Exoplanet Classification

Maikel Sousa (m20200735@novaims.unl.pt), David Sotto-Mayor Machado (m20201023@novaims.unl.pt), Catarina Moreira (m20201034@novaims.unl.pt)

Abstract: Ever since the dawn of humanity, a doubt has been documented even by the archeological findings of extinct civilizations: Are we alone in the universe? Such question is still unanswered to this day, but NASA has devoted many resources to keep searching for the answer. Using data from NASA Exoplanet Archive, single-label and multi-label multi-class classification models are developed to classify new findings by the telescope network. Initial accuracy results raised questioning among the developers, for fear of being over optimistic, which motivated the attempt to find any difference between the single-label and multi-label approaches. This research doesn't answer if we are unique in the universe, but if the truth is out there, a deep learning model is able to classify the possible exoplanets

Keywords: Deep Learning, Single Label Classification, Exoplanets, Keppler Objects of Interest

I. Introduction

Man has, since the dawn of humanity looked up and tried to realize if he was alone in the universe, as proven by the depiction of otherworldly beings in pre-historic paintings (Gulliford, 2018).

Unrecognized as a legit field of science for long, many were the scientists making open references to the search for extraterrestrial intelligence (SETI). In 1899, Nikola Tesla while conduction experimentation on electric wiring would notice that under certain alignments of Mars, he would detect static interference on his transmission system which made him state those were transmissions being made from the red planet (Seifer, 2016).

Later, in mid twentieth century, the space race would be underway, and Astrobiology would become a recognized field of science, especially when American National Aeronautics and Space Administration (NASA) instated that department.

Decades later, in the 80s, NASA would launch Kepler mission, deploying a radar in space searching objects of interest *"out there"* (Keppler Objects of Interest – KOI) (NASA, 2021).

Exoplanet would become a term that refers to a planet like Earth, orbiting a sun-like star but not belonging to our solar system.

Reducing the result space, the term *"Habitable Zone"* refers not to places where life would definitely be sustainable for its validation is still impossible, but to planets whose distance to the star they orbit, and the consequent likely atmospheric temperature is in an interval that allows the hypothesis thar they present water in liquid state.

Should the intention be to evacuate the planet after a catastrophic ecological rupture the knowledge of a most likely suitable planet would still not solve humanity emergency as current technology wouldn't allow travelling that far or reach the target in the lifespan of a human being, or keeping the human being that far under suspended animation for that long.

Figure 1 lists the Exoplanet Missions so far by NASA and ESA.

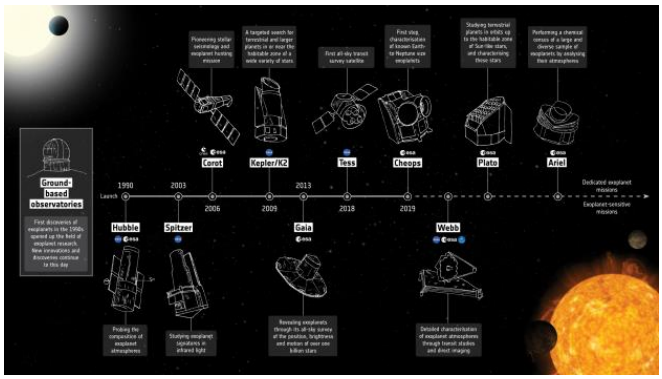

Figure 1 – NASA and ESA Exoplanet Missions Timeline, extracted from (ESA, 2021)

This paper covers KOI data obtained by NASA's Kepler Missions (Kepler/K2).

Figure 2 exhibits statistics describing the results of these missions.

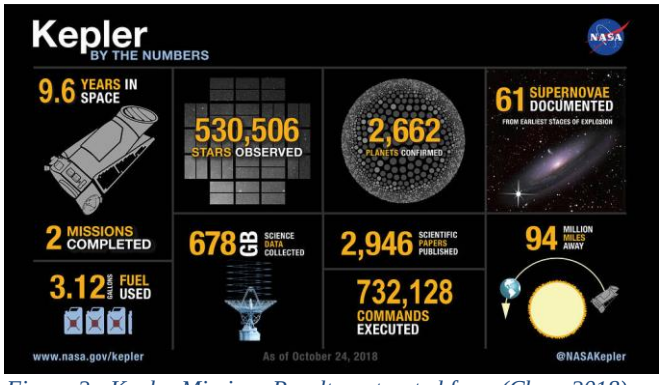

Figure 2 - Kepler Missions Results, extracted from (Chen, 2018)

The 530506 stars observed were inside a grid of 21 areas observed on a rotation every few hours to detect transits (celestial bodies moving in front of a star). The 2662 confirmed planets were obtained

between a pool of 9564 objects of interest that ended being classified as confirmed, candidates or false positives.
All confirmed correspond to a named planet, but 2 false positives have been named while believed they were confirmed planets.

