Universidad Complutense de Madrid
Capstone Project
Master Big Data & Data Science
09/22/2022

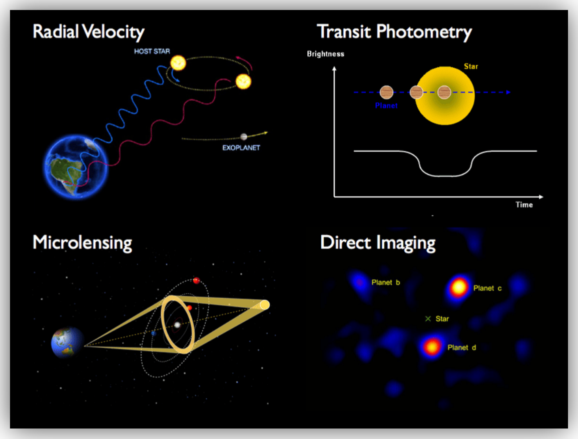
Exoplanet Identification using Deep Learning

Christopher A. Rodriguez Principe

Index

Mission Overview

The National Aeronautics and Space Administration (NASA) has designated the Kepler Satellite to survey our region of the Milky Way galaxy to discover hundreds of Earth-size and smaller planets and determine the stars in our galaxy that might have such planets. Currently, Kepler has confirmed 3,381 planetary systems and 5,084 confirmed exoplanets using various exoplanet detection techniques displayed below.



Project Overview

The Exoplanet Exploration Program needs a deep learning model to predict and identify confirmed exoplanets in distant star systems. The stakeholders of this project are all exoplanetary scientist working at NASA. I have been contracted as a Data Scientist to create this model given the Kepler Object of Interest (KOI) Dataset.

Kepler Object of Interest (KOI) Dataset

All object observations gathered by Kepler are registered under the Kepler Object of Interest (KOI) Dataset. The data is publicly accessible through the *Exoplanet Archive Application*

Programming Interface (API), under cumulative.csv. For the purpose of this capstone project, the data was downloaded through Kaggle. Here Is the following link to download the data : https://www.kaggle.com/datasets/nasa/kepler-exoplanet-search-results.

The file was downloaded as a CSV file under the name 'cumulitive.csv '. It has a size of 3.7 Megabytes. The Data frame is composed of 9,564 rows and 50 columns. The data itself is comprised mostly of *numerical* values, yet also containing some *string* values. There are 49 independent variables which will be processed and implemented into a model and 1 dependent variable which is our variable of interest (our prediction variable).

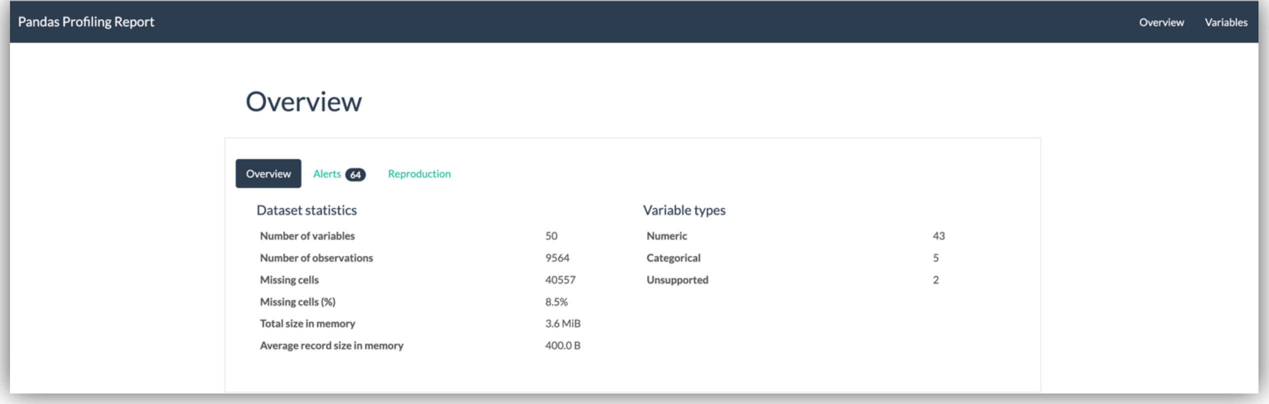
Our variable of interest is called **koi_disposition.** Current values are CANDIDATE, FALSE POSITIVE or CONFIRMED. For the purpose of this project we are only focusing on CONFIRMED and FALSE POSITIVE. CANDIDATE designations may change over time as the evaluation of KOIs proceeds to deeper levels of analysis using Kepler time-series pixel and light curve data, or follow-up observations.

