Towards a Machine Learning Approach for Facilitating Exoplanet Habitability Discovery

Leandro Motta J.
Unilasalle-RJ
Niterói, Rio de Janeiro, Brazil
lemottajr@hotmail.com

Raphael Abreu
Unilasalle-RJ
Niterói, Rio de Janeiro, Brazil
prof.raphael.abreu@soulasalle.com.br

ABSTRACT

We present a machine learning model for identifying habitable exoplanets using a concise set of variables related to planetary location and host star characteristics. The model, trained with the Random Forest algorithm, achieves exceptional performance, with an f1 score of 96.94%, precision of 99.95%, and recall of 94.44% on a 20% test subset. Additionally, we have developed a web application that utilizes this model, enabling astronomers to quickly assess potential habitability based on input variables. This integrated approach expedites the search for life-sustaining exoplanets, providing a valuable tool for researchers in their exploration of the universe.

KEYWORDS

Machine learning, Astronomy, Random Forest, Exoplanets, Habitability.

1 Introduction

Exoplanets are planets that orbit other stars, that is, they are outside our solar system, inside or outside our galaxy. The estimate of existing planets in the universe is 20 sextillion and around 4 sextillion of these planets may be capable of harboring life [4], discovering them is not a trivial task, nor defining whether a given planet is humanly habitable, in all from 1992 to the present day there is a record of 5,000 exoplanets in NASA archives [1] and the searches go on and on. This search for habitable exoplanets has become increasingly popular driven by the incessant desire to find suitable conditions for the existence of molecules that indicate the presence of living organisms. Due to the exponential increase in data generated by space telescopes and its complexity to define in fact the habitability of a planet [3], several variables must be taken into account when analyzing and calculating the condition of an exoplanet to be habitable. Therefore, in order to facilitate the work of astronomers and observers, a machine learning model that interprets such a large amount of data can be very useful [2].

The exoplanets are classified as habitable or non-habitable by the habitability index, this index can be defined through some methods as Earth Similarity Index (ESI), where the researches look for an “Earth-like” planet by a similar bulk composition to Earth, for example [15]. That said, we developed a model that predicts the planet’s chance of habitability and a system that uses this model to help astronomers quickly identify whether a set of variables represents a potential habitable planet, so the astronomer can delve into the internal analysis of the planet's components in a more accurate way.

Our paper will be presented as follows. In section 2 we will present related work. The dataset used will be explained in Section 3. Section 4 encompasses our main contributions on our implementation, mainly the dataset construction and model training. Section 5 will present the model evaluation and Section 6 concludes our work.

2 Related Work

As mentioned in [19], astronomy is going through a moment of great volume of complex data, for this reason, the use of machine learning has become popular among professionals who deal with them constantly. A good example of this is the use of machine learning in the [20] where a classification machine learning model was trained to identify the characteristic of X-ray reverberation near black holes using the Power Spectral Density (PSD) fundament to train and evaluate the model in order to classify the source height and inclination. As evidenced in [23], the next decade will be possible to see a continued rise in data-driven discovery and data volumes grow which will show a great potential on the part of machine learning.

As well, the habitability index, as Earth Similarity Index (ESI) is decisive to find a planet similar to the Earth, the only planet known to date capable of harboring life as we know it. As stated in [21] the ESI ranges are from 1 (Earth) to 0 (totally dissimilar from Earth). The ESI uses some factors to be unveiled, among them are: radius, density, escape velocity and surface temperature, in order to find metrics that resemble those of planet Earth. The latter parameter is a greater challenge to be determined, since it is not an observable value. In this case, as shown in [22], it's necessary to model the surface temperature.

This modeling consists of treating the unknown quantities and using fast climate calculations to explore how variations affect the surface temperature. In view of these researches, it is clear that machine learning models can help a lot in dealing with complex astronomical data.

3 Dataset

The PHL's Exoplanets Catalog (PHL-EC) [8] was chosen for the project, this dataset uses a piece of data from the NASA Exoplanet Archive dataset [7], this one contains more features like number of planets in the planetary system, Jupiter data, proper motion, Parallax, and more. Originally, NASA's dataset had 5,445 samples and around 300 features. The PHL-EC [8] is responsible for providing the habitable feature calculated through the various habitability indexes, such as Earth Similarity Index (ESI), Habitable Zone Distances (HZD) and the Global Primary Habitability, this dataset contains some variables included in the NASA's dataset but uses, as a plus, strategic variables aiming at a greater relationship with the habitability feature. The dataset is available as a comma separated value format, known as CSV.