II. Background

If one fits a classification problem of a single label *l* from a set of disjoint labels *L* with $|L| > 2$ then such problem is classifiable as a single-layer multi-class classification problem (Tsoumakas and Katakis, 2007).
Deep Learning introduced the capacity to perform automated machine learning tasks of multiclass classification (Wever et al., 2018). It's uses have included automated Alzheimer diagnostics (Ramzan et al., 2019), breast cancer detection (Nawaz et al., 2018), gene expression (Hamena and Meshoul, 2018), skin lesions (Iqbal et al., 2021), and many non-medical issues like pest detection (Liu et al., 2019) e.g.

III. Methodology

The project follows the practical methodology recommended by Goodfellow, Bengi & Courville (Goodfellow et al., 2016).
Figure 3 represents the workflow.

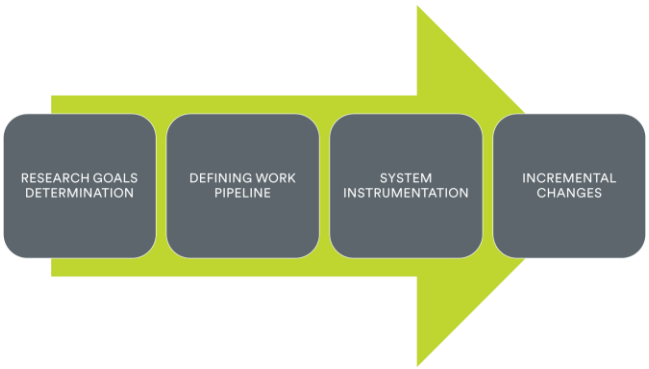

Figure 3 - Practical Methodology, adapted from (Goodfellow et al., 2016)

The investigation question is which deep learning algorithm will better suit the classification of a Kepler Mission's Key Object of Interest Disposition?
The objective is the creation of a model that learns from NASA's classification of Kepler's KOI to classify observations from future missions, provided the same features are registered.
Accuracy was chosen as the model evaluation metric to which a goal was set of 95% or higher. Processing time will be a tiebreak criterion if two models are too close accuracy wise.
This will allow combining effectiveness and efficiency.

The modelling process is presented in figure 4.

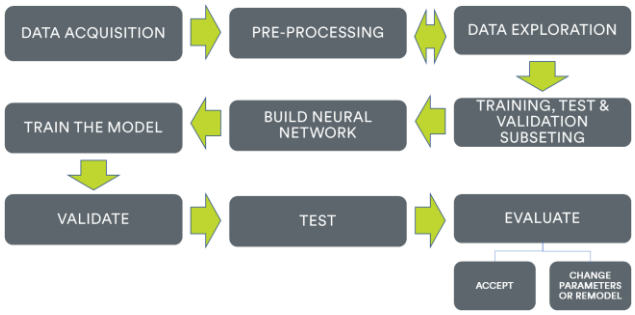

Figure 4 - Modeling Process

Data Acquisition: The dataset chosen for this project was the KOI Table (cumulative list) from NASA's Exoplanet Archive ("Kepler Objects of Interest," 2018). This dataset was last updated on September 2018 and its features are listed and described on the Exoplanet Archive[1]. The features can be grouped as:

- Identification Columns;
- Exoplanet Archive Information;
- Project Disposition Columns;
- Transit Properties;
- Threshold-Crossing Event (TCE) Information;
- Stellar Parameters;
- KIC Parameters;
- Pixel-Based KOI Vetting Statistics.