It is due to the lack of exoplanet scientific guidance that I have chosen to proceed with a focus on CONFIRMED and FALSE POSITIVE designations. KOI's Labeled with CONFIRMED mean that it has rightfully identified an exoplanet. Meanwhile, FALSE POSITIVE designations are objects that have not met the qualifications of an exoplanet.

Exploratory Data Analysis (EDA)

Most Deep Learning models need clean data as input. Cleaning data requires two necessary steps. First we must implement EDA. The Analysis refers to the critical process of performing initial investigations on data to discover patterns,to spot anomalies and to check assumptions with the help of summary statistics and graphical representations.

I implemented a Pandas Profiling Report which automatically generates a standardized univariate and multivariate HTML report for data understanding. Pandas Profiling Report's Overview gives us quick dataset statistics to understand the composure of our data.



Overview

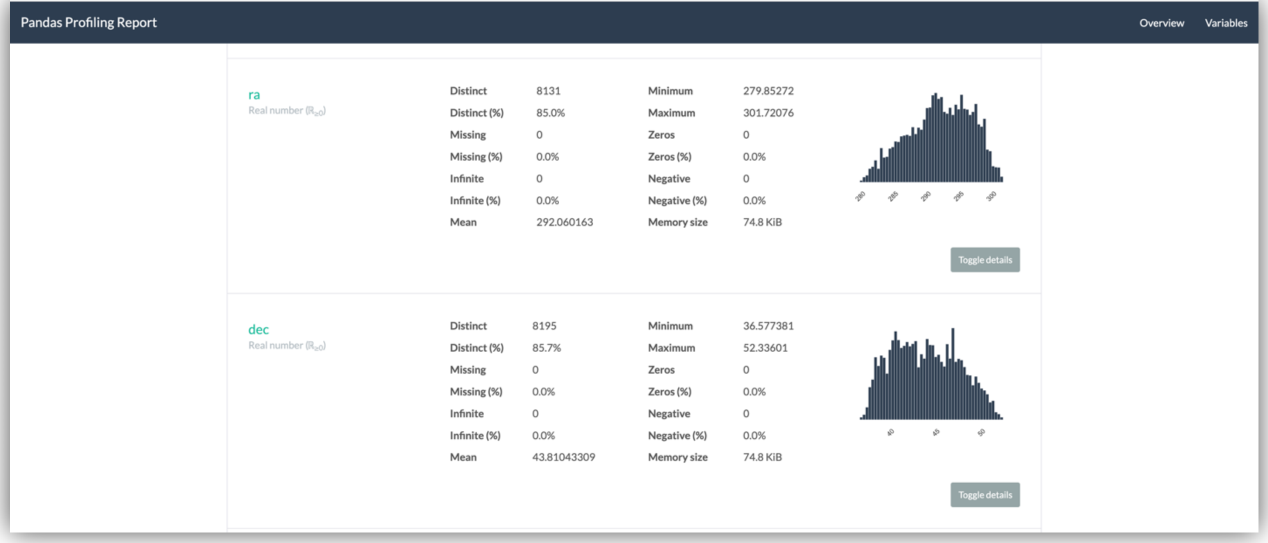
Dataset statistics

Number of variables	50
Number of observations	9564
Missing cells	40557
Missing cells (%)	8.5%
Total size in memory	3.6 MiB
Average record size in memory	400.0 B

