist have studied these planets and identified several planets which may be habitable or suitable for life. "Planetary Habitability Laboratory" has published list of habitable planets [4]. A recently published paper "Planetary candidates observed by Kepler. VIII" [5] identified several new potentially habitable planets from Kepler data.

This project's goal was to take these data as training data, and use Planetary and Stellar features to build a machine learning model which could predict potentially new habitable planets as and when more "confirmed" planets are published in Kepler's exoplanet archive.

Output of this project is a computer program [6] that builds machine learning model and predict habitability of planets from Kepler exoplanet archive data [7].

2. Dataset and Features

Primary source of our data was NASA exoplanet archive [7]. Each record in this archive represented one potential planet orbiting its parent star.

For our training, we needed set of planets which have been identified as potentially habitable. We obtained this list of habitable planets from “Planetary Habitability Laboratory” [4] and from a recently published paper “Planetary candidates observed by Kepler Kepler. VIII” [5]. In total, these sources identified 126 habitable exoplanets from Kepler’s data.

Since there was no explicit list of non-habitable planets, an assumption was made that all “confirmed” Kepler planets have been vetted for habitability and thus after removing habitable planets from “confirmed” planet list, whatever remained in “confirmed” planet list must be potentially non-habitable planet.

Since habitable planets were scarcity, planets with both "confirmed" and "candidate" disposition were selected [8]. Reasoning behind this was, even if "candidate" planet ultimately turned out to be "false positive", still its features would be a valid data point for our purpose. For non-habitable planets, we selected planets with only "confirmed" disposition [9].

126 habitable exoplanets and 2247 potentially non-habitable exoplanets were finally used in our training, dev & test exercise.

Each record of Kepler data was identified by a unique id called 'KOI' (Kepler Object of Interest). Each record contained 140 attributes about observation. There were several categories of attributes such as "exoplanet archive information" or "threshold crossing event" or other optical attributes [10] which were not directly related with planet’s habitability. For habitability analysis we were interested only in planetary and stellar features.

After analyzing each attribute of Kepler data from its description [10], 14 planetary and stellar features were identified. Some of these features are "Planetary Radius", "Isolation Flux", "Equilibrium Temperature", "Orbital Period", "Distance from parent Star", "Stellar Temperature" etc.

"Forward Feature search algorithm" as well as “Recursive feature elimination with cross validation” were experimented with to find feature set which provided most accuracy on dev data.

“Forward feature search” adds one feature at a time in feature set, each time with a feature that produces lowest dev error with given model (SVM with ‘rbf’ kernel in our case).

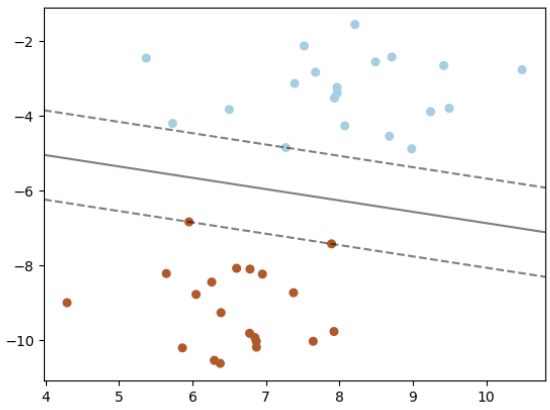
50/20/30 percent ratio of training, dev and test data was used during “forward feature search” algorithm.

In “Recursive feature elimination” algorithm, given an external estimator (SVM with ‘Linear’ kernel in our case) that assigns weights to features, recursive feature elimination is to select features by recursively considering smaller and smaller sets of features [11].

3. Methods

This was a binary classification problem. Support Vector Machines (SVM) is a good model for this kind of classification problem.

In a nutshell, SVM separates 2 classes of data through a hyperplane. It tries to find a biggest margin between nearest training data of 2 classes and hyperplane.



Above image [12] demonstrate SVM separating 2 classes of data with a 2 dimensional line (This would be hyperplane in higher dimensions) with largest margin and with closet 3 training data called support vectors.

