UNIVERSITÀ DI PISA

Artificial Intelligence and Data Engineering

Data Mining and Machine Learning Project

Machine Learning Pipeline for Exoplanet Classification and Clustering

Academic Year 2021/22

Luana Bussu, Luca Caprioli

Sommario

Introduction

Planets orbiting stars outside our solar systems are called extra-solar planets or exoplanets. Planet identification has typically been a task performed exclusively by teams of astronomers and astrophysicists, using several methods and tools accessible only to those with years of academic education and training, being the analysis of periodicities in star light-curves the most successful so far. However, NASA's Exoplanet Exploration program has introduced modern satellites capable of capturing many objects of interest and this has opened the task to find use of automatized methods that can reproduce the analysis performed by astronomers to decide if the data supports the existence of an exoplanet.
In this study, several classification models and datasets are utilized to assign a probability of an observation being an exoplanet. A Random Forest Classifier was selected as the optimal machine learning model to classify K2 and Kepler objects of interest. The Random Forest Classifier obtained a cross-validated accuracy score of 93%.
Starting from the confirmed exoplanets on the original dataset with full planetary measurements, we add the confirmed exoplanets classified by the trained classifiers and we perform clustering to discover different classes of exoplanets.
The Machine Learning Pipeline has been developed using Python and can be found at the link:
https://github.com/lcaprioli17/MLPECC

Dataset

The dataset used in this application is made up of different datasets related to two NASA missions [1]: Kepler and K2.
The Kepler and K2 datasets are more similar because K2 is the continuation of the Kepler mission, they contain approximately 9500 rows and 140 columns for the first and 2600 rows and 290 columns for the second, both containing a high percentage of Nan values, correlated attributes and descriptive attributes [2].
Considering that most of the discovered exoplanets have been detected through the transit method, 9241 FITS files were collected from the MAST (Mikulski Archive for Space Telescopes) [3] through scraping, thanks to the Lightkurve [4] and Astroquery [5] libraries for python, where some of them contain more than one light curve because different objects were detected for the same star. Moreover, each file contains the error associated to each measure, the time (in Julian date) when the measure was made, the raw light curve and the filtered light curve.

Pre-processing

The pre-processing was carried out starting from the two distinct datasets in order to have clean and dimension reduced data for each of them.
After data cleansing, data integration was performed by merging the different files. Given the not negligible difference in numerosity and the structural similarity of the two datasets, Nan value filling was performed on the merged dataset to preserve important attributes that, in case of divided pre-processing, could be lost.

Removing Irrelevant attributes

In each dataset, the first operation of the pre-processing is to perform a manual attribute selection removing all the features that are not domain specific as the unique identifiers, planet names, the discovery date and so on.
As second step were removed those features having no variance (attributes that had a unique value), duplicate rows on the attribute *'name'* and all the features which name like '*error'* or *'limit'* because they represent the errors on the measurement of some attribute (es. orbital time $\pm$ error) because they are no relevant for the classification.
As last step of this phase labels of classes was changed from string to numeric for sake of simplicity: we changed labels 'FALSE POSITIVE', 'CONFIRMED', 'CANDIDATE' in '0', '1', '2'.

Feature extraction

After the scraping of fits files regarding the light curves we performed the feature extraction from these files. We used extraction techniques specialized on time series, which in this case corresponds to measurements of intensity of the light along time.
We extracted the following features following the study by [Margarita Bugueño et al., 2018]:

- Amplitude, defined as the difference between the maxi- mum value and the minimum value divided by 2.
- Slope, defined as the slope of a linear fit to the light curve.
- Max, the maximum value of the sequence.
- Mean, the average of the sequence.
- Median, the median magnitude of the sequence.
- Median Abs Dev, defined as the median of the difference between every point to the median of the sequence.
- Min, the minimum value of the sequence.
- Q1, the first quartile of the data in the sequence.
- Q2, the second quartile of the data in the sequence.
- Q31, the difference between the third and first quartiles.
- Residual bright faint ratio is the rate between the residual of the fainter intensities over the brighter intensities, with the mean of the sequence as threshold.
- Skew, defined as a measure of the asymmetric of the sequence (third standardized moment).
- Kurtosis, defined as the fourth standardized moment of the sequence.
- Std, standard deviation of the sequence.

Due the few features generated over the sequence that can summarize the light-curve, we decided to integrate these data with the metadata in the two datasets described above.

Univariate analysis

Then univariate analysis was then performed to visualize the distribution of relevant variables.

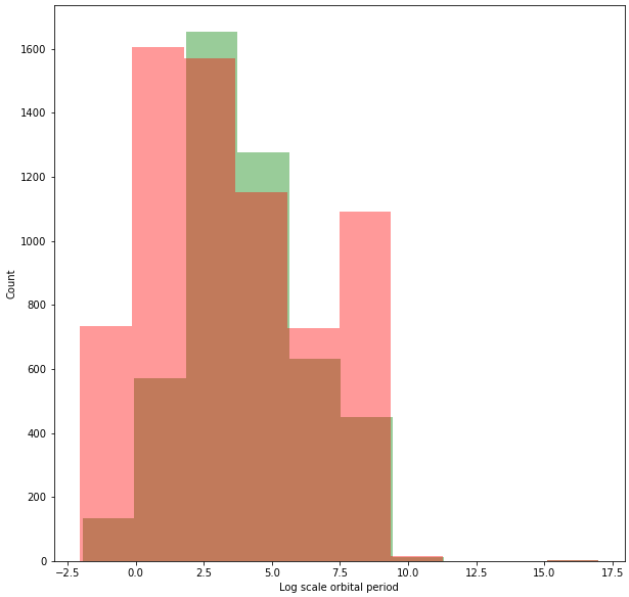

Fig.1: Kepler Log scale orbital period

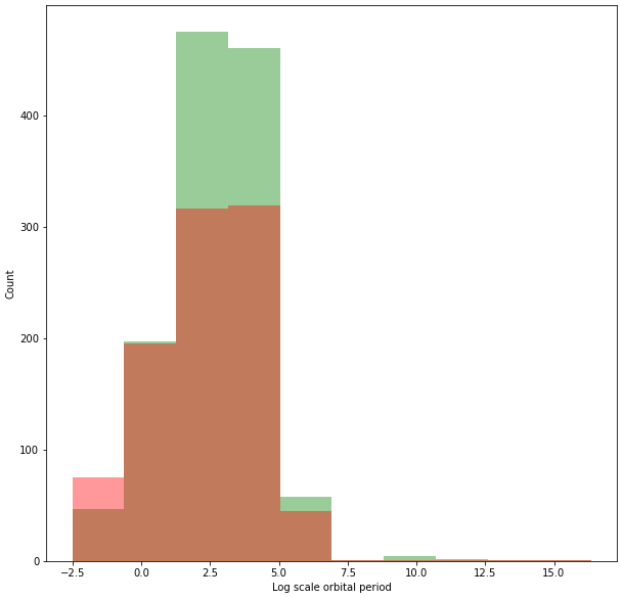

Fig.2: K2 Log scale orbital period

Fig.1 is a histogram that indicates the relationship between an important feature, the orbital period on a logarithmic scale, and the target variable in the Kepler dataset. From it, the confirmed exoplanets follow a Gaussian distribution with respect to the log scale orbital period in the range between -2 and 11. If the value of the log scale orbital period is too high or too low, the candidate is more likely to be a false positive observation.