The dataset has a shape of 4,048 samples and 112 features, which comes to 453,376 instances, the features correspond to stars, habitable zones and planets data, as shown in the Figure 1 and Figure 2, in addition, each feature has an important description [8]. Among these 112 features there is the feature that defines the habitability of the exoplanet and are found as follows: 0 - not habitable; 1 - conservative habitability; 2 - optimistic habitability. However, there is a considerable imbalance between these data, with 3993 values corresponding to 0, 21 equal to 1 and 34 corresponding to 2, this factor will be dealt with in section 4.2.

	P_NAME	P_STATUS	P_MASS	P_MASS_ERROR_MIN	P_MASS_ERROR_MAX	P_RADIUS	P_RADIUS_ERROR_MIN	P_RADIUS_ERROR_MAX	P_YEAR	P_UPDATED	...
0	11 Com b	3.0	6165.86330	-476.74200	476.74200	NaN	NaN	NaN	2007	2014-05-14	...
1	11 UMi b	3.0	4684.78480	-794.57001	794.57001	NaN	NaN	NaN	2009	2018-09-06	...
2	14 And b	3.0	1525.57440	NaN	NaN	NaN	NaN	NaN	2008	2014-05-14	...
3	14 Her b	3.0	1481.07850	-47.67420	47.67420	NaN	NaN	NaN	2002	2018-09-06	...
4	16 Cyg B b	3.0	565.73385	-25.42624	25.42624	NaN	NaN	NaN	1996	2018-09-06	...

Figure 1: **Part of the original PHL-EC data.**

...	P_HABZONE_CON	P_TYPE_TEMP	P_HABITABLE	P_ESI	S_CONSTELLATION	S_CONSTELLATION_ABR	S_CONSTELLATION_ENG	P_RADIUS_EST	P_MASS_EST	P_SEMI_MAJOR_AXIS_EST
...	0	Hot	0	0.083813	Coma Berenices	Com	Berenice's Hair	12.082709	6165.86330	1.29
...	0	Hot	0	0.082414	Ursa Minor	UMi	Little Bear	12.229641	4684.78480	1.53
...	0	Hot	0	0.081917	Andromeda	And	Andromeda	12.848516	1525.57440	0.83
...	0	Cold	0	0.145241	Hercules	Her	Hercules	12.865261	1481.07850	2.93
...	1	Warm	0	0.368627	Cygnus	Cyg	Swan	13.421749	565.73385	1.66

Figure 2: **Another part of the original PHL-EC data.**

4 Implementation

The following steps (Figure 3) was used to implement the work. Data Gathering from the PHL-EC dataset, where it contains important features which indicate the planets' habitability. Upon reaching the Data Preprocessing stage, it was seen that there were several missing values (Figure 4), a little less than 150,000 null values, so the data were treated, passing by noise removal to null treatment, in addition to Feature Selection where through the Select K Best algorithm the model chose the most important features based on correlation with the habitable column from the dataset. After this, oversampling was used to balance the habitable data [9], which had more non-habitable data than habitable. Finally, the model training with clean and organized data. At the end of the process the model was saved and integrated to a system with an friendly interface which connected the user to the model (Figure 5, Figure 6 and Figure 7).