Finding the hyperplane with maximum margin boils down to solving below primal problem

$$\min_{w,b,\xi} \left(\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}\xi_i\right)$$
$$s.t.\ \ y_i\left(w^T\phi\left(x^{(i)}\right) + b\right) \geq 1-\xi_i$$
$$\xi_i \geq 0, \quad i=1,..,m$$

Its dual is

$$\min_{\alpha}\left(\frac{1}{2}\sum_{i,j=1}^{m}\alpha_i y_i y_j K\left(x_i,x_j\right)\alpha_j - \sum_{i=1}^{m}\alpha_i\right)$$
$$s.t.\ \ \sum_{i=1}^{m}\alpha_i y_i = 0$$
$$0 \leq \alpha_i \leq 1 \quad i=1,..,m$$
$$where \quad K\left(x_i,x_j\right) = \phi\left(x_i\right)^T\phi\left(x_j\right)$$

Here K is a kernel with training data mapped to higher dimension using function ϕ

Decision function used was $sgn\left(\sum_{i=1}^{m} y_i\alpha_i K\left(x_i,x\right)+\rho\right)$

ρ is a intercept term

"Radial Basis Function" (rbf) kernel was used in our model, which is of the form

$$K\left(x,x'\right) = \exp\left(-\Upsilon\|x-x'\|^2\right)$$

Υ determines how far one training example's influence reaches. Low value means influence is far while higher value localizes that influence.

Low value of 'C' tries to make decision surface smooth while a higher value make decision surface fit data better by selecting more support vectors.

'Linear' kernel was also used in experiments, which is of form:

$$K\left(x,x'\right) = \langle x,x'\rangle$$

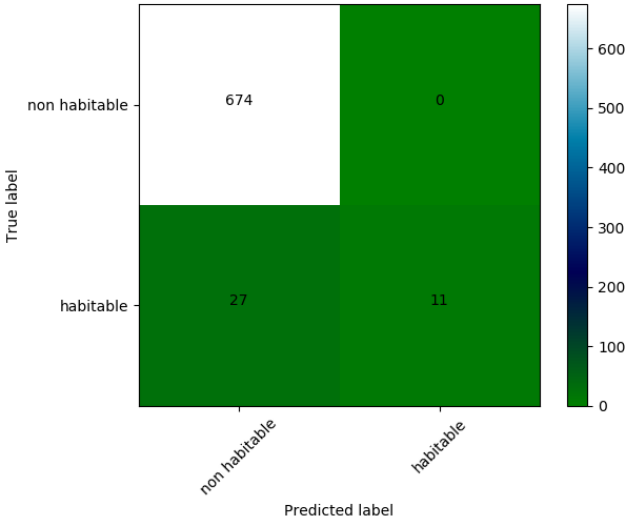
class_weight parameter value 'balanced' was used for scikit-learn [5] SVM API [13] to compensate for the fact that habitable planet training data were very small compared to number of non habitable planet training data.

scikit-learn SVM API [14] were used to implement our model.

4. Experiments

One of the early experiment was with Support Vector Machine classification model and with all of earlier mentioned 14 planetary and stellar features selected.

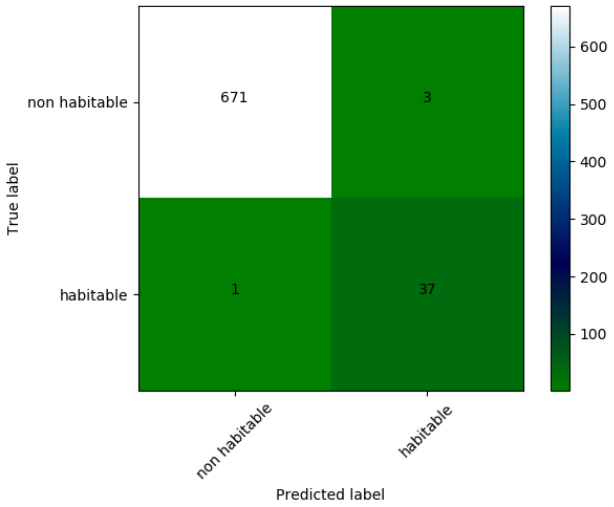
In following diagram which is confusion matrix with all features selected, we can see that almost 70% true habitable planets (27) were predicted as non-habitable, which was unacceptably high.



Next, "forward feature search" with 'rbf' kernel was tried to select most relevant features.

One observation was, depending on distribution of training and dev data, some time, different feature sets were selected by "forward search" in different runs. To find feature set which represented lowest overall dev error, multiple iterations of "forward search" was done, each time randomly reshuffling training and dev data. At the end we selected feature set which gave lowest error on dev data. During this logic, we used 'rbf' kernel.