Variable types

Numeric	43
Categorical	5
Unsupported	2

It can also give us an understanding of the distribution of each variable, percentage of missing values and insight into identifying atypical values.



ra
Real number ($\mathbb{R}_{\geq 0}$)

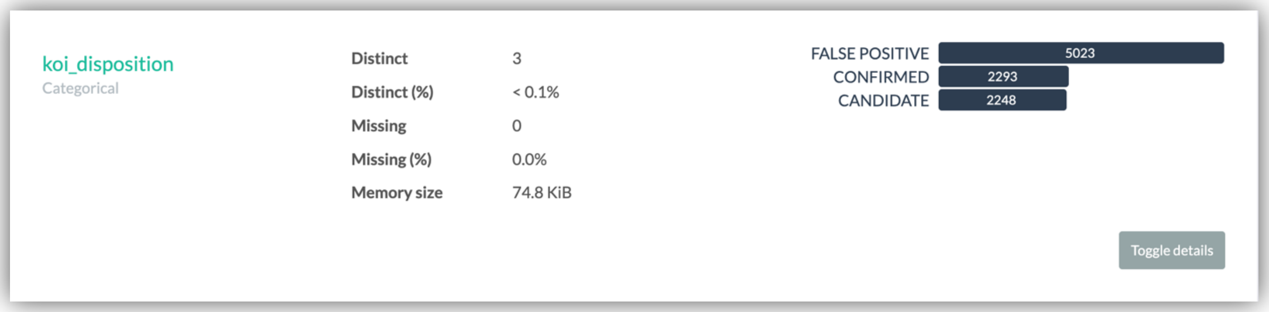
Distinct	8131	Minimum	279.85272
Distinct (%)	85.0%	Maximum	301.72076
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	292.060163	Memory size	74.8 KiB

dec
Real number ($\mathbb{R}_{\geq 0}$)

Distinct	8195	Minimum	36.577381
Distinct (%)	85.7%	Maximum	52.33601
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	43.81043309	Memory size	74.8 KiB

As I mentioned before koi_disposition is our variable of interest (dependent variable). Since we are only focusing on FALSE POSITIVE and CONFIRMED, we can observe the distribution of these.

There are double the amount of FALSE POSITIVE’s than CONFIRMED.


koi_disposition
Categorical

Distinct	3
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	74.8 KiB

Data Wrangling

The second phase into cleaning our data is Data wrangling. It refers to a variety of processes designed to transform raw data into more readily used formats. Given our analysis of the data done in EDA, I proceeded eliminating variables with unique values, Null values and columns that served as identification. The primary reason we eliminate these variables is because they do not add any additional information.

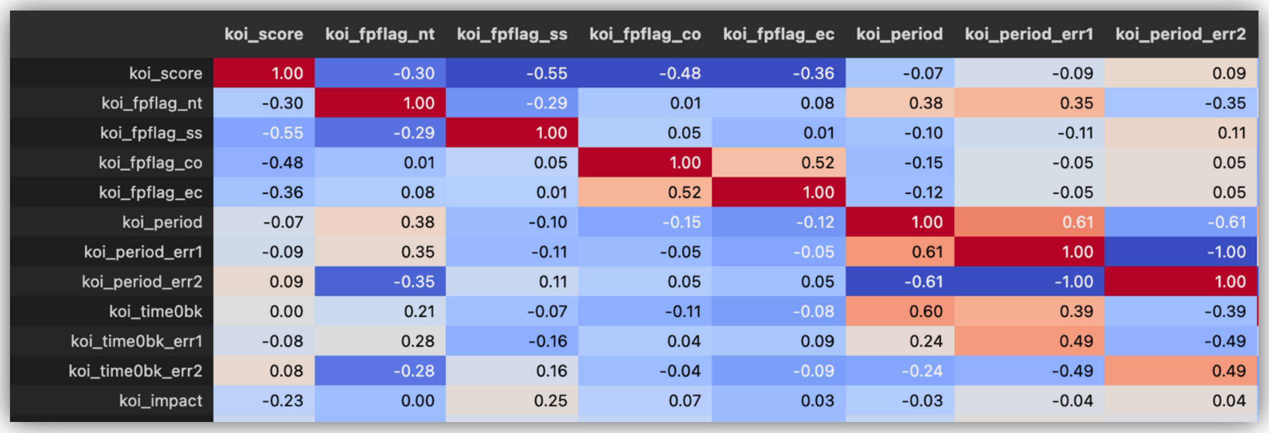
It is very important to clarify that some Null values are many times replaced by the Min, Mean or Max of the column. This decision is taken by subject matter experts. I do not have scientific expertise in exoplanets. I am oblivious to implications of many of the variables therefore I chose to eliminate null values from the data frame. These transformations now leaves us with a cleaner data frame of 6,031 rows and 43 variables.