Pre-Processing: Data exploration allowed realizing that *koi_longp* and *koi_sage* are features with 100% of Nan values. It was also possible to detect that *koi_tce_delivname* and *koi_disp_prov* are of the string type, describing other features, and have no relevance for the model.
The column *koi_disposition* is the labels of the dataset containing the values: "CONFIRMED"; "CANDIDATE"; "FALSE POSITIVE". This classification is dependent of the object disposition tests and confirmed indicate an object who passed all disposition tests, a candidate has passed all tests so far, but not all tests have been concluded a false positive has failed at least one of the tests, what can happen when: 1) the KOI is in reality an eclipsing binary star, 2) the Kepler light curve is contaminated by a background eclipsing binary, 3) stellar variability is confused for coherent planetary transits, or 4) instrumental artifacts are confused for coherent planetary transits.
Column *koi_pdisposition* was dropped for being considered redundant as *koi_disposition* delivers the same information after being further processed.

[1] https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html

The remaining numeric *Nan* values were dealt with using KNN Imputer (k=5 for multi-label, k=3 for single-label) on the following features: *koi_score, koi_smass, koi_srad, koi_smet, koi_slogg, koi_steff, koi_tce_plnt_num, koi_model_snr, koi_insol, koi_teq, koi_prad, koi_ror, koi_depth, koi_impact,* and *koi_eccen*.
After this process a single record was dropped on the multi-label approach for having a *Nan* value in the feature *koi_kepmag*.

The target feature for the single-label model was encoded so CONFIRMED and CANDIDATE values would be 1 and FALSE_POSITIVE would be 0.

On the multi-label approach OneHotEncoding was used to split the original *koi_disposition* in:

- *koi_disposition_CONFIRMED;*
- *koi_disposition_CANDIDATE;*
- *koi_disposition_FALSE_POSITIVE.*
- .

The Data Exploration and Pre-Processing stages have a bi-directional flow as several iterations of each are needed until a satisfactory dataset is ready to feed the model.

Training, Validation and Test sub-setting: Regarding the single-label approach 50% of the original dataset was reserved for testing. The remaining 50% are split into 80% for training and 20% for validation which results in a distribution of 40% of the original dataset for training, 10% for validation and 50% for testing. As for the multi-label approach the initial split was of 30% for testing and for the second split the proportion of 70/30 % was maintained resulting in a final distribution of 56% for training, 14% for validation and 30% for testing. StandardScaler is fitted to the training sub-set and the transformation is applied to both the training and test sets.

The single-layer architecture consists of three dense layers receiving and input with the same number of dimensions as the input dataset after pre-processing, excluding the label feature. The input and hidden layer output 64 dimensions each and have a *relu* activation function to introduce non-linearity. The output layer returns one dimension and has a *sigmoid* activation function to produce binary a result. The model's weights are initialized with an uniform distribution and use *binary_crossentropy* as loss function and *adam* as optimizer. As already stated, the metric used to evaluate all models is the accuracy.

Figure 6 illustrates the single-layer multi-class model.

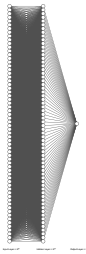

Figure 5- Single-Layer Multi-Class Architecture

On the multi-label approach, four network designs were assessed to transform an input of 26 dimensions in an output of three.

Model 1
A model with two dense layers: the first an input layer that receives the 26 dimensions of the dataaset and outputs 32 dimensions, and the output layer that transforms the 32 dimensions that the input layer outputs into a three-dimensional final output. It uses an *he_uniform* weights initializer (draws samples from a uniform distribution within [-sqrt(6 / #inputs), sqrt(6 / #inputs)]), *relu* activation on the input layer and *sigmoid* activation on the output layer. Uses *binary_crossentropy* as loss function and *adam* as optimizer.

Model 2
In this model the input layer outputs a 64 dimensions intermediate output, and there is a hidden layer with the same intermediate output. The output layer uses softmax activation which produces a categorical probability, the other layers keep using *relu*. Uses *categorical_crossentropy* as loss function.

Model 3
This model uses four layers with intermediate outputs of 100, 64 and 50. The other differences to the architecture of Model 2 is the use of sigmoid as output layer activation and binary_crossentropy as the model's loss function.

Model 4
Similar to Model 2 but with intermediate outputs of 100 and 64 dimensions.

Figure 6 illustrates the four models' architectures.