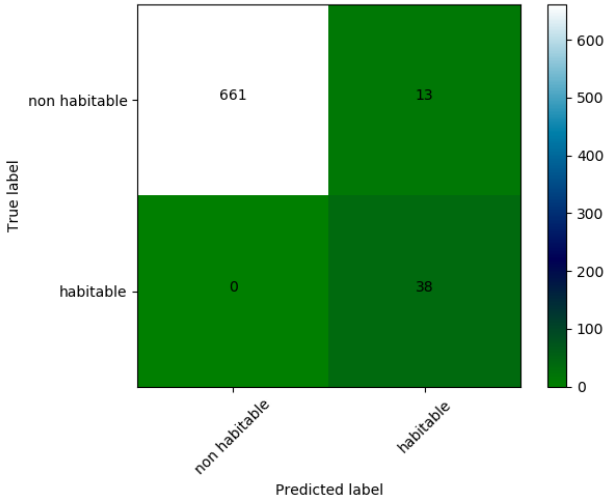
Confusion matrix from one of these runs is below:



Above we can see prediction of habitable planets becomes much more accurate compared with selecting all feature set.

An experiment was also performed to recursively eliminate features with K-fold cross validation using scikit-learn API [11] and with ‘Linear’ kernel. Scikit-learn “recursive elimination of feature” API did not support ‘rbf’ kernel.

Below is confusion matrix with “Recursive elimination of feature” experiment:



Above we can see, ‘Linear’ kernel with feature elimination identified habitable planets with high accuracy, but failure rate was high for non-habitable planets where it inaccurately identified 13 planets as habitable even though they were non-habitable.

Error on training, dev and test data in percentage with “Forward feature search” and ‘rbf’ kernel was

Training	Dev	Test
0.35 ± 0.10	1.0 ± 0.50	1.0 ± 0.25

Error on training and test data in percentage with “Recursive feature elimination with cross validation and Linear kernel”:

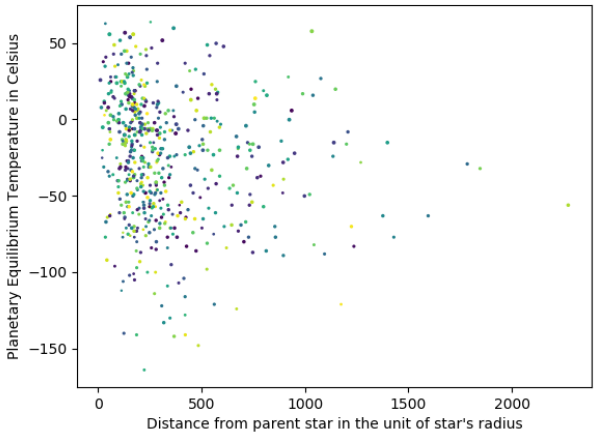
Training	Test
1.5 ± 0.10	1.5 ± 0.75

5. Results

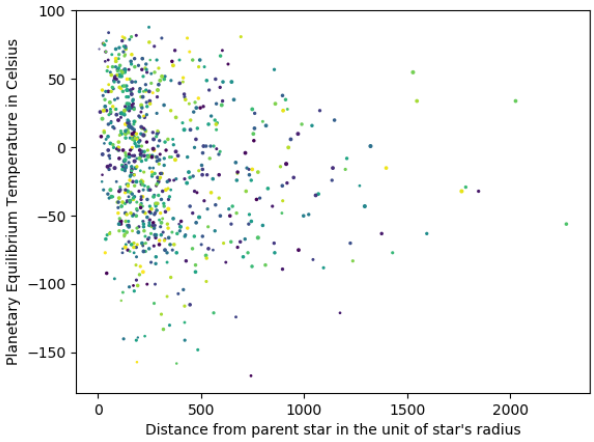
After training our model [6] on training data, we ran our model to do prediction on cumulative Kepler data [15] of planets of all disposition which contained 9564 planets data.

This test was done to see what was the general trend of planets predicated by our model as habitable.

A scatter plot was created for habitable planets predicted with our model. X axis is planet's distance from parent star in the unit of parent star’s radius. Y axis is planetary equilibrium temperature [16] in degree Celsius. Size of scattered points are planet's radius compared to Earth radius. Below scatter plot is with 'rbf' kernel and "Forward feature search" :



Below second scatter plot is with ‘Linear’ kernel and "Recursive feature elimination with K-fold cross validation" of predicted habitable planets on same cumulative data.



Every little speck of dot in these plots are one almost earth size planet.

In these results we can see, 'rbf' kernel predicted smaller number, around 570 planets as habitable, while 'Linear' kernel predicted much higher number, approximately 800 planets as habitable.

This difference in behavior about predicted class from ‘rbf’ and ‘Linear’ kernel was similar to we had seen earlier in confusion matrix on “test data” that 'Linear' kernel tends to identify even non-habitable planets as habitable (higher false positive rate), while 'rbf' kernel misclassified some habitable planets as non-habitable. In other words, Linear kernel made more optimistic prediction thus larger number of planets were predicted as habitable, while ‘rbf’ made less optimistic prediction with lower number.