Feature Engeneering

Feature engineering is the process of selecting, manipulating, and transforming raw data into features that can be used in supervised learning. I chose to implement One-Hot Encoding to create new variables. Many machine learning algorithms are unable to process categorical variables. Therefore, it is important to encode the data into a suitable form so you can preprocess these variables. It converts categorical variables into binary. I created two new variables based on koi_pdisposition variable. Now that we created the variables and our data is numerical we can proceed to the next stage.

Eliminating Correlated variables

It is recommended to avoid having correlated features in your dataset. Indeed, a group of highly correlated features will not bring additional information, but will increase the complexity of the algorithm, thus increasing the risk of errors. I have therefore created a correlation matrix (52x52).

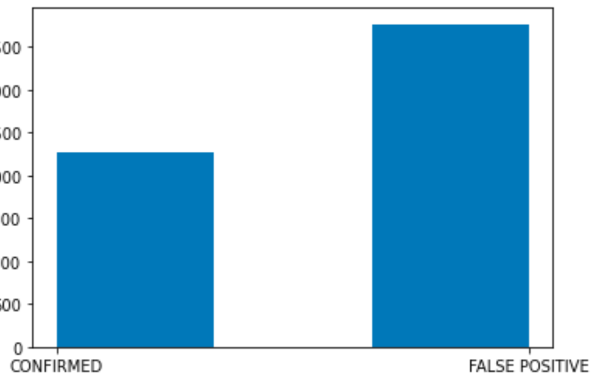


	koi_score	koi_fpflag_nt	koi_fpflag_ss	koi_fpflag_co	koi_fpflag_ec	koi_period	koi_period_err1	koi_period_err2
koi_score	1.00	-0.30	-0.55	-0.48	-0.36	-0.07	-0.09	0.09
koi_fpflag_nt	-0.30	1.00	-0.29	0.01	0.08	0.38	0.35	-0.35
koi_fpflag_ss	-0.55	-0.29	1.00	0.05	0.01	-0.10	-0.11	0.11
koi_fpflag_co	-0.48	0.01	0.05	1.00	0.52	-0.15	-0.05	0.05
koi_fpflag_ec	-0.36	0.08	0.01	0.52	1.00	-0.12	-0.05	0.05
koi_period	-0.07	0.38	-0.10	-0.15	-0.12	1.00	0.61	-0.61
koi_period_err1	-0.09	0.35	-0.11	-0.05	-0.05	0.61	1.00	-1.00
koi_period_err2	0.09	-0.35	0.11	0.05	0.05	-0.61	-1.00	1.00
koi_time0bk	0.00	0.21	-0.07	-0.11	-0.08	0.60	0.39	-0.39
koi_time0bk_err1	-0.08	0.28	-0.16	0.04	0.09	0.24	0.49	-0.49
koi_time0bk_err2	0.08	-0.28	0.16	-0.04	-0.09	-0.24	-0.49	0.49
koi_impact	-0.23	0.00	0.25	0.07	0.03	-0.03	-0.04	0.04