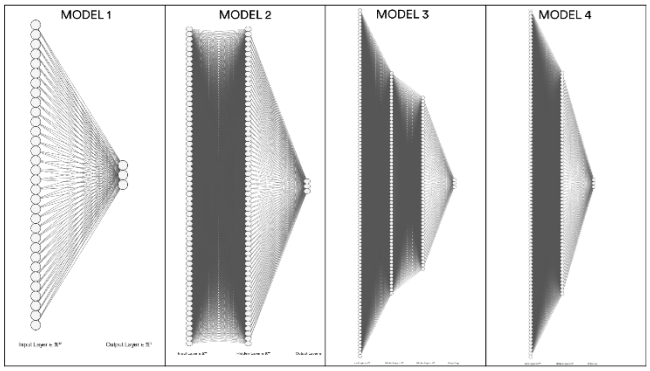

Figure 6 - Multi-Label Multi-Class Models Architectures

Training the model: The models described above are fitted to the training set, with 100 epochs for the single-class model and 200 for the multi-class, batch size of 200 for both.

Validation: Accuracy for training and validation are accessed for each iteration, and its mean and standard deviation are aggregated for the model.

Test: There isn't a noticeable decrease of accuracy over successive epochs on any of the models so there is no need to reduce this parameter.

Evaluation: The test stage of the *single-layer* model is assessed by a *sklearn* classification report that measures precision, recall, f1_score and support for positive and negative values, accuracy, and macro and weighted averages. No changes to the parameters were necessary for this model given the results presented in the next chapter, so it was accepted after the first iteration. As for the multi-label models some overfitting was noticed on model 2 which is the one with best results. Therefore, dropout and L2 regularizations were tested with default parameters. The dropout option ended up being accepted.

A *github* repository was created to host this project code, datasets and report[2].

IV. Results

The single-label multi-class model delivered a mean accuracy of 99,22% with a standard deviation of 0,45
Table 1 details these results.

Table 1 - Training and Validation Accuracy for Single-Label Multi-Class Model

	mean	std deviation
accuracy	0.992159	0.004536
execution time	26.652824	1.886111
training	0.998617	0.002880

On the train and validation stage, the model shows some over-fitting above 60 epochs although maintaining a validation accuracy over 98%.
This trend is represented in Figure 7.

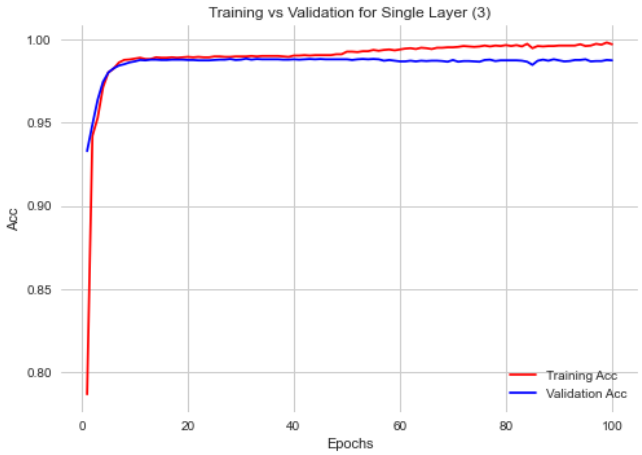

Figure 7 - Graphical Representation of Training and Validation Accuracy over Epochs for the Single-Layer Model

Finally, the model is accessed by a classification report where accuracy records the 85,30% accuracy recorded on table 2.

Table 2 - Single-Label, Multi-Class Model Evaluation

	precision	recall	f1-score	support
0	0.988815	0.986364	0.987588	2420.000000
1	0.986064	0.988569	0.987315	2362.000000
accuracy	0.987453	0.987453	0.987453	0.987453
macro avg	0.987440	0.987466	0.987451	4782.000000
weighted avg	0.987456	0.987453	0.987453	4782.000000

As for the multi-label approach where four sets of parametrizations are tested and consequently four sets of results will be analyzed and compared.

For Model 1, table 3 shows an accuracy of 86,08% and a standard deviation of 1,47%.

[2] https://github.com/maikelps/KOI_ExoplanetClassification

Table 3 - Training and Validation Accuracy for Multi-Label Multi-Class Model 1

	mean	std deviation
accuracy	0.860821	0.014763
execution time	13.150347	0.346272
training	0.874075	0.002643

Figure 8 depicts a slightly growing overfit and a validation accuracy of 86%.