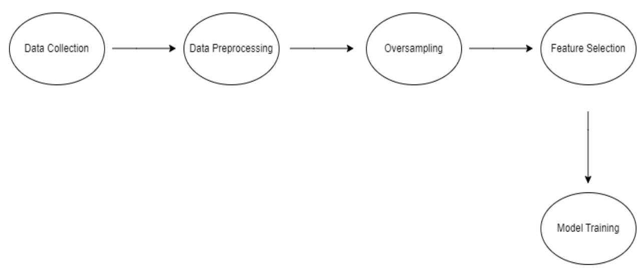

Figure 3: **Project Model Steps.**

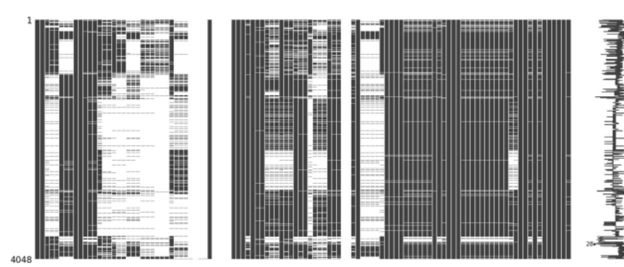

Figure 4: **Missing values in the dataset indicated by blank spaces.**

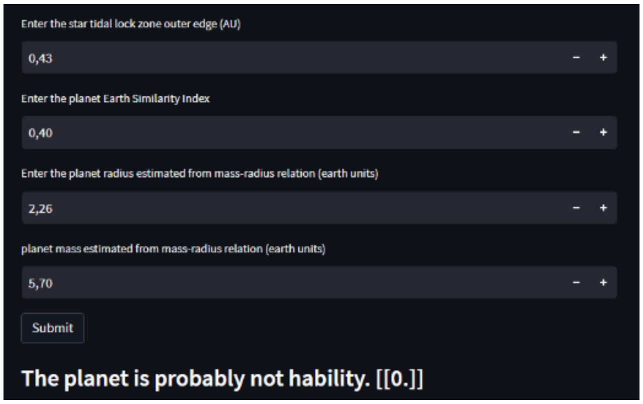

Figure 5: **Not habitable result.**

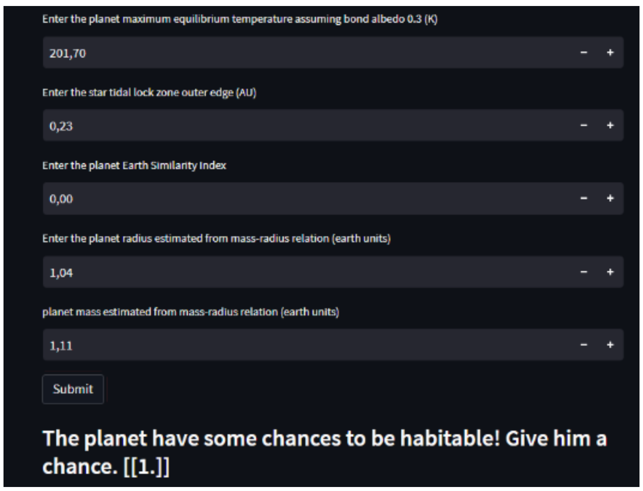

Figure 6: **Conservative habitable result.**

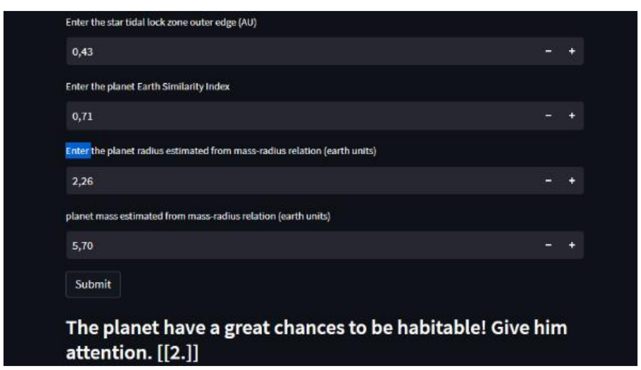

Figure 7: **Optimistic habitable result.**

4.1 Data Preprocessing

Among so much data, some samples have missing data (Figure 4), noises or even incorrect records, because of this it's necessary to do some processing in order to clean the dataset for a better final model result. Therefore, in the first place, we checked the ratio of missing data for each feature to the dataset as a whole (Figure 8) where 0.0 corresponds to no null values and 1.0 corresponds to all null values in that feature. After the analysis, we decided to only keep columns with less than 20% missing values, considering that features with a large percentage of missing values can affect the model performance, in addition, the categorical features will not be used either.

The kept features with missing values were treated with imputation through the process of Iterative Imputer [17], which estimates the missing values based on a machine learning model. As hyperparameters of the method were used the Random Forest Regressor as an estimator, max_iter equal to 3, missing_values passing np.NaN to impute only null values and the verbose equal to 2 just to print messages related to the cell execution.

Finally, a new dataset with the numeric and treated values was created with 49 features (Figure 9). In this new dataset with treated values, some variables will not be used to train the machine learning model, including because of the low correlation between them and the habitability variable (target), so both were dropped. The status variable, which specifies the planet status, if the value is equal to 3, it is a confirmed planet, so all the samples have the value 3 and don't have any kind of correlation with the Habitable feature. The variable year, which specifies the planet's discovered year will be dropped too since it is not relevant information about the planet's situation. Another variable removed from the dataset was the variable which indicates if the planet is in the optimistic habitable zone or not but it can skews the model result since all exoplanets with a habitability index equal to "2" are in the optimistic zone, not giving so much importance to other variables. Finally, the independent variables were assigned to the X dataset and the dependent variable (P_HABITABLE) with the habitability data was assigned to the y.