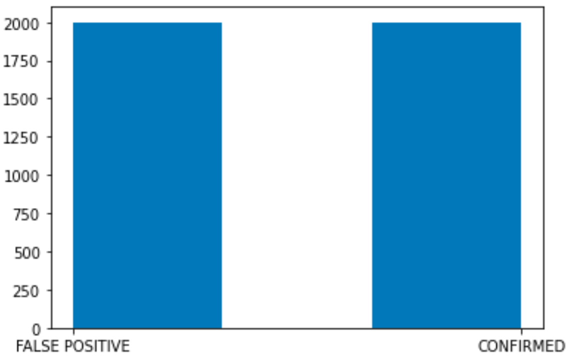
Based on the following matrix I eliminated variables with correlation stronger than ±70. This results with a subtraction of 12 correlated variables from our data frame.

Data Binning

Data binning or data bucketing, is a data pre-processing technique used to reduce the effects of minor observation errors. It is necessary procedure so that models can train on low represented values within a dataset. It is also needed to uniformaly distribute our data so that it can be implemented on unsupervised machine learning models that will be discussed further.



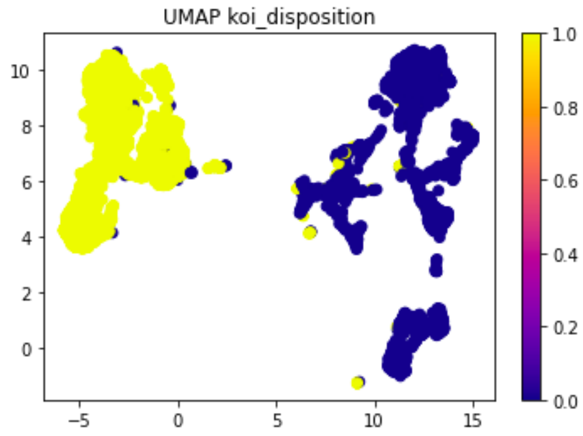
Given the distribution of our variable of interest we need to level the plain field and create a sampled data frame that has equal amount of FALSE POSITIVE and CONFIRMED koi_disposition. I have created a sub-sampled data frame selecting 2,000 rows of FALSE POSITIVE and 2,000 CONFIRMED. Our new uniformly distributed sub-set of dataset has 4,000 rows. This is our final Data Frame that we will start to implement our models.



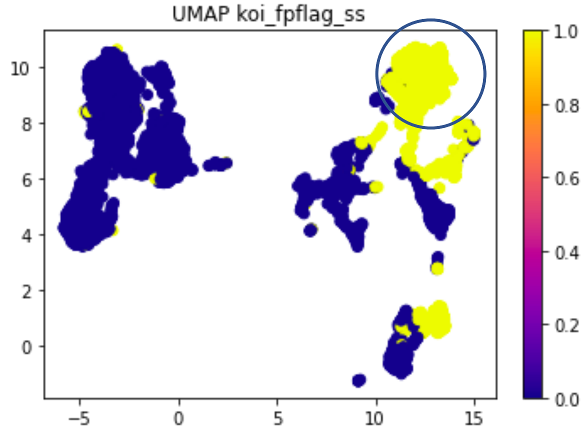
UMAP (Uniform Manifold Approximation and Projection)

UMAP is a dimension reduction technique that can be used for visualisation, but also for general non-linear dimension reduction. Now that we have created a new data frame which has uniformaly distributed data we can apply it to UMAP. It is also a Unsupervised Machine Learning technique which creates Clusters of our data, it can be vary useful for Scientist since it allows to visualize the clusters. UMAP is Clustering all of our data. In UMAP clusters closer to each other mean that the data points are related to each other.