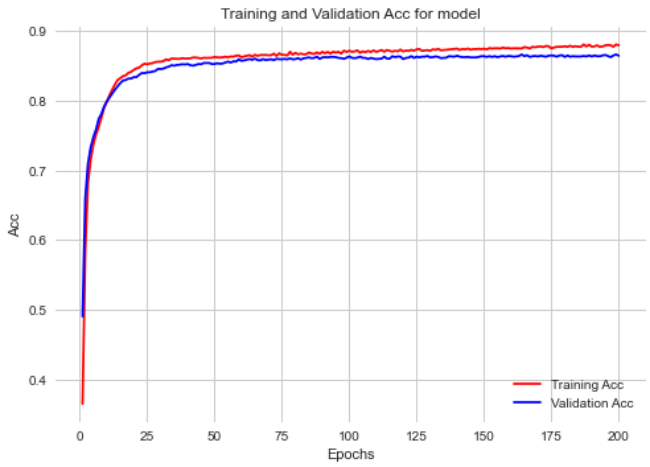

Figure 8 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 1

For Model 2, table 4 shows a validation accuracy of 86,20% and a standard deviation of 1,06%.

Table 4 - Training and Validation Accuracy for Multi-Label Multi-Class Model 2

	mean	std deviation
accuracy	0.862075	0.010696
execution time	13.489426	0.772701
training	0.944857	0.005477

Figure 9 depicts an obvious case of overfit and a validation accuracy of 86%.

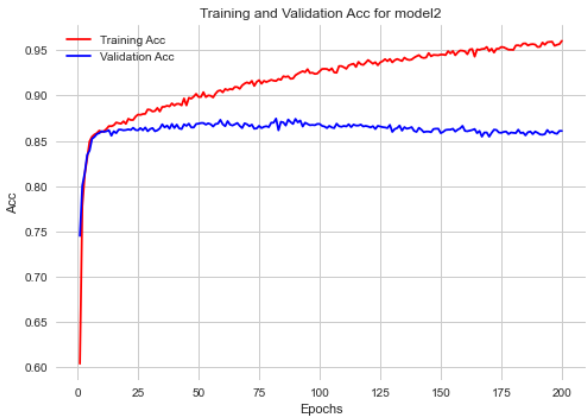

Figure 9 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 2

Model 3 exhibits a validation accuracy of 84,71% and a standard deviation of 0,63% as exhibited in table 5.

Table 5 - Training and Validation Accuracy for Multi-Label Multi-Class Model 3

	mean	std deviation
accuracy	0.847120	0.006396
execution time	17.971869	2.416356
training	0.976251	0.006789

Figure 10 shows that Model 3 behaves like Model 1, having an inferior accuracy.

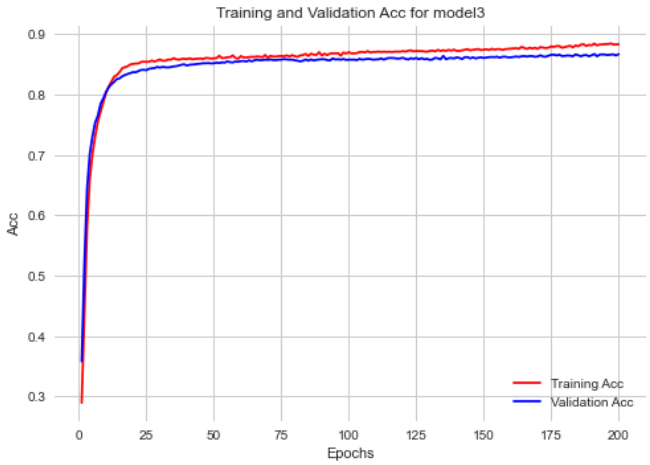

Figure 10 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 3

Model 4 exhibits a validation accuracy of 85,62% and a standard deviation of 0,72% as exhibited in table 6.

Table 6 - Training and Validation Accuracy for Multi-Label Multi-Class Model 4

	mean	std deviation
accuracy	0.856218	0.007244
execution time	13.830466	0.629068
training	0.958486	0.005577

Model 4 show a similar behavior to Models 1 and 3 but having an inferior validation accuracy compared to Model 2.

This behavior is represented on Figure 11.

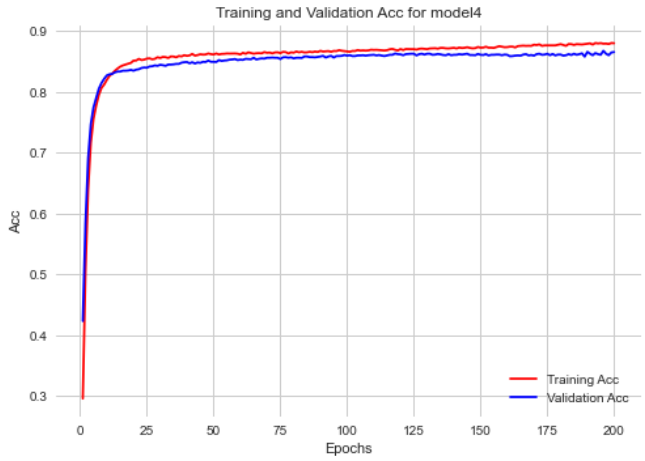

Figure 11 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 4

The regularization of Model 2 by dropout results in a validation accuracy of 88% with a standard deviation of 0,93% as shown in table 7.

