KU LEUVEN

GROUP T LEUVEN CAMPUS

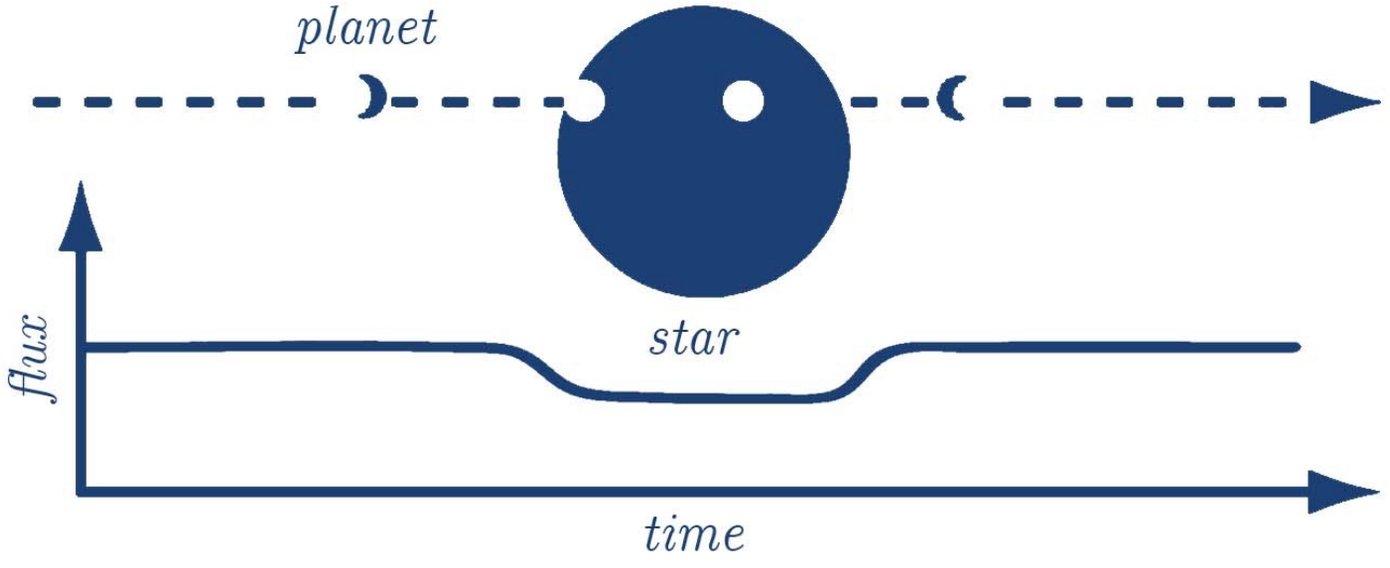
Exoplanet Hunting in Deep Space

ML Project

ALKEN RROKAJ, FATJON BARÇI

ABSTRACT

Exoplanets are planets that orbit stars other than the Sun, their detection has gained significant attention due to the possibility of discovering extraterrestrial. In this study, machine learning techniques were used to analyze an exoplanet detection dataset consisting of time-series data of exoplanet and non-exoplanet stars, with the goal of identifying which model performs the best. Logistic regression and neural network models were tested, and while the results showed promise, further research is needed to find a more accurate model. Suggested methods such as data augmentation, unsupervised learning using deep auto-encoders and clustering, and feature selection using principal component analysis. Overall, this project was a valuable learning experience about machine learning, and unbalanced data sets.



INTRODUCTION

Exoplanets are planets that orbit stars other than our Sun. The search for exoplanets has been an active area of research for decades, and has gained significant attention in recent years with the discovery of many new exoplanets, and the potential for discovering habitable extraterrestrial life and gaining a deeper understanding of the universe.. NASA's Kepler space telescope is one of the primary tools used to search for exoplanets, and has discovered thousands of exoplanets to date.