On the other hand, Fig.2 represents the same variable in the K2 dataset, from which we can see that an orbital period between 2 and 5 maybe be from an exoplanet.

A similar study was conducted for the planet radius attribute, bringing up similar results, albeit with obviously different intervals

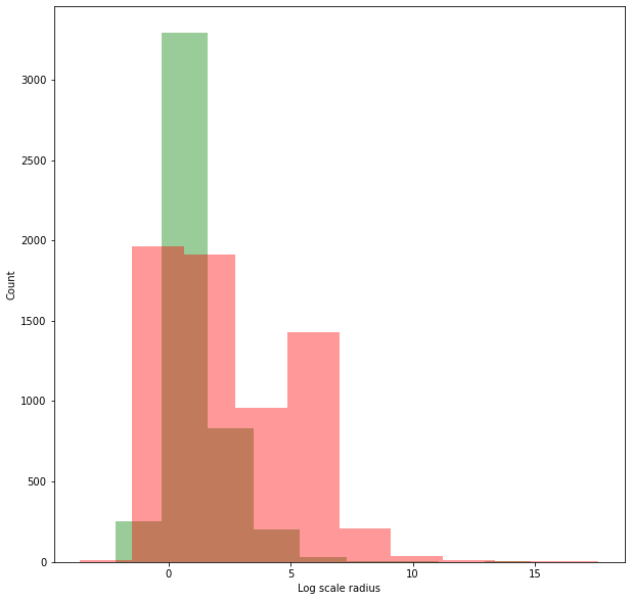

Fig.3: Kepler Log scale planet radius

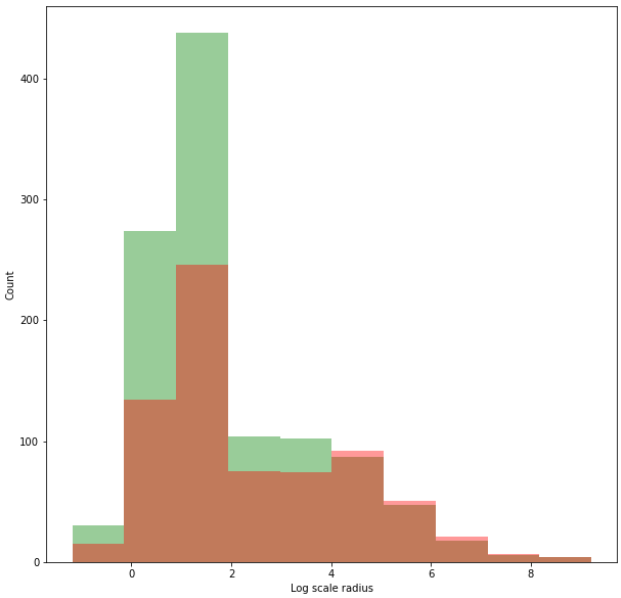

Fig.4: K2 Log scale planet radius

Bivariate analysis

Finally, we used bivariate analysis to investigate the pairwise relationship of different features and observe how they affect the target variable.

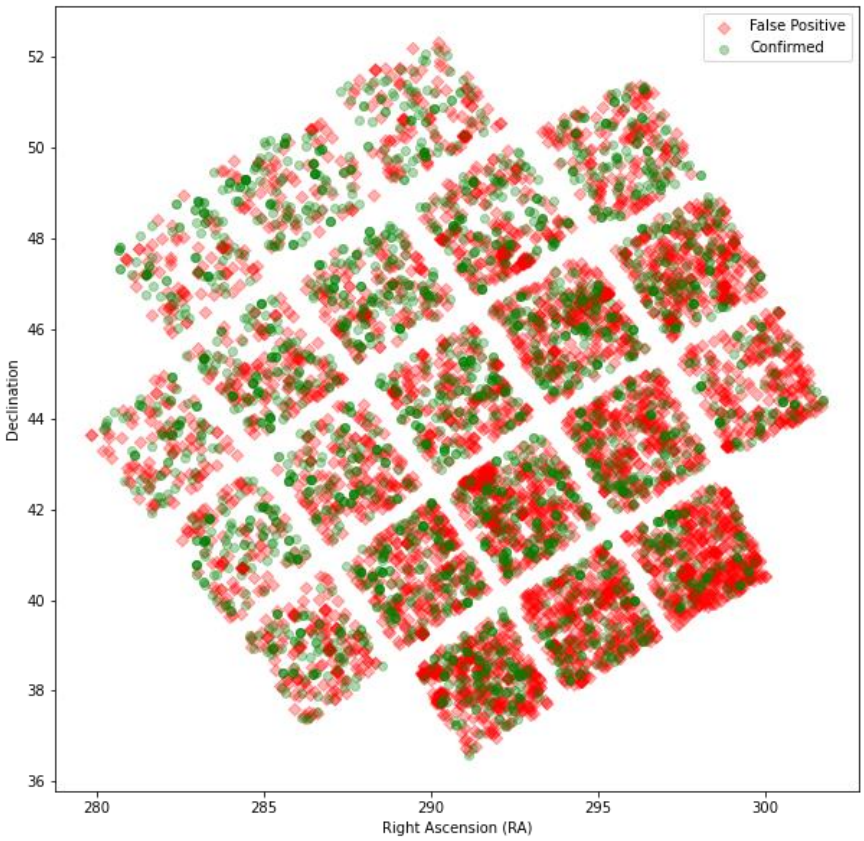

Fig.5: Star Map by Right Ascension (RA) and Declination (Dec)

This figure is a star map of the exoplanet candidates of the Kepler dataset. In this star map, each pair of celestial coordinates is measured by right ascension (RA), the celestial coordinate that represents longitude, and declination (Dec), the celestial coordinate that represents latitude. It demonstrates that there is no strong correlation between the target variable and coordinates.