Table 7 - Training and Validation Accuracy for Multi-Label Multi-Class Model 2 Regularized by Dropout

	mean	std deviation
accuracy	0.880476	0.009276
execution time	15.393595	1.724357
training	0.892642	0.004415

Figure 12 shows the effect of regularization on the previously obvious overfit.

Figure 12 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 2 Regularized by Dropout

The same regularization by L2 results in an accuracy of 85,51% with a standard deviation of 0,82% as shown in table 8.

Table 8 - Training and Validation Accuracy for Multi-Label Multi-Class Model 2 Regularized by L2

	mean	std deviation
accuracy	0.855171	0.008234
execution time	15.207894	0.802020
training	0.863862	0.006462

Figure 13 shows this regularization, in this model doesn't produce the same results as dropout besides resulting in a lower accuracy.

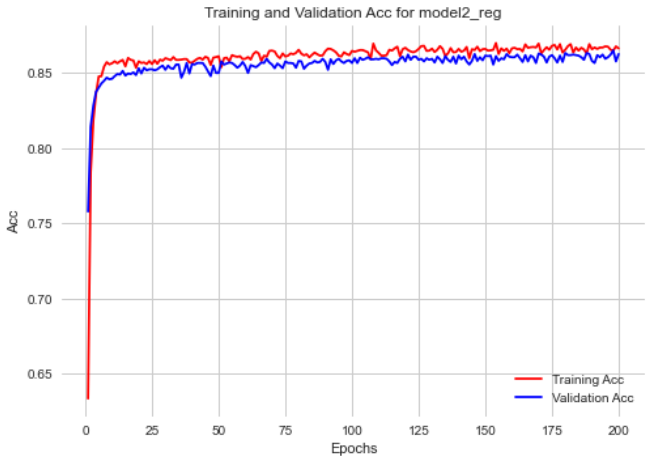

Figure 13 - Graphical Representation of Training and Validation Accuracy over Epochs for the Multi-Layer Multi-Class Model 2 Regularized by L2

Table 9 summarizes the results of k-fold cross validation of the multi-label, multi-class models. This results in different classifications than the ones shown in tables 3 to 8 where the accuracy was measured with the same training and validation sets over each epoch. K-fold cross validation repeats the same process resorting to randomly selected new training and validation sets which is suitable for datasets with fewer records, as the one used in this process.

Table 9 - Comparative Analysis of the Multi-Label Multi-Class Models

	seconds		trainingacc		valacc	
model	**std**	**mean**	**std**	**mean**	**std**	**mean**
model0	0.365003	13.150347	0.002786	0.874075	0.015561	0.860821
model2	0.814499	13.489426	0.005773	0.944857	0.011275	0.862075
model2_dropout	1.817632	15.393595	0.004654	0.892642	0.009778	0.880476
model2_reg	0.845404	15.207894	0.008680	0.855171	0.006811	0.863862
model3	2.547063	17.971869	0.007157	0.976251	0.006742	0.847120
model4	0.663096	13.830466	0.005879	0.958486	0.007636	0.856218

Figure 14 allows comparing results effectiveness-wise (accuracy) and efficiency-wise (processing time).

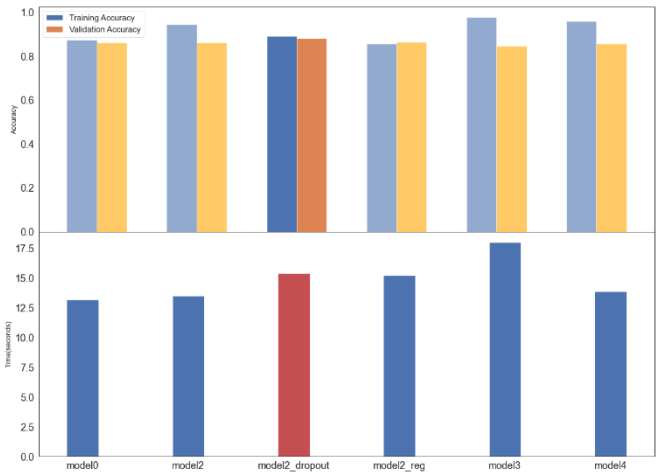

Figure 14 - Effectiveness and Efficiency Criteria

V. Conclusions

The single-label multi-class model turning out 99% of validation accuracy seemed too much obvious fit, and the authors were tempted to accept it on the first attempt finding little reason to research further if not for the doubt raised by the common expression "Too good to be true".