UMAP only accepts numerical values. Meanwhile, our variable of interest is Categorical. Therefore, I label encoded the categorical values into binary. CONFIRMED values are 1. FALSE POSITIVE values are 0. Now we can look at Clusters of the data.

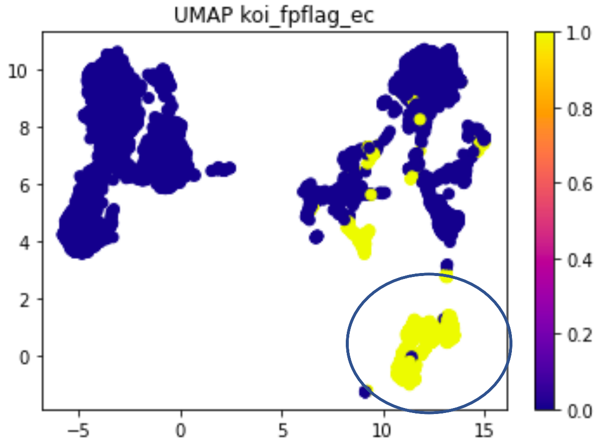


This UMAP is Clustering all of our data and coloring it by koi_disposition. Values equal to 1 are confirmed exoplanets and values of 0 are false positive. Immediately we notice that the UMAP algorithm is easily detecting the differences between CONFIRMED and FALSE POSITIVE values. We now that because we can see that colors are cluster apart in the UMAP.

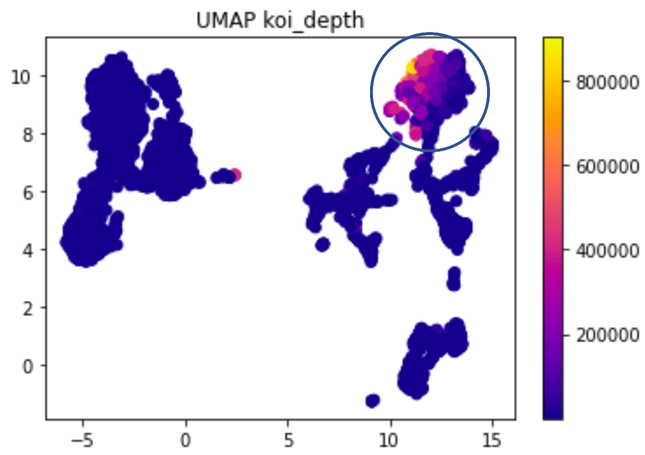


This next UMAP is Clustering on koi_fpflag_ss. This is the Solar Eclipse Flag variable. It gives the value of 1 when the observation indicates that the transit-like event is most likely caused by an eclipsing binary star. I have indicated with a circle over a Cluster that over-represents Solar Eclipse Flags. Here, all observations where koi_fpflag_ss identifies a binary eclipse have a koi_disposition of 0 (FALSE POSITIVE).

Exoplanet are detected by measuring a periodic decrease in the flux received from the host star, as a consequence of the exoplanet transiting in front of the host star. When another bright star passes in front of our observing star the flux of light changes and a flag is raised on our koi_fpflag_ss variable thinking it has detected an exoplanet when in fact it is a binary star system.