The motivation for our research lies in the potential for machine learning to enhance our understanding of exoplanets and the universe as a whole. By developing more sophisticated models, we hope to not only detect the presence of exoplanets around stars, but also to determine the number and size of those planets. Additionally, our passion for space and the cosmos also played a role in our selection of this particular dataset.

One method for detecting these exoplanets is by observing the regular dimming of the flux, or light intensity, of a star over a period of time. This phenomenon is caused by a planet passing in front of the star. The input to our algorithm is a time-series dataset consisting of exoplanet and non-exoplanet stars, with 3198 samples of the flux for each observation. The output of our algorithm is a prediction of whether each observation corresponds to an exoplanet-star or a non-exoplanet-star.

We will be using logistic regression and neural network (NN) models to analyze this dataset, with the goal of identifying which model performs the best at this task. We will test each model individually, using the same dataset and evaluation metric, in order to compare their performance.

RELATED WORK

As we got more familiar with the course of "Machine Learning" and with the related works, it became increasingly apparent that the dataset we were working with would pose a significant challenge. In this section, we will discuss the related work we encountered in Kaggle notebooks.

The original publisher of the Kaggle data attempted to build a classification algorithm using various approaches. These approaches include using a 1-D CNN in Torch7, XGBoost in R, and PCA in Python. However, none of these methods have yielded particularly strong results thus far. (Winterdelta, 2019)

A. Pandey evaluated several machine learning models for exoplanet classification, including the MLP Classifier, Gradient Boosting Classifier, and Light GBM Classifier. These models achieved the highest accuracy, but unfortunately classified all exoplanet candidates as non-exoplanets. In contrast, the logistic regression model was able to correctly classify 2 out of 5 exoplanet stars, but also resulted in 246 false positives.

J. Szturemski evaluated the performance of two machine learning models, random forest and logistic regression, on a classification problem. He found that the random forest model performed well, likely due to its use of decision trees, which are well-suited for classification tasks. Szturemski also applied the SMOTE data augmentation method, which may have influenced the results and potentially reduced their naturalness.

DATASET AND FEATURES

According to the Kaggle Data Card, this data analyzed is a cleaned version of the observations of the NASA Kepler space telescope. NASA applied de-noising algorithms to remove artifacts generated by the telescope. This data is chosen because "it was felt" that there are no wrongly labelled exoplanets.

Column 1 is the label vector and Columns 2 - 3198 are the flux values over time. The data is binary labelled with 2 indicating an exoplanet, and 1 indicating no exoplanet.