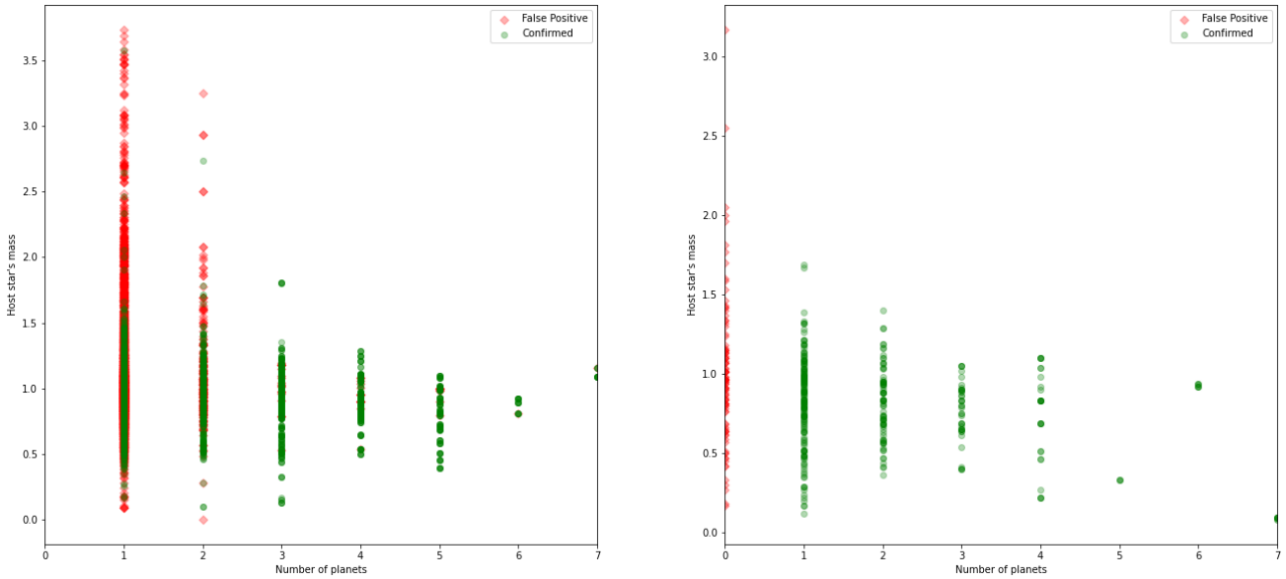

Fig.6, 7: Kepler and K2 Host star mass versus # of planets

Fig: 6,7 are categorical plots of the host star's mass versus the number of planets in the exoplanet candidate's solar system of respectively the Kepler and K2 datasets.
The result shows that false positive samples are most likely to have just one or two planets in their solar system. With a higher number of planets in a candidate's solar system, the probability of it being a real exoplanet is also higher.
It is interesting to note that the higher the star mass, less the probability of exoplanets in its neighbourhood.

Correlation Analysis

Before analysing correlated attributes, we made a manual evaluation based on the Random Forest classifier feature importance analysis conducted by [Yucheng et al., 2022] (Fig.8)

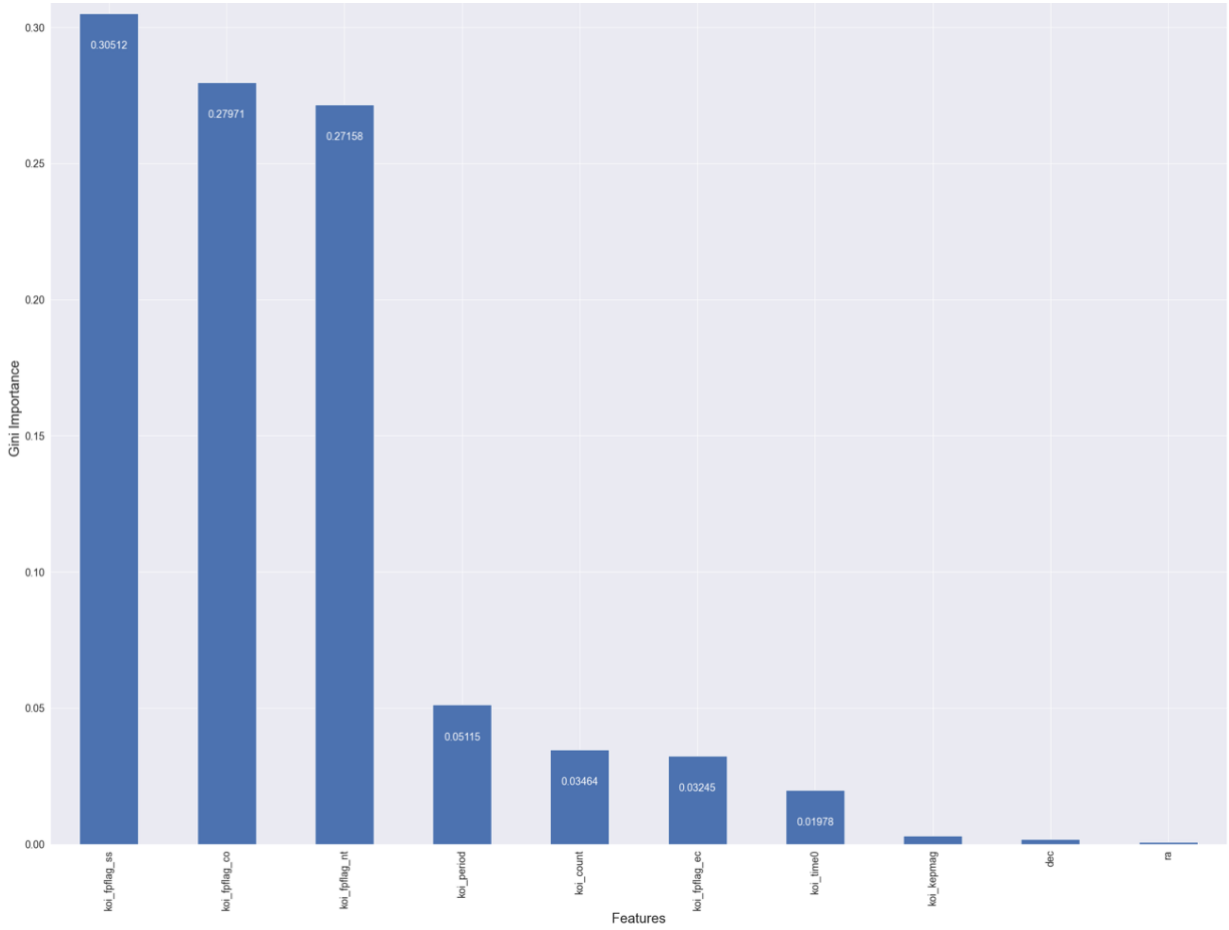

Fig.8: Gini importance score of features in Kepler dataset

Using this study as ground truth we decided to remove the attributes containing flags computed on other attributes, this in order to avoid eventual overfitting and to base our analysis purely on celestial objects measurements.
As a last step of pre-processing on the distinct datasets, a correlation analysis was conducted to identify highly correlated variables: once identified, they were removed to reduce data redundancy for the two datasets. Most of the features removed in this way were same measurements in different units and deviation of observations.