Some algorithms require that the data be normalized to perform better. Normalizing the data means putting it on a specific scale in order to make the values more comparable and facilitate both training and analysis of the model. The normalization technique used was the min-max scaling, this technique is responsible for transforming the data to a specific range, 0 and 1, in this case.

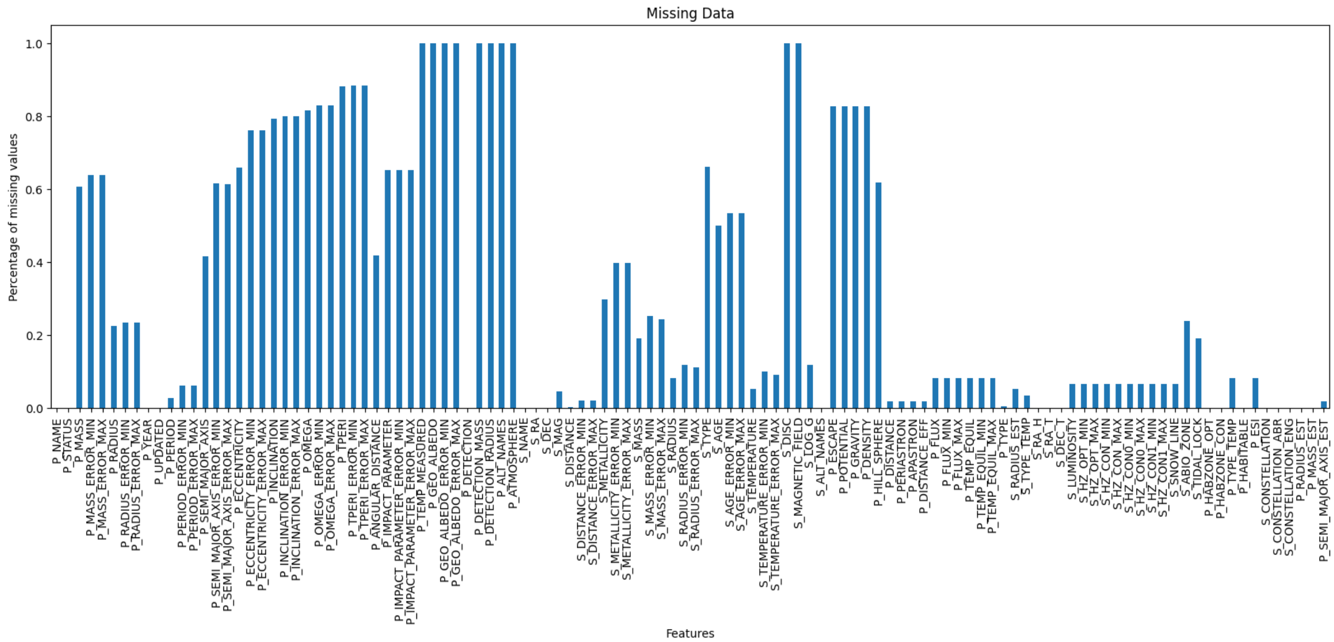

Figure 8: **Percentage of null values for each feature before the Data Preprocessing.**

P_PERIOD_ERROR_MIN	P_PERIOD_ERROR_MAX	S_DISTANCE_ERROR_MIN	S_DISTANCE_ERROR_MAX	S_TEMPERATURE_ERROR_MIN	S_TEMPERATURE_ERROR_MAX	S_RADIUS_ERROR_MIN	S_RADIUS_ERROR_MAX
P_STATUS	P_YEAR	P_PERIOD	S_RA	S_DEC	P_SEMI_MAJOR_AXIS_EST	S_MAG	S_DISTANCE
S_MASS	S_RADIUS	S_TEMPERATURE	S_LOG_G	P_DISTANCE	P_PERIASTRON	P_APASTRON	P_DISTANCE_EFF
P_FLUX	P_FLUX_MIN	P_FLUX_MAX	P_TEMP_EQUIL	P_TEMP_EQUIL_MIN	P_TEMP_EQUIL_MAX	S_RADIUS_EST	S_RA_H
S_LUMINOSITY	S_HZ_OPT_MIN	S_HZ_OPT_MAX	S_HZ_CON_MIN	S_HZ_CON_MAX	S_HZ_CON0_MIN	S_HZ_CON0_MAX	S_HZ_CON1_MIN
S_HZ_CON1_MAX	S_SNOW_LINE	S_TIDAL_LOCK	P_HABZONE_OPT	P_HABZONE_CON	P_HABITABLE	P_ESI	P_RADIUS_EST
P_MASS_EST							