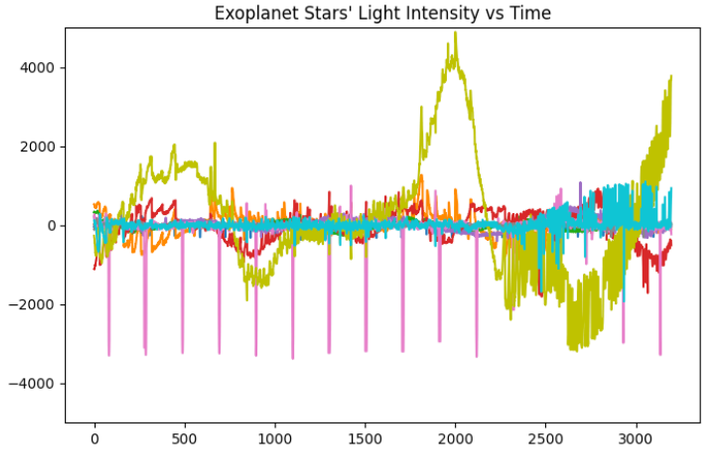
Train Set

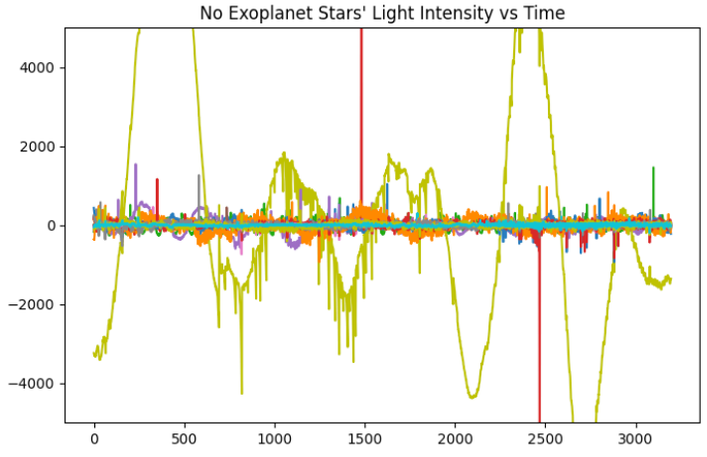
- 5087 star observations
- 3198 features
- 37 exoplanet-stars
- 5050 non-exoplanet-stars

Test Set

- 570 star observations
- 3198 features
- 5 confirmed exoplanet-stars
- 565 non-exoplanet-stars

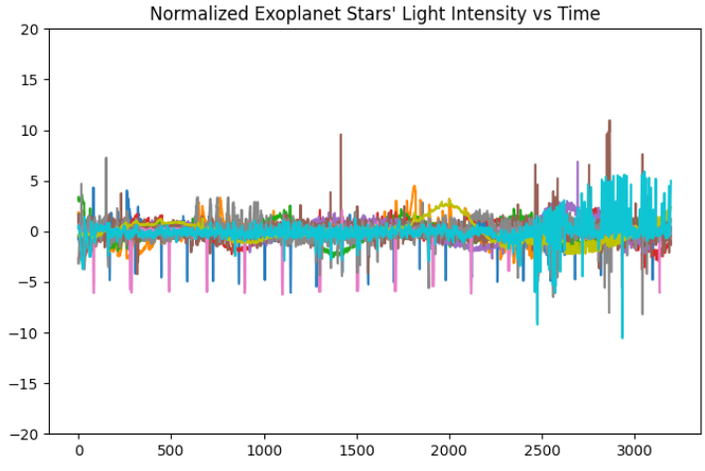
LABEL	FLUX.1	FLUX.2	FLUX.3	FLUX.4	FLUX.5	FLUX.6	...
1	93.85	83.81	20.10	-26.98	-39.56	-124.71	...
2	-38.88	-33.83	-58.54	-40.09	-79.31	-72.81	...

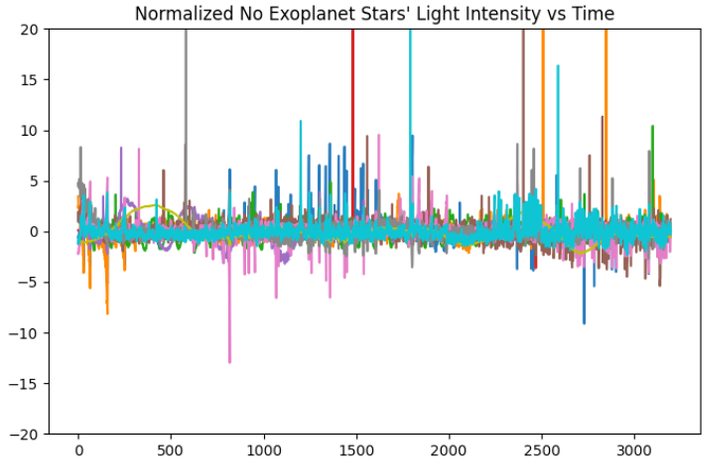




NORMALIZATION

The relative magnitude of the flux depends on the distance from the observer rather than the presence or absence of exoplanets. Therefore, we applied normalization function from the assignments to remove the effect of distance.





Given the large number of features, we considered downsampling the data to simplify the analysis. However, we determined that this approach could result in the loss of important information about the variability and trends in the data, and therefore decided against it.

METHODS

Multiple machine learning algorithms can be used to analyze the data from the Kepler space telescope in order to classify stars as exoplanet-hosting or non-exoplanet-hosting. The two that we are going to explore in this paper are neural networks and logistic regression. In addition, data balancing and data normalization were also pre-applied to the dataset.