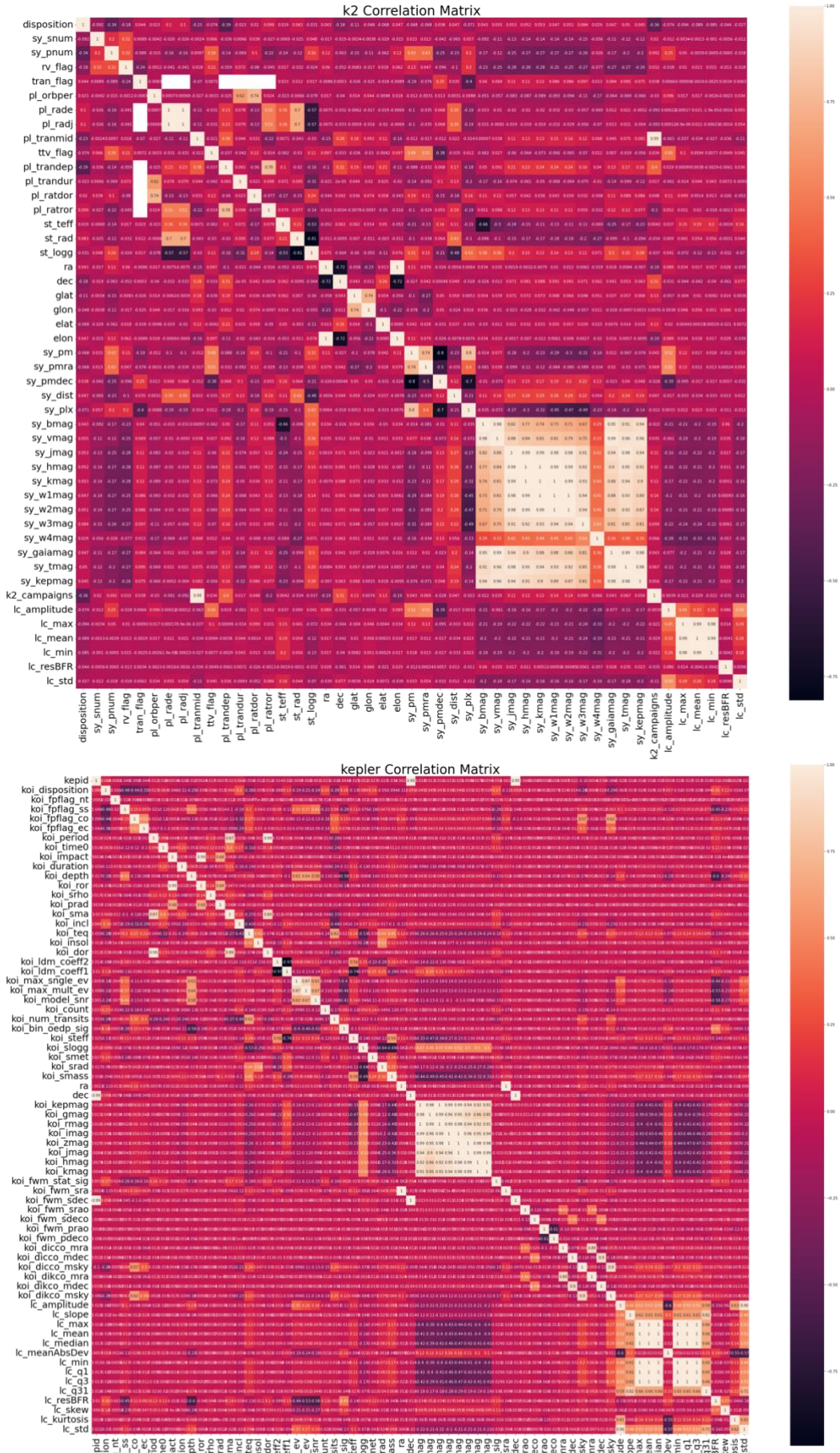

Fig.9: K2 and Kepler correlation matrices

Data Integration

After data cleaning and feature extraction, data integration was performed in order to merge the different data frames in in the idea that K2 and Kepler are basically two part of the same mission and had the many common features.
To perform the integration, we renamed the attributes with same meaning but different names (keeping K2' names), we selected the common columns and removed the others: after merging the resulted dataset is composed by 30 features, starting from the 55 and 43 features respectively for Kepler and K2.
The starting datasets were composed by the 440 confirmed planets and 153 false positives for K2 and 2657 and 4410 for Kepler, so the final dataset has 3105 confirmed samples and 4563 false positives and is, therefore, unbalanced.

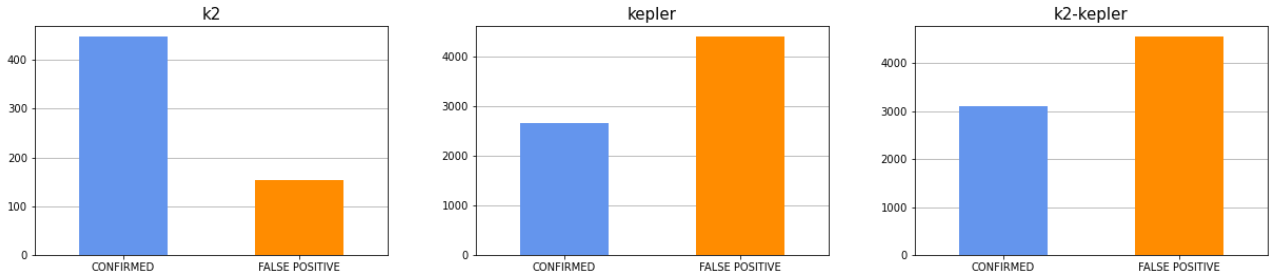

Fig.10: K2, Kepler and merged dataset class frequency

As last step before handling the missing values, rows, and columns with a high percentage of Nan values (higher than 30%) were removed. These percentiles were chosen considering all the attributes remaining after feature selection and integration as fundamentals for the analysis.

Handling missing values

KNN imputation was carried out to ultimately fill the remaining missing values with the mean value of the very observation's K neighbors: different K were tested using 10-fold cross-validation with the highest scoring classifier in the interval [3, sqrt(n)] (with n number of rows) and following the succession:

$$x_n = x_{n-1} * 2 + 1$$

Results are reported in the table below:

K Neighbors	F1-Score	Std
3	0.910	0.059
7	0.909	0.064
15	0.912	0.049
31	0.913	0.057
63	0.903	0.053
85	0.916	0.051

Table 1: Comparison of different K results

Ultimately the K chosen was 85, that is square root of the length of the dataset.

Regarding the FITS files, even though each stored light curve was about 70000 measurements, every point in the series is not generated independently. As the dispersion varies through time, the series is governed by a trend and could have cycles. This fact is important because the Kepler measurements are not recorded uniformly, getting light curves with missing data. On average, the missing data is about 22.98% of the size of the fully sampled light curve. This means that every light curve has approximately about 55000 effective measurements. Considering this analysis, these missing values were processed using linear interpolation following [Margarita Bugueño et al., 2018]:

k2-kepler

	pl_name	disposition	pl_orbper	pl_rade	pl_trandep	pl_trandur	pl_ratdor	pl_ratror	st_teff	st_rad	st_mass	st_logg	dec	sy_kepmag	lc_amplitud	lc_mean	lc_median	lc_q3	lc_q31	lc_resBFR	lc_skew	lc_kurtosis	lc_std
0	BD+20 594	1.0	41.6855	2.578	49	3.902151351	55.8	0.189779186	5766.0	1.08	1.67	4.5	20.5990205	11.04	630.59375	599867.25	105366.2950	375246.0824	270381.8007	1.022753792	-0.52718764	4.855617322	98.26539611
1	EPIC 20117	1.0	6.7987	1.047	3.829943783	3.861794594	15.26363951	34	3648.0	0.28	0.29	5.0	-4.8070039	15.673	1088.940917	3903.893310	107861.354	359996.7342	252695.3058	0.988318943	-0.58507621	4.851499802	45.91563796
2	EPIC 20123	1.0	28.1696	2.2	316	3.36	61.5	54	5558.432432	0.37	0.689189189	4.491081081	-3.3894613	14.904	287.9497070	9583.538085	108489.1898	360113.5912	252196.5715	0.996430392	-0.64978828	4.851499775	45.46023555
3	EPIC 20125	0.0	50.285869	3.47	184	12.62	11.63029710	0.05345	4859.0	0.74	0.956486486	4.328648648	-3.0949095	11.511	3636.640625	324765.9062	133550.0374	421689.5387	288682.3087	1.431677018	-0.77167663	5.703034243	560.5809936
4	EPIC 20132	0.0	2.51881781	45.65516216	215	1.58	15.68389051	0.191326216	6380.0	1.76	0.874864864	4.414054054	-2.0851597	12.146	594.6328125	195738.0312	107861.354	397242.0032	289940.5746	1.121951219	-0.58461070	5.703925116	177.4826202
5	EPIC 20142	1.0	0.72091	1.5	0.360691893	3.996592433	3.1	0.0142	5633.0	0.94	0.769189189	4.4	-0.5512886	14.219	322.4667968	30969.26177	30967.72656	30977.78710	19.28369140	1.187459598	11.34106833	83865128.0	24.82035827
6	EPIC 20149	1.0	2.13174	692	3.729700540	3.786832432	15.84661246	0.0115	5650.525135	2.375405405	0.78	4.391891891	0.4880606	13.74	528.1289062	42619.9375	109464.8840	366439.2867	257539.8966	1.012853470	-0.45458163	4.853507385	39.04349517
7	EPIC 20156	0.0	5.7968861	24.2	9.43	2.57	11.79028816	0.3232	4721.0	0.72	0.914594594	4.275945945	1.5774487	11.771	16298.25781	263963.25	107423.7834	399569.5218	292676.3382	0.965361445	-0.53864786	5.703095638	2858.794677
8	EPIC 20159	1.0	0.87703	1.2	25	1.0008	5.4	0.0114	5711.17	0.99	1.02	4.45	1.9681688	11.678	1119.8125	288391.5312	288383.3593	288438.6093	91.7734375	1.271994621	-0.71590034	3462402816	112.669998

Fig.11: Snapshot of the merged dataset

Classification

Being the application domain composed only of two target class, the prediction is made using the classification, a supervised machine learning technique where observations are assigned a known class value based upon their explanatory variables. The selected algorithms have been tested both on the full dimensionality of the dataset after the pre-processing (28 features with light curves attributes and 19 without) and on the reduced subspaces identified by PCA (even if it works transforming dimensions), testing the performances on both cases.
This study focuses on a binary classification of objects of interest as 'FALSE POSITIVE' (0) or 'CONFIRMED' (1) exoplanets using the following classifiers:

- K-Neighbors
- Gaussian Naïve Bayes
- Logistic Regression
- Random Forest
- Support Vector Machines

Learning Phase

To choose the best classifier model for our purpose, we tested our dataset against different classifier algorithms and, for two of them, we performed some test for parameter tuning and finally we compare all the models to choose the best one.
Before starting on building the model, we divided our dataset in train test (70%) and test set (30%) and we used the first one to train the models but, being our dataset unbalanced, we first apply Smote algorithm to oversample our dataset.

Parameters Tuning

K-nearest neighbors

The K-nearest neighbors' model (k-NN) is a popular, yet simple approach that classifies based on the voted majority among his k nearest neighbors based on a distance metric. Despite its simplicity, it shows a very good performance in several problems but it's very important to choose a good value for *k*; for this purpose, we tested the model several time using a different *k* belonging to the range [sqrt - sqrt/3, sqrt + sqrt/3], being sqrt is the square root of the number of samples.
For each trial we compute the mean squared error and, at the end, we choose the k that gives the lowest error.

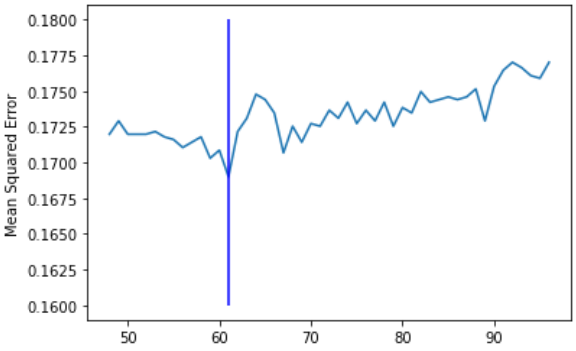

Fig.12: Tests visual representation

Random Forest

The Random Forest classifier is an ensemble in which many models (in this case decision trees) are trained over different samples of the data; the prediction of the ensemble corresponds to the majority class among all the models. Regarding the parameters for this classifier, we used 40 trees (Sturrock et al., 2019) and for the max depth of each tree we perform some test, and we select the case with higher performances.

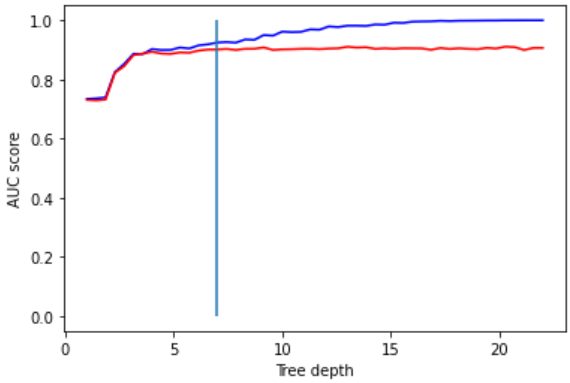

Fig.13: Tree depth selection

As first step we split again the training set into training set and test set and, for each trial, we compute the area under the curve (AUC) for the training and the test set; at the end of all tests, we chose the maximum depth such that the difference between the AUC value of training and testing was at most 0.3 so we prevent to overfit the model.
The other classifiers (Gaussian Naïve Bayes, Logistic Regression and Support Vector Machines) were trained using default attributes.

Performance Evaluation

All the classifiers were evaluated with the same strategy: k-fold cross validation. This means that a model is learned from the folds of the training set, and they are evaluated against the corresponding test fold: this the operation is repeated k times (10 in our study) and, at each iteration, a different fold acts as test set. This strategy gives us 10 different scores for each classifier and avoid concluding that one classifier is better than the others only by chance (we perform 10 executions because it is a good number to have statistics quite reliable).
All these trials are computed in the function *compute_all(data, target)* in the file *build_model.py* and this test is performed with and without the features extracted from light curves tested with and without applying PCA for select best attributes to compare results and find the better solution.
An execution of the *compute_all* function creates all the classifiers with the chosen attributes, then performs the cross validation calculating several metrics and returns the best classifier considering the higher score of the 'F1-score' being our original dataset unbalanced.
An example of execution which considers the extracted attributes and does not use PCA is shown below: in *Table* 1 are shown the values of F1-score of the 10 folds and in *Table* 2 are shown the mean values of the 10 iterations of all the considered metrics.