Figure 9: **Features that remained without null values after the Data Preprocessing**

After the most Data Preprocessing step it's time to prepare the clean data for the model training and testing, so the data was splitted into 80% for training and assigned to variables X_train and y_train, the remaining 20% was reserved for testing, assigned to variables X_test and y_test. The purpose of this division is to ensure that the evaluation takes place on data not previously seen by the model. Furthermore, by setting aside a substantial percentage for training, we allow the model to learn complex patterns and relationships from the available data. Test data provides an independent check on performance and helps identify whether the model is overfitting the training data. Finally, it is important that the test set represents the actual data that the model will encounter during deployment.

4.2 Oversampling

We currently have more non-habitable exoplanets than those known to be habitable and this is reflected in the collected dataset. So, even with a large number of data, we do not have balanced data, in this case, the dataset has a lot of data from non-habitable exoplanets, referring to 0 in the feature P_HABITABLE, and few data of habitable exoplanets (Figure 10), for this reason oversampling comes into play. Oversampling is a technique that consists of increasing the amount of minority class instances until the dataset has a balanced amount of data between the classes of the target variable, this method can be with producing new samples or repeating some samples [9].

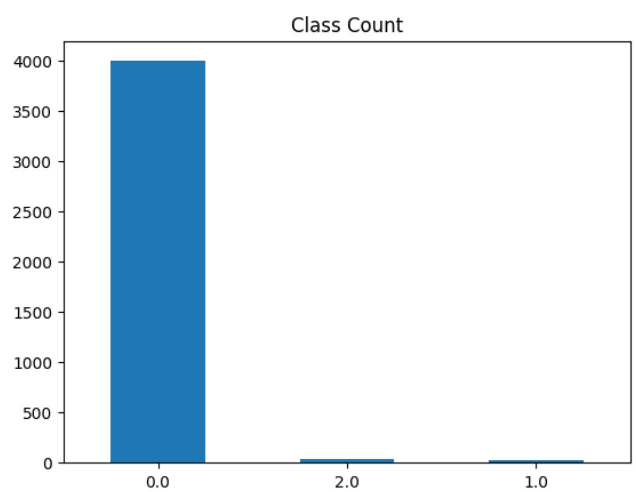

Figure 10: **Graph indicating data imbalance.**

One of the oversampling methods is the Synthetic Minority Over-sampling Technique algorithm (SMOTE), this algorithm obtains new instances through the method of random linear interpolation between a few samples and their neighbor samples. In this way, the data become balanced by generating an effective number of artificial minority class samples, therefore the classification accuracy of the dataset is improved [14]. That said, we instantiate the SMOTE method to the “smote” variable, after that, the “smote” variable was used to fit the training data (X_train and y_train), this procedure was assigned to the X_train_resampled and the y_train_resampled. The testing data remained with the original data, not causing any kind of bias. Figure 11 shows the balanced data after the application of the SMOTE method to the training data.