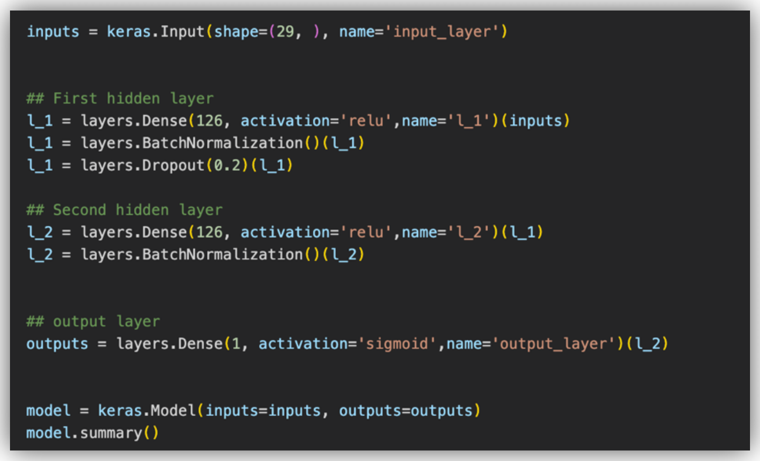
This next UMAP is Clustering on koi_fpflag_ec. The KOI shares the same period and epoch as another object and is judged to be the result of flux contamination in the aperture or electronic crosstalk. When koi_fpflag_ec equals True (1) , our variable of interest (koi_disposition) results with a FALSE POSITIVE.



This last UMAP colored by koi_depth or Transit Depth. The transit depth is the ratio of the surface area of the star's disk blocked out by the planet's disk. So the transit depth is the square of the planet radius divided by the star's radius. All of our Confirmed exoplanets have a Transit Depth lower than 200,00 ppm. Meanwhile some False Positive exoplanets exceed a transit depth of 400,000 ppm.

Deep Learning Model

Finally, I have created Deep Learning Models using Neural Networks. Although I have created three different models, I will be discussing the model that gave best results. The Code below is our best scoring model for predicting and identifying exoplanets. I'll go into detail on how I built this model and why I chose the parameters for the model.



```
inputs = keras.Input(shape=(29, ), name='input_layer')


## First hidden layer
l_1 = layers.Dense(126, activation='relu',name='l_1')(inputs)
l_1 = layers.BatchNormalization()(l_1)
l_1 = layers.Dropout(0.2)(l_1)

## Second hidden layer
l_2 = layers.Dense(126, activation='relu',name='l_2')(l_1)
l_2 = layers.BatchNormalization()(l_2)


## output layer
outputs = layers.Dense(1, activation='sigmoid',name='output_layer')(l_2)


model = keras.Model(inputs=inputs, outputs=outputs)
model.summary()
```

I created a simple neural network with 2 hidden layers with 126 neurons per each layer. To prevent overfitting of the data I implemented a few techniques, such as implementing Batch, normalization on both hidden layers, implementing Dropout Layer to help train our data and reducing batch size of the model. All of these techniques are important to prevent overfitting our data in our model and help us achieve a better accuracy in our model.

Furthermore, applying activation functions for each layer can determine the score accuracy of the model. The first two layers I applied 'relu' activation function because all input values fall under [0, inf). Our output layer is in charge of returning our variable of interest which is 1 if CONFIRMED and 0 if FALSE POSITIVE. Since our output result is binary (1 or 0) the best activation function is Sigmoid, which returns a binary classification.

Results

Our model returns a Test Accuracy of ≈ 0.98 and a Test Loss of ≈ 0.012. The lower the loss and the higher the Accuracy**,** the better our model. In conclusion, our model has great predictive power to identify CONFIRMED exoplanets and FALSE POSITIVE objects.

Future Work and Considerations

Given our current model, It has great predictive power to identify CONFIRMED and FALSE POSITIVE. Future work should include the following suggestions:

- Add CANDIDATE to our variable of interest and model if it can distinguish it from CONFIRMED.
  - CONFIRMED and CANDIDATE values are very similar and our model still does not know how to distinguish between both.
- Identify if new variables can be added to the model.
- Evaluate if Nan values can be replaced by the Mean of Median of the Variable.
- Implement further Normalization Techniques into the model such as:
  - Reduce amount of neurons per layer and evaluate if results maintain.
  - Add a second Dropout layer.

- Revise Data Binning Technique when implementing a third output value (CANDIDATE), making sure the data is uniformly distributed.

All of these suggesting should be further investigated for implementation and discussed with subject matter experts, this would increase the predictive power of the model.

- Consider Revising Prediction values. Values of prediction come out as float instead of integer.