	KNN	Gaussian NB	Logistic Regression	Random Forest	RBF SVM
0	0.876877	0.848656	0.879747	0.924559	0.895476
1	0.876506	0.830601	0.88535	0.933544	0.905956
2	0.881098	0.80597	0.897764	0.910569	0.888189
3	0.875576	0.834473	0.874404	0.921569	0.881517
4	0.876161	0.830345	0.880645	0.919094	0.894231
5	0.85842	0.815115	0.865293	0.910543	0.86875
6	0.879389	0.835616	0.900641	0.94375	0.908517
7	0.884211	0.82973	0.894904	0.946541	0.904239
8	0.854167	0.831737	0.89557	0.940625	0.912226
9	0.888559	0.838621	0.910236	0.964006	0.922118

Table 2: F1-scores with 10-fold cross validation

	KNN	Gaussian NB	Logistic Regression	Random Forest	RBF SVM
Accuracy	0.866	0.799	0.886	0.93	0.895
Acc Std	0.012	0.015	0.013	0.016	0.015
Precision	0.819	0.719	0.872	0.915	0.871
Recall	0.94	0.983	0.906	0.949	0.927
F1-score	0.875	0.83	0.888	0.931	0.898
ROC AUC	0.949	0.927	0.954	0.978	0.96
Time (s)	1.021	0.044	0.454	5.586	6.044

Table 3: Means scores with 10-fold cross validation

Through the cross validation we can see that the Random Forest is the classifier with best results, so it has been chosen as classifiers despite the higher computational time.
Once the classifier has been chosen, it is tested against the test set and the result is compared with the one of the training sets to understand if the model is overfitted. In *Table 3* we can see this comparison in different cases, and we can understand that the model is not overfitted and that the best case is obtained using the full dimensionality (Planet Metadata and Light curves features) dataset and without using PCA.

	F1-Score (Training Set)	F1-Score (Test Set)
Full Dimensionality (Planet Metadata + Light curves features)	0.931	0.90804
PCA on Planet Metadata + Light curves features	0.899	0.86645
Planet Metadata	0.837	0.9011
PCA on Planet Metadata	0.901	0.8853

Table 4: Comparison of training and test results

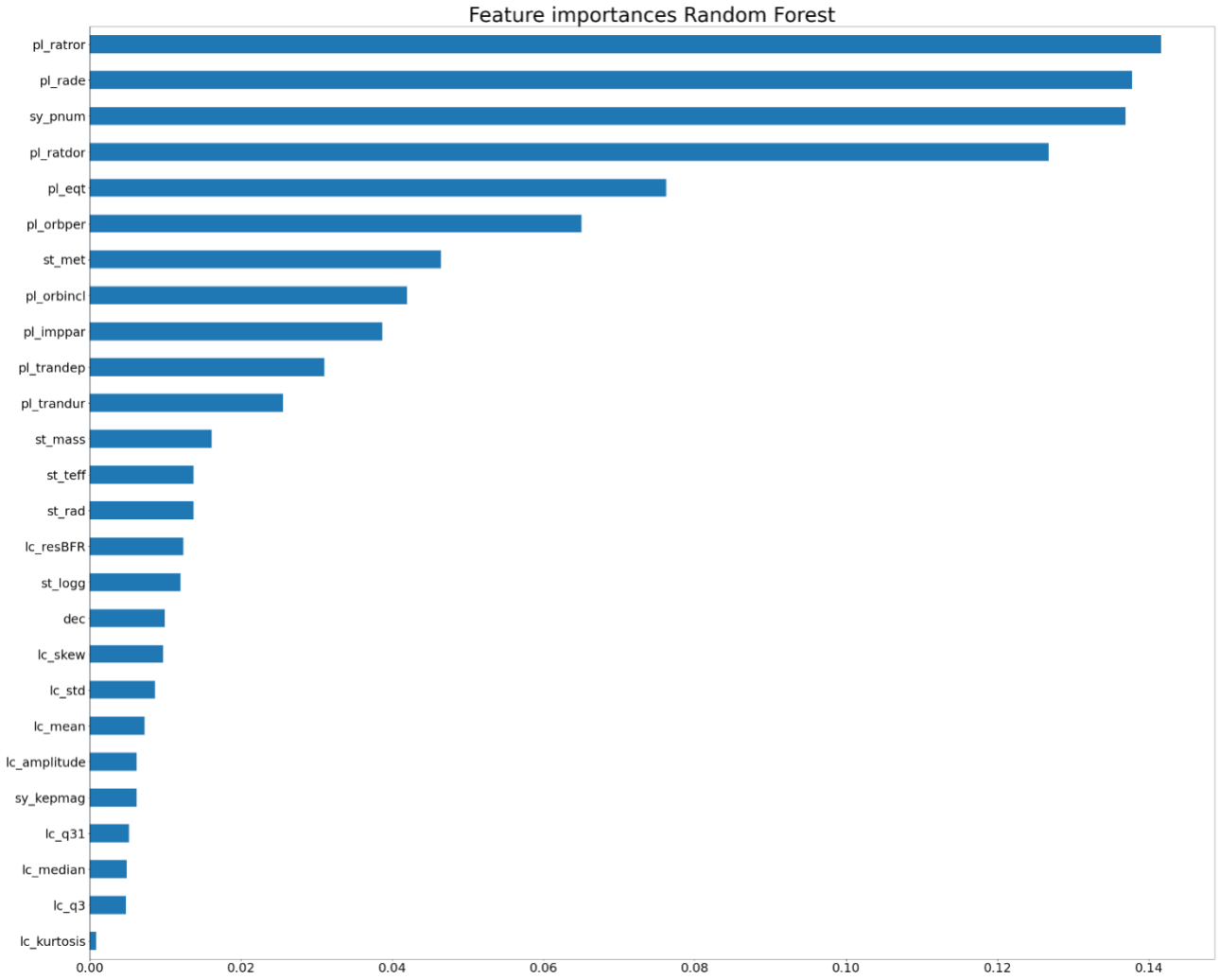

Fig.14: Feature importance of the merged dataset

Particularly (Fig:14), the radius of the object and the number of confirmed planets in the planetary system are the features who generates most impact on the prediction. The features with less importance are the ones extracted from the light curve, where the kurtosis and third quartile are the ones with less impact.

Candidate Classification

After choosing and training the best classifier, having accuracy of 93%, we performed the prediction of the target class on the objects of interest classified as 'CANDIDATE' in the initial dataset.

After this classification we merged the candidate objects classified as 'CONFIRMED' by the classifier with the 'CONFIRMED' objects of the initial dataset.
Starting from this new dataset we performed clustering of confirmed exoplanets to understand their similarity and dissimilarity.

Clustering

As already seen, over 5000 exoplanets have been detected so far, and their diversity is remarkable. In this section the aim is to determine the main types of exoplanets, develop a method that automatically associates' exoplanets to their type, classifying them into labels with a machine learning algorithm.
Given the planetary mass, planetary radius, planetary equilibrium temperature and orbital period, we used different clustering methods to find relations and significant groups between exoplanets.
For the clustering study two different kinds of clustering algorithms were used: K-Means clustering and Agglomerative clustering.
The idea of using the DBSCAN algorithm to evaluate the differences in the output to analyse more in detail the kinds of clusters but, based on the distribution of the data points (as shown later in the section), its performance was low, identifying most of the data as outliers.