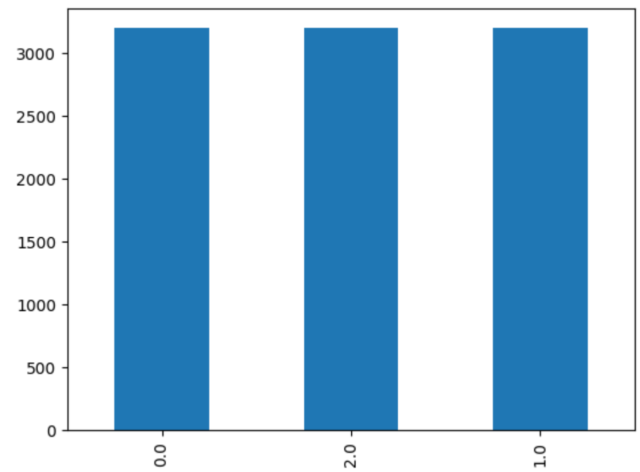

Figure 11: **Graph after the data balancing through the oversampling method.**

4.3 Feature Selection

The next step is the feature selection on training data, this process job is made to choose the best and relevant features for the model based on univariate statistical tests [12], in addition to improving classification accuracy [13] and optimizing the model training. Therefore, the algorithm chose the 13 best variables of the dataset. The algorithm chosen for this step of the model was the Random Forest Classifier due to their performance power, low overfitting and easy interpretability. Furthermore, feature selection using Random Forest comes under the category of Embedded methods that combine filter and wrapper methods. The Embedded method is famous for its highly accurate, better generalization and interpretability [18]. The features chosen as important by the model were: S_DISTANCE, S_MASS, S_TEMPERATURE, P_FLUX, P_FLUX_MIN, P_FLUX_MAX, P_TEMP_EQUIL, P_TEMP_EQUIL_MIN, P_TEMP_EQUIL_MAX, S_TIDAL_LOCK, P_ESI, P_RADIUS_EST, P_MASS_EST.

The description of each one is contained in the PHL'S EXOPLANETS CATALOG page [8].

4.4 Random Forest Model Training

The selected model for data training was the Random Forest Classifier. This model was chosen for some reasons, including: precision and performance, by combining multiple independent decision trees, it reduces bias and variance, resulting in a more robust and accurate model [25]. Also, prevents overfitting. Another reason for using this model is its computational efficiency, the Random Forest algorithm is highly scalable and can handle large datasets efficiently [26]. It can be parallelized and distributed across distributed computing environments, speeding up the training process and allowing the use of larger datasets. As the input, variables chosen in the Feature Selection step were used, they are: 'S_DISTANCE' (represents the distance between the host star and the exoplanet), 'S_MASS' (represents the star mass in solar units), 'S_TEMPERATURE' (means the star temperature in Kelvin), 'P_FLUX' (energy flux from the host star reaching the exoplanet), 'P_FLUX_MIN' (indicates the smallest recorded value of the flux of stellar energy reaching the exoplanet), 'P_FLUX_MAX' (contrary to the previous one, indicates the highest recorded value of the flux of stellar energy reaching the exoplanet), 'P_TEMP_EQUIL' (represents the estimated equilibrium temperature of the exoplanet), 'P_TEMP_EQUIL_MIN' (represents the estimated minimum equilibrium temperature of the exoplanet), 'P_TEMP_EQUIL_MAX' (represents the estimated highest equilibrium temperature of the exoplanet), 'S_TIDAL_LOCK' (This feature indicates whether the exoplanet is in a tidally locked state with its host star. Tidal locking occurs when an exoplanet has a rotation period equal to the orbital period around the star, resulting in one face always facing the star and one face always facing away from it), 'P_ESI' (Earth Similarity Index method), 'P_RADIUS_EST' (planet radius estimated from mass-radius relation), 'P_MASS_EST' (planet mass estimated from mass-radius relation). The only hyperparameter used was the "n_estimator" that defines the number of decision trees that will be created. Finally, the feature importance analysis [24], responsible for quantifying the relative contribution of each feature, was carried out and it was seen that planet estimated radius, planet estimated mass and the planet ESI have a considerable influence on the model while the others have balanced influence (Figure 12).

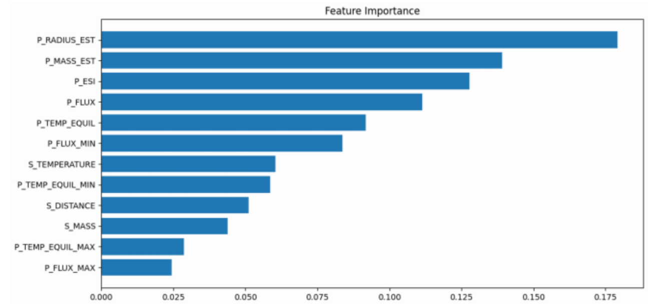

Figure 12: **Feature Importance graph.**

5 Model Evaluation

With the model trained we used some metrics in the validation data for model performance evaluation, they are: Precision, Recall and F1-Score. We will introduce them briefly below in this section. For the equations presented, consider that |TP| represent the True Positives, that is, examples correctly classified as positive by the model, and |FP| is the False Positives, which means the examples erroneously classified as positive by the model. One of the metrics we chose to use is the Precision (Figure 13), this metric measures the proportion of examples correctly classified as positive relative to all examples classified as positive by the model. A high Precision indicates that the model has a low false positive rate, that is, it is able to avoid misclassifying negative examples as positive.