DATA BALANCING

To address imbalanced class distribution in the dataset, the Synthetic Minority Oversampling Technique (SMOTE) was employed using the `imblearn` library in Python. The SMOTE algorithm generates synthetic examples of the minority class to balance the class distribution in the dataset. The `random_state` parameter was set to ensure reproducibility of the synthetic examples generated by SMOTE.

LOGISTIC REGRESSION

Logistic regression is another type of machine learning algorithm that can be used for classification tasks. It is a type of linear model that is used to predict the probability that an input belongs to a certain class. The prediction made by logistic regression is a probability between 0 and 1, with values closer to 1 indicating a higher likelihood of belonging to the positive class (in this case, exoplanet-hosting stars) and values closer to 0 indicating a higher likelihood of belonging to the negative class (non-exoplanet-hosting stars).

The prediction made by logistic regression is based on a linear combination of the input features, which are multiplied by weights and summed to produce a linear output. This output is then passed through a sigmoid function, which maps the linear output to a probability between 0 and 1. The weights of the model are adjusted during training in order to minimize the error between the predicted probability and the true class label.

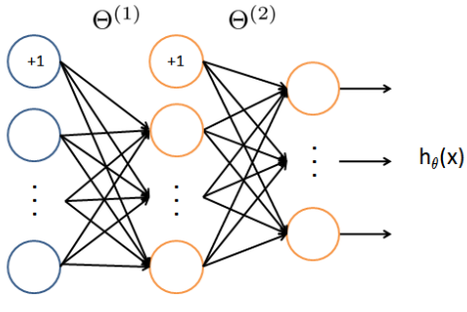
Mathematically, logistic regression can be represented as:

$$p = \sigma(w_0 + w_1x_1 + w_2x_2 + \ldots + w_nx_n)$$

where p is the predicted probability, σ is the sigmoid function, w_0 is the bias term, and $w_1, w_2, \ldots, w_n$ are the weights for the input features $x_1, x_2, \ldots, x_n$.

NEURAL NETWORKS

Neural networks are a type of machine learning algorithm that is inspired by the structure and function of the human brain. They are composed of interconnected nodes, or "neurons," that process and transmit information. In a neural network, input data is passed through multiple layers of interconnected neurons, with each layer applying weights to the input data and generating an output that is passed on to the next layer. The final layer of the neural network produces the output of the network, which is the prediction made by the model.