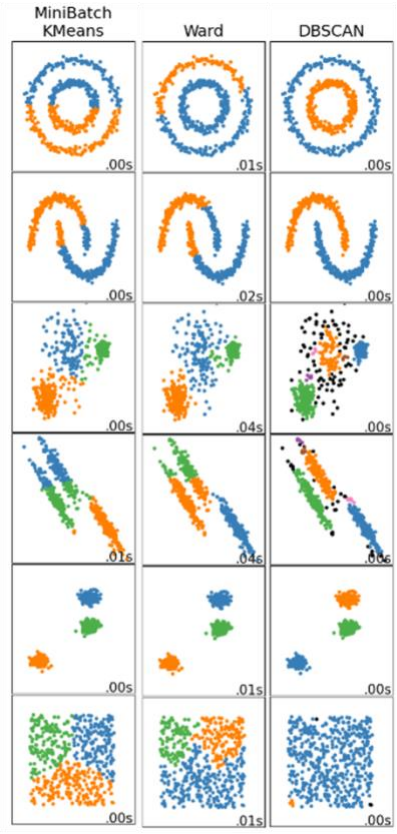

Fig.15: Sklearn's examples of clustering algorithms of different distributions

The orbital parameters of the known exoplanets span several orders of magnitude and are clearly distinct from those of the Solar system planets.
In addition, a complete characterization is difficult because only few observables can be measured for many planets so there are a lot of missing, particularly regarding planetary mass, but fortunately,

in the recent years machine learning algorithms have been developed to estimate these parameters, namely by [Ulmer-Moll et al., 2019] for the planetary radius and
[Tasker et al., 2020] for the planetary mass.
Apart from this, new measurements will keep being performed. This means that, in the next years, more and more exoplanets will be candidates for classification.

There are different classifications for Exoplanets like the NASA classification:

- Gas Giants: M >> MJ, R >> RJ with possibly rellay low orbital periods.
- Neptunian: ME << M < MJ, RE < R < RJ.
- Super-Earth: M ~ 10*ME. R ~ 2*ME not gaseous.
- Terrestrial: 0.5*RE < R < 2*RE not gaseous.

Or the classification defined by [Laughlin, 2018]:

- Hot Jupiters: M ~ MJ, P ~ 3 days.
- Long Period Giants: M ~ MJ, P > 100 days.
- Non-Giants: all the other exoplanets.

So, the aim of this analysis is to determine these kinds of different classifications, or even find different interesting groups.
The data used was the same used for the classification, after removing the FALSE POSITIVEs and filtering the ones with the attributes cited above.

Pre-analysis

The first step for the clustering analysis was to determine whether the dataset contained meaningful clusters.
One of the means to at least assess the level of spatial randomness of the data is the Hopkins statistic, with the aim to reject the null hypothesis stating that the data is generated by a Poisson point process and are thus uniformly randomly distributed.
The scores given by this test can be described like this:

- A value close to 1 tends to indicate the data is highly clustered.
- A value close to 0.5 tends to indicate random data.
- A value close to 0 tends to indicate uniformly distributed data.

To deepen the analysis different scores were computed for different set of attributes with the following results:

```
['pl_rade ', 'pl_bmasse ', 'pl_eqt ', 'pl_orbper ', 'st_mass ', 'st_rad ', 'st_met ']
0.9881837530203653

['pl_rade ', 'pl_bmasse ', 'pl_eqt ', 'pl_orbper ', 'st_mass ', 'st_rad ']
0.9878832942026641

['pl_rade ', 'pl_bmasse ', 'pl_eqt ', 'pl_orbper ']
0.9883867284773554

['pl_rade ', 'pl_bmasse ', 'pl_eqt ']
0.9085573421168046

['pl_rade ', 'pl_bmasse ']
0.8933657980306161
```

Fig.16: Hopkins scores on different set of attributes

The five computed statistics have all high scores (Fig.16), but particularly the ones having all the most important attributes regarding the planet and its host star.
All the attributes about the planet were chosen [Ana C. Barboza et al., 2020] because they had the highest score, to ultimately reduce the dimensionality to only the most important attribute and in order the represents better the clusters.

K-Means

The first algorithm tested was the K-Means algorithm, with the initial cluster centre selection "k-means++".
The first step was to find the number of clusters to consider in order to have a good analysis.
Both elbow method and silhouette score were used so to compare them and have multiple sources from which choose the optimal number of clusters (the graphs represented are with "k-means++" initialisation, as shown further in the study).

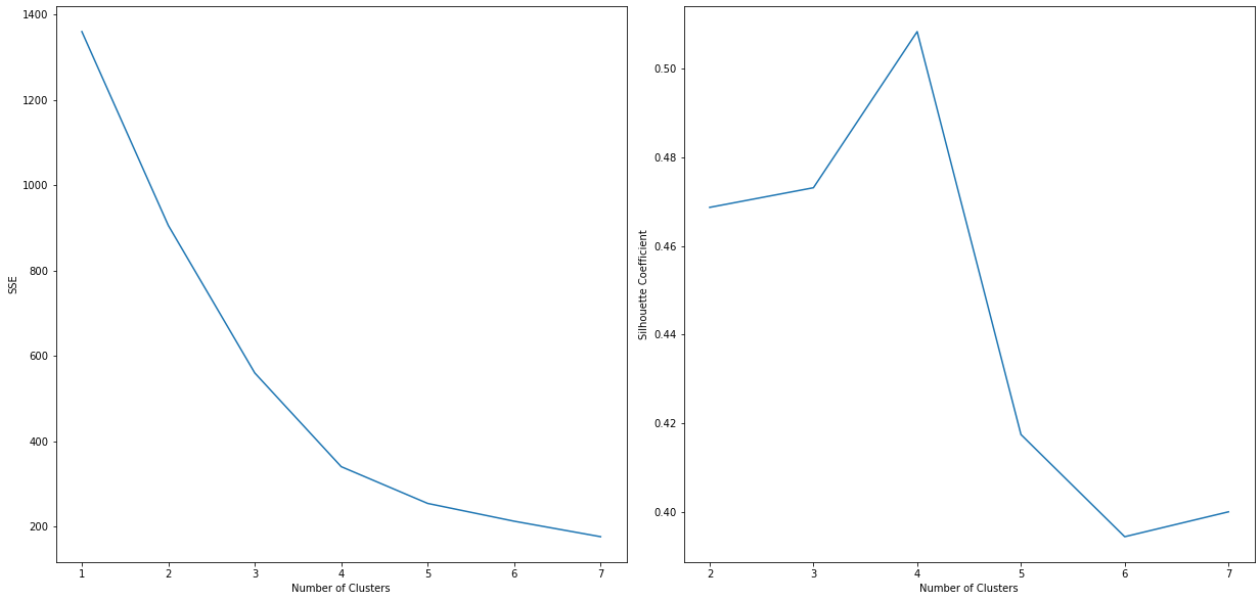

Fig.17, 18: Elbow method and Silhouette score graph for the optimal number for clusters

Fig.17 describes the inertia of the data points in function of the number of clusters, from which can be seen that, starting from 4 clusters, the information gain tends to drop significantly.
In the same way, the silhouette coefficient (Fig.18) describing the cohesion between points of the same cluster in function of their number presents an obvious peak at 4, reaffirming the first analysis.
Considering these diminishing returns, 4 was chosen as the optimal number of clusters to divide the dataset to follow up with the analysis.
Following is reported the outcome, represented as different scatterplot on the combinations of different attributes in logarithmic scale:

Fig. 19, 20, 21, 22, 23, 24: K-Means results displayed on all the combinations of attributes

As we can see, particularly from Fig.21, representing the mass in function of radius as they seem to be the most important attributes, that there 3 main clusters, two of which contain the Earth and Jupiter. With this initial analysis we can deduce that there are at least 3 main groups of exoplanets:

- Hot Jupiter: R > RJ, MJ < M < 3*MJ.
- Super-massive Giants: RJ~R, M >> MJ.
- Others: 0.5*RE < R < 10*RE, 0.5*ME < M < 100*ME.