$$precision = \frac{tp}{tp + fp}$$

Figure 13: **Precision formula.**

The other metric chosen was Recall (Figure 14), this one measures the proportion of correctly classified positive examples relative to all examples that are actually positive (true positives + false negatives). The best value is 1 and the worst value is 0 like Precision. Recall is an important metric to assess the model's ability to correctly identify positive examples. A high recall indicates that the model has a low false negative rate, that is, it is able to correctly detect the positive class.

$$recall = \frac{tp}{tp + fn}$$

Figure 14: **Recall formula.**

And finally, the F1-Score (Figure 15) is a performance evaluation measure that combines precision and recall into a single value. It is calculated by taking the harmonic mean of accuracy and recall. The harmonic mean is used to give more weight to lower values, ensuring that the F1-Score is sensitive to low performance in both accuracy and recall. The best value is 1 and the worst value is 0. A higher F1-Score indicates a model with high Precision and recall.

$$F1\ score = 2 * \frac{Precision\ * Recall}{Precision + Recall}$$

Figure 15: **F1-Score formula.**

With this in mind, we can finally define that we got a Test Accuracy equal to 99.87%, furthermore, a F1-Score of 96.94%, a precision of 99.95% and a recall of 94.44% were achieved. The model was wrong in predicting some data, but it was within a very acceptable margin of failure (Figure 16).

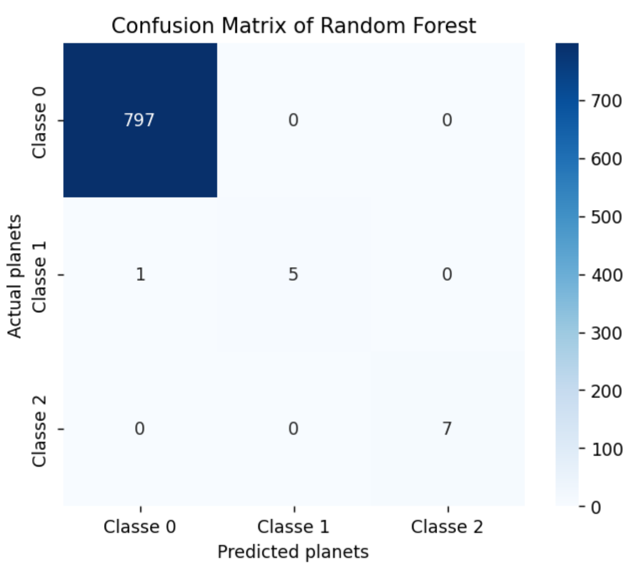

Figure 16: **Confusion matrix of the test performed on the validation data.**

For purposes of comparison with the Random Forest algorithm, we trained a Convolutional Neural Network using the same dataset and the same data division, with the difference that in the training of this model normalized data were used. The Network architecture relied on Keras [28] and Tensorflow [27]. To train the neural network, 50 epochs and 31 iterations were used for each epoch, that is, batch_size with a value of 32, the optimizer algorithm used was the Adaptive Moment Estimation (Adam). To validate the performance of the model, the same metrics (Accuracy, Precision, Recall and F1-Score) were used on the test data, referring to 20% of the percentage of the entire dataset. The results of the Network were as follows: Test Accuracy equal to 98.89%, an accuracy of 75.55%, Recall of 88.59% and, finally, F1-Score equal to 78.64%.

6 Conclusion

In view of the results achieved, it can be considered that the use of machine learning to facilitate the work of professionals who deal with an abundant load of exoplanets data on daily basis is completely possible and seeks to optimize the search time for habitability on new planets, with this the professional can put efforts to study more deeply the atmosphere of planetas indicated by the model. Furthermore, it is possible to improve the performance of this solution with a more in-depth study of the thousands of variables used for the study of new planets and also for new variables discovered over time and model re-training. Another point for the future is to create a subset to train the model with specific features so that it’s possible to find habitable exoplanets even if they don’t have much similarity with the Earth in terms of metrics.

REFERENCES

[1] Christiansen, J. L. Five thousand exoplanets at the NASA Exoplanet Archive. *Nat. Astron.* 6, 516–519 (2022).

[2] Baron, D. (2019). *MACHINE LEARNING IN ASTRONOMY: A PRACTICAL OVERVIEW*.

[3] Lammer, H., Bredehöft, J.H., Coustenis, A. *et al.* What makes a planet habitable?. *Astron Astrophys Rev*17, 181–249 (2009).