Repeating the process with multi-label multi-class models with different parameters offered the chance to compare results between these results and the ones obtained with the single-label multi-class approach. These models' results were consistent between each other with small differences regarding validation accuracy, but still effectiveness-wise model 2 had the better result.

Checking the 2nd criterion, efficiency-wise, model 2 wasn't the worst, and there was no evident reason to disregard it having the best validation accuracy. As for regularization methods, between dropout and L2, the first proved to be the most effective dealing with the problem being studied, assuring less overfit and better validation accuracy, even if compromising efficiency while requiring some extra processing time.

For the above stated reasons, the authors opt for the model 2 regularized by dropout to classify future observations of potential exoplanets with knowledge learnt from Kepler mission's data, answering so the investigation question.

The major limitation the authors faced was the lack of a scientifically based reason to dismiss the results obtained by the single-label multi-class model.

Future work can include further testing of the rejected model with raw data from Kepler to confidently reject or accept the initial model.

VI. Bibliography

Chen, R., 2018. Kepler By the Numbers – Mission Statistics [WWW Document]. NASA. URL http://www.nasa.gov/kepler/missionstatistics (accessed 3.18.21).

ESA, 2021. ESA Science & Technology - Exoplanet mission timeline [WWW Document]. URL https://sci.esa.int/web/exoplanets/-/60649-exoplanet-mission-timeline (accessed 3.18.21).

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning, Adaptive computation, and machine learning. The MIT Press, Cambridge, Massachusetts.

Gulliford, A., 2018. Art Rocks in Archaic Canyon. Humanities 39.

Hamena, S., Meshoul, S., 2018. Multi-class Classification of Gene Expression Data Using Deep Learning for Cancer Prediction. International Journal of Machine Learning and Computing 8, 6.

Iqbal, I., Younus, M., Walayat, K., Kakar, M.U., Ma, J., 2021. Automated multi-class classification of skin lesions through deep convolutional neural network with dermoscopic images. Computerized Medical Imaging and Graphics 88, 101843. https://doi.org/10.1016/j.compmedimag.2020.101843

Kepler Objects of Interest [WWW Document], 2018. URL https://exoplanetarchive.ipac.caltech.edu/cgi-bin/TblView/nph-tblView?app=ExoTbls&config=cumulative (accessed 3.27.21).

Liu, L., Wang, R., Xie, C., Yang, P., Wang, F., Sudirman, S., Liu, W., 2019. PestNet: an end-to-end deep learning approach for large-scale multi-class pest detection and classification. IEEE Access 7, 45301–45312.

NASA, 2021. Exoplanet Exploration: Planets Beyond our Solar System [WWW Document]. Exoplanet Exploration: Planets Beyond our Solar System. URL https://exoplanets.nasa.gov/ (accessed 3.16.21).

Nawaz, M., A., A., Hassan, T., 2018. Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network. ijacsa 9. https://doi.org/10.14569/IJACSA.2018.090645

Ramzan, F., Khan, M.U.G., Rehmat, A., Iqbal, S., Saba, T., Rehman, A., Mehmood, Z., 2019. A Deep Learning Approach for Automated Diagnosis and Multi-Class Classification of Alzheimer's Disease Stages Using Resting-State fMRI and Residual Neural Networks. J Med Syst 44, 37. https://doi.org/10.1007/s10916-019-1475-2

Seifer, M., 2016. Wizard: The Life and Times of Nikola Tesla: Biography of a Genius, Reprint edition. ed. Citadel.

Tsoumakas, G., Katakis, I., 2007. Multi-Label Classification: An Overview. IJDWM 3, 1–13. https://doi.org/10.4018/jdwm.2007070101

Wever, M., Mohr, F., Hüllermeier, E., 2018. Automated Multi-Label Classification based on ML-Plan. arXiv:1811.04060 [cs, stat].