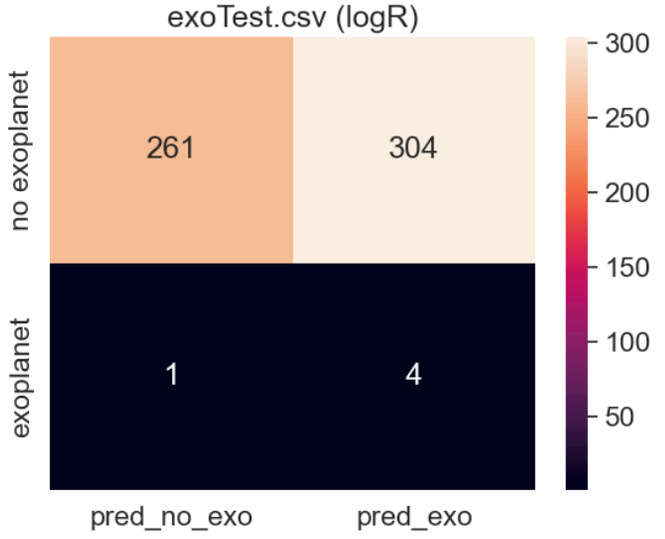
A. Ng., Neural Network ex.4

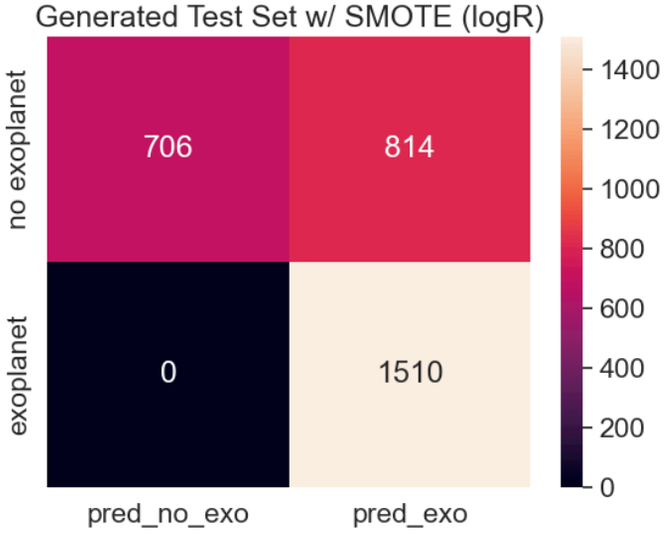
The process of training a neural network involves adjusting the weights of the connections between neurons in order to minimize the error between the predicted output and the true output. This is done using an optimization algorithm, such as stochastic gradient descent, which adjusts the weights based on the gradient of the error function with respect to the weights.

RESULTS/DISCUSSION

LOGISTIC REGRESSION

The results of the logistic regression model on the test set showed an overall accuracy of 46.49%. Despite the use of SMOTE to balance the classes in the dataset, the model struggled to accurately classify samples from both classes. The model's precision for class 1 was 100%, indicating that all of the samples it predicted as class 1 were actually class 1. However, the precision for class 2 was very low at 1%, suggesting that only a small fraction of the samples predicted as class 2 were actually class 2. Similarly, the model's recall for class 1 was 46%, while its recall for class 2 was 80%. The f1-score for class 1 was 63%, while the f1-score for class 2 was 3%. These results suggest that the model is not performing well for either class and further efforts may be needed to improve its performance. This could involve trying different model architectures, tuning the hyper-parameters, or applying additional techniques for model optimization. It is also possible that the dataset itself may contain factors that are challenging for the model to learn, such as noise or structural complexity.