[4] Jones, Brent M. *How many planets are there in the universe?[PowerPoint]*. No. SAND2015-9919PE. Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2015.

[5] Geng, Xiubo, et al. "Feature selection for ranking." *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. 2007.

[6] Wang, Juanjuan, et al. "Classification of imbalanced data by using the SMOTE algorithm and locally linear embedding." *2006 8th international Conference on Signal Processing*. Vol. 3. IEEE, 2006.

[7] Planetary Systems. Disponível em: <https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=PS>.

[8] PHL @ UPR Arecibo - PHL’s Exoplanets Catalog. Disponível em: <https://phl.upr.edu/projects/habitable-exoplanets-catalog/hec-data-of-

potentially-habitable-worlds/phls-exoplanets-catalog>. Acesso em: 19 jun. 2023.

[9] Mohammed, Roweida, Jumanah Rawashdeh, and Malak Abdullah. "Machine learning with oversampling and undersampling techniques: overview study and experimental results." *2020 11th international conference on information and communication systems (ICICS)*. IEEE, 2020.

[10] Planetary Systems and Planetary Systems Composite Parameters Data Column Definitions. Disponível em: <https://exoplanetarchive.ipac.caltech.edu/docs/API_PS_columns.html>.

[11] Liu, Haoyue, MengChu Zhou, and Qing Liu. "An embedded feature selection method for imbalanced data classification." *IEEE/CAA Journal of Automatica Sinica* 6.3 (2019): 703-715.

[12] 1.13. Feature selection — scikit-learn 0.21.3 documentation. Disponível em: <https://scikit-learn.org/stable/modules/feature_selection.html>.

[13] Post, Martijn J., Peter Van Der Putten, and Jan N. Van Rijn. "Does feature selection improve classification? a large scale experiment in OpenML." *Advances in Intelligent Data Analysis XV: 15th International Symposium, IDA 2016, Stockholm, Sweden, October 13-15, 2016, Proceedings 15*. Springer International Publishing, 2016.

[14] Wang, Shujuan, et al. "Research on expansion and classification of imbalanced data based on SMOTE algorithm." *Scientific reports* 11.1 (2021): 1-11.

[15] Schulze-Makuch, Dirk, et al. "A two-tiered approach to assessing the habitability of exoplanets." *Astrobiology* 11.10 (2011): 1041-1052.

[16] Zhang, Shichao, Xindong Wu, and Manlong Zhu. "Efficient missing data imputation for supervised learning." *9th IEEE International Conference on Cognitive Informatics (ICCI'10)*. IEEE, 2010.

[17] sklearn.impute.IterativeImputer. Disponível em: <https://scikit-learn.org/stable/modules/generated/sklearn.impute.IterativeImputer.html#:~:text=impute%20import%20IterativeImputer->.

[18] Saeys, Yvan, Thomas Abeel, and Yves Van de Peer. "Robust feature selection using ensemble feature selection techniques." *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2008, Antwerp, Belgium, September 15-19, 2008, Proceedings, Part II 19*. Springer Berlin Heidelberg, 2008.

[19] Baron, Dalya. "Machine learning in astronomy: A practical overview." *arXiv preprint arXiv:1904.07248* (2019).

[20] Chainakun, P., et al. "Machine learning application to detect light echoes around black holes." *Monthly Notices of the Royal Astronomical Society* 506.4 (2021): 5318-5327.

[21] Kashyap Jagadeesh, Madhu, et al. "Indexing of exoplanets in search for potential habitability: Application to Mars-like worlds." *Astrophysics and Space Science* 362.8 (2017): 146.

[22] Vladilo, Giovanni, et al. "Modeling the surface temperature of earth-like planets." *The Astrophysical Journal* 804.1 (2015): 50.

[23] Ntampaka, Michelle, et al. "The role of machine learning in the next decade of cosmology." *arXiv preprint arXiv:1902.10159* (2019).

[24] Zien, Alexander, et al. "The feature importance ranking measure." *arXiv preprint arXiv:0906.4258* (2009).

[25] Breiman, Leo. "Random forests." *Machine learning* 45 (2001): 5-32.

[26] Ali, Jehad, et al. "Random forests and decision trees." *International Journal of Computer Science Issues (IJCSI)* 9.5 (2012): 272.

[27] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, ´ S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., ´ Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[28] Chollet, F. et al. (2018). Keras: The python deep learning library.