In both cases we can see a common general trend. We can see there is a high concentration of planets with equilibrium temperature of around $0 \pm 50^{o}C$ and with distance from parent star ranging from 100 unit distance to 400 unit distance.

For comparison, Earth is 215 unit distance from Sun and equilibrium temperature of Earth is around $-13^{o}C$. The actual planetary surface temperature could be higher depending on amount of greenhouse gas effect similar to Earth where mean surface temperature is $27^{o}C$ due to greenhouse [16].

Presence of liquid water is first requirement for planet to be able to support life.

Equilibrium temperature is very important as it tell us whether planet could be in Goldilocks zone of its parent star. Goldilocks zone of Star is a zone which is neither too cold nor too hot and allow water to exists in liquid form and thus have chance of supporting life. Considering the effects of greenhouse gas, temperature range of $0 \pm 50^{o}C$ could place planet in Goldilocks zone.

Being relatively closer to parent star in comparison to parent star’s radius means there is a high probability that these are rocky planets as rocky planets generally form closer to their parent star. We can see this in our Solar system where all inner planets, Mercury, Venus, Earth & Mars are rocky.

Size of these planets were very similar to earth (0.5 to 2 times). Size is very important, as planet much smaller than Earth means it does not have enough gravity to hold onto an atmosphere, and very large size compared to Earth means it is probably a gas giant and may not be suitable to sustain life.

All of above data indicates, features of planets predicated as habitable by our model were very similar to Earth. Since we know Earth supports life, predicted planets could also have high chance of supporting life.

Model's prediction matching real world observation gives us confidence that model developed was on right track.

6. Conclusion

A program representing machine learning model [6] was successfully created using data from Kepler mission and it also predicted habitable planets with characteristic that we expect from habitable planets. SVM turned out to be a good machine learning model for this classification project.

7. Future

Several new similar missions to study exoplanets such as TESS (Transit Exoplanet Search Satellite), JWST (James Webb Space Telescope), WFIRST (Wide Field InfraRed Survey Telescope) are in the pipeline. There would be an explosion of exoplanets data in future. Machine learning would greatly help us in accumulating the existing knowledge about habitability through machine learning model and applying it on new data to quickly narrow down the habitable planet's list. Machine learning model will also become more accurate as and when more training data about habitable planets from new sources become available.

Bibliography

[1] Geoffrey W. Marcya,1, Lauren M. Weissa, Erik A. Petiguraa, Howard Isaacsona, Andrew W. Howardb, and Lars A. Buchhavec Occurrence and core-envelope structure of 1–4× Earth-size planets around Sun-like stars 2013 "http://www.pnas.org/content/111/35/12655.full"
[2] NASA Kepler mission "https://www.nasa.gov/mission_pages/kepler/overview/index.html"
[3] NASA exoplanet archive "https://exoplanetarchive.ipac.caltech.edu/docs/data.html"
[4] Planetary Habitability Laboratory "http://phl.upr.edu/projects/habitable-exoplanets-catalog/data"
[5] Susan E. Thompson et. al. “PLANETARY CANDIDATES OBSERVED BY Kepler.VIII 2017 "https://arxiv.org/pdf/1710.06758.pdf"
[6] Python program that build machine learning model and predict habitability "https://github.com/rkmisra/cs229_project/blob/master/src/predict_habitability.py"
[7] NASA cumulative data "https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative"
[8] Habitable planets list "https://github.com/rkmisra/cs229_project/blob/master/data/habitable_planets_detailed_list.csv"
[9] Non habitable planets list "https://github.com/rkmisra/cs229_project/blob/master/data/non_habitable_planets_confirmed_detailed_list.csv"
[10] NASA Kepler data column definition "https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html"
[11] Recursive feature elimination "http://scikit-learn.org/stable/modules/feature_selection.html#rfe"
[12] scikit learn mathematical formulation "http://scikit-learn.org/stable/modules/svm.html#mathematical-formulation"
[13] Buitinck et al. API design for machine learning software: experiences from the scikit-learn project 2013 "https://arxiv.org/abs/1309.0238"
[14] scikit learn SVM page "http://scikit-learn.org/stable/modules/svm.html"
[15] Cumulative kepler test data "https://github.com/rkmisra/cs229_project/blob/master/data/cumulative_test.csv"
[16] Equilibrium Planetary Temperature "http://www.astro.princeton.edu/~strauss/FRS113/writeup3/"