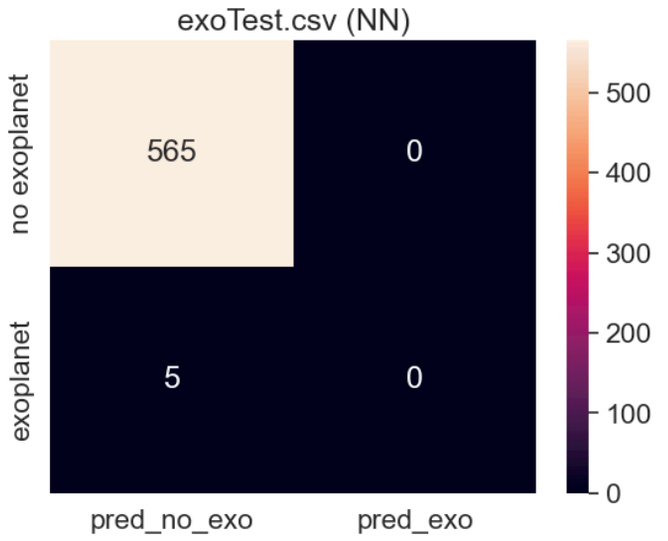


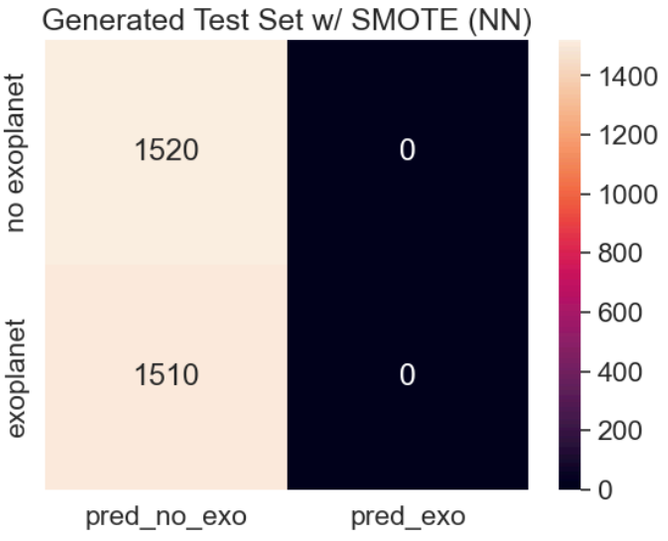
NEURAL NETWORKS

The results for the neural network show that the model achieved a high overall accuracy of 99% on the test set, with 565 true positive and 0 false negative predictions. However, the model struggled to classify samples from class 2, as indicated by the 0 true positive and 5 false negative predictions. This is reflected in the low precision and recall values for class 2, as well as the low f1-score. The model's precision for class 2 is 0, which means that none of the samples it predicted as class 2 were actually class 2. The model's recall for class 2 is also 0, which means that it did not correctly classify any of the samples from class 2. The model's f1-score for class 2 is also 0, which is the harmonic mean of the precision and recall values.

The neural network used in this model consisted of an input layer with 128 neurons and an input dimension of 3198, a hidden layer with 256 neurons, and an output layer with one neuron. The Adam optimization algorithm was used to train the model, and the ReLU activation function was used in all layers except the output layer, where the sigmoid activation function was used. The number of epochs was chosen to be 20, as there was no improvement in the model's performance beyond this point.

One possible reason for the poor performance of the model on class 2 could be the imbalanced nature of the dataset and the lack of clear patterns in the data. Further efforts may be needed to improve the model's performance, such as collecting more data, trying different model architectures, or tuning the hyper-parameters. It may also be helpful to explore the characteristics of the samples from class 2 and identify any potential challenges that the model is facing in learning to classify these samples.¶





CONCLUSION

In conclusion, the analysis of this exoplanet detection dataset was a valuable and informative experience. While this study showed promise in using machine learning techniques the results were not completely accurate. The opportunity to work with a dataset featuring a high disproportion of the target data allowed for the learning of new methods, such as re-sampling, that can be applied to future analyses. Overall, this project was a valuable and enjoyable learning experience that has further sparked our interest in the universe.

Further research is needed to find a more accurate model. These could include data augmentation through the systematic shifting of rows and adding noise to create synthetic trends, using deep auto-encoders and clustering for unsupervised learning and applying principal component analysis to select important flux components.

CONTRIBUTIONS

Fatjon was responsible for tasks such as class balancing, code and algorithm implementation, and writing parts of the discussion. Alken contributed to data normalization, data visualization, writing the remainder of the paper, as well as researching various methods. The team worked collaboratively and effectively to complete the project, with both members making important contributions to the of the project.

REFERENCES

Cover Photo Based On: *Light curve of a planet transiting its star - exoplanet exploration: Planets beyond our solar system* (2022) *NASA*. NASA. Available at: https://exoplanets.nasa.gov/resources/280/light-curve-of-a-planet-transiting-its-star/

Szturemski, J. (2020) *Jean Szturemski - exoplanet hunting*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/jeanszturemski/jean-szturemski-exoplanet-hunting (Accessed: December 19, 2022).

Pandey, A. (2022) *Classification with Classic ML approach*, *Kaggle*. Kaggle. Available at: https://www.kaggle.com/code/anantpandey29/classification-with-classic-ml-approach (Accessed: December 19, 2022).

Turol, S. (2019) *How NASA uses artificial intelligence to detect exoplanets*, *Altoros*. Available at: https://www.altoros.com/blog/how-nasa-uses-artificial-intelligence-to-detect-exoplanets/ (Accessed: December 19, 2022).

Winterdelta (2019) *Winterdelta/Keplerai: Machine Learning Project to discover exoplanets.*, *GitHub*. Available at: https://github.com/winterdelta/KeplerAI (Accessed: December 19, 2022).