Given the limits of the K-Means algorithm, the intervals of the other planets are broad, specifically the mass. On the other hand, for the Hot Jupiter planets the clusters are much more defined dividing effectively the different kinds of super-massive planets in different categories.
Analysing the 3 graphs containing the orbital period, there is a third "cluster", containing only one data point.
This outlier, being an extreme regarding orbital period but having a mass and radius like Jupiter, could be defined in two ways:

- An extreme Long Period Giant.
- A Jupiter-like rogue planet.

In any case, it could very well represent a cluster on its own.
Apart from the different clusters is interesting how the last graph describing the equilibrium temperature in function of the orbital period has a distribution like an exponential one.
Finally, a specific score was computed in order to determine, per cluster, which of the attributes was the most relevant for the Within-Cluster Sum of Squares minimization through finding the maximum absolute centroid dimensional movement:

Cluster	Most important attributes	
Yellow	***pl_rade***	***pl_eqt***
	0.9956512124212715	***0.8505620077170917***
Red	***pl_rade***	***pl_eqt***
	0.7054046549603565	0.6103352701157794
Green	***pl_orbper***	***pl_bmasse***
	18.462492828330653	1.8238282403683042)
Blue	***pl_bmasse***	***pl_rade***
	5.796564321588109	0.7988947409228376)

Table 5: Most important attributes for the K-Means clustering

Table 5 describes the importance of the scaled feature during the clustering, confirming the examination just conducted.
A further analysis was overseen by selecting the initial centroids with a "random" input, starting again from the different scores for cluster number selection to the clustering itself:

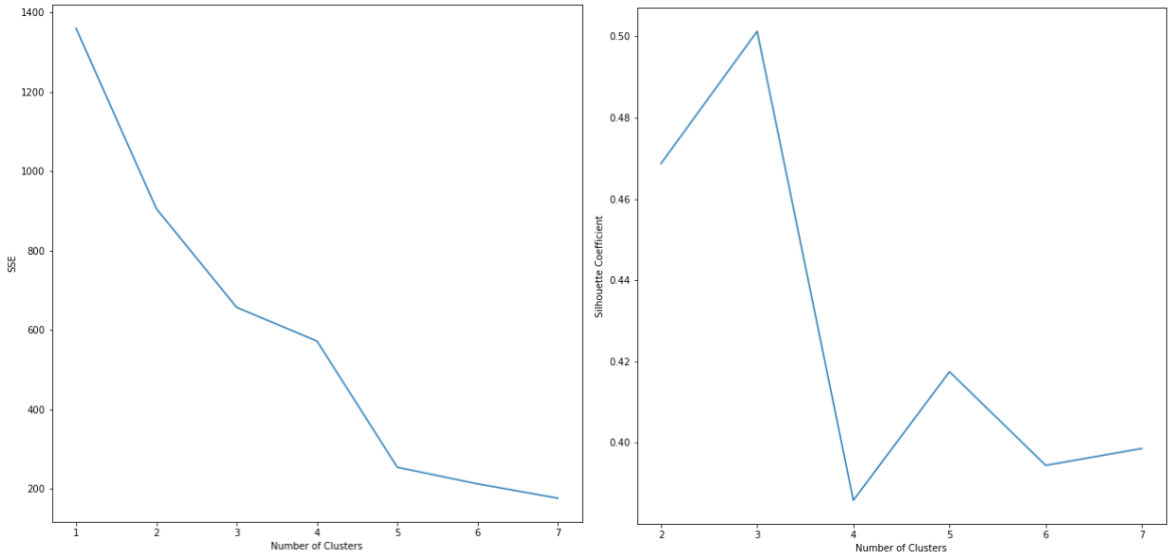

Fig. 25, 26: K-Means Elbow and Silhouette with "random" initialization

In this case there are two elbows as well as two peaks, so a parallel study was conducted (only plots with the mass in function of the radius are represented for the sake of simplicity)

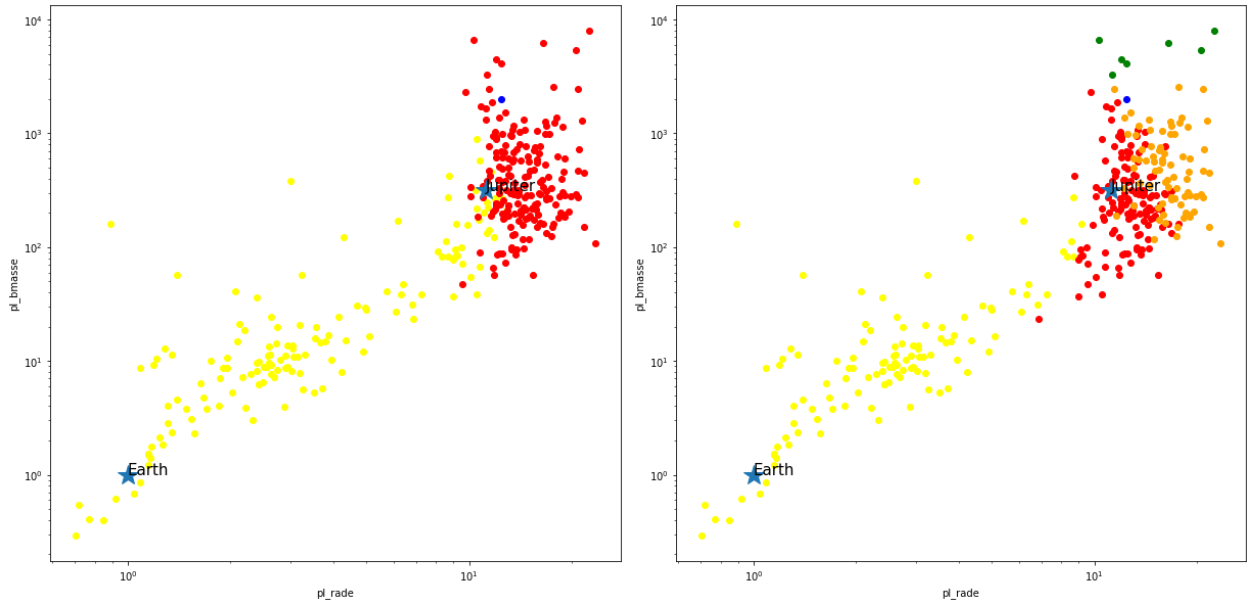

Fig. 27, 28: K-Means partial results with "random" initialization

Selecting 3 as the number of clusters gives an output like the "k-means++" initialization, but there is a loss of information as the Hot Jupiter and Super-massive Giants clusters are fused together even if very different in mass. On the other hand, with 5 as the number of clusters the algorithm finds the Others and the Super-massive Giants but creates two other clusters that basically overlap each other's in each dimension.
In the end it becomes obvious that the best initialisation for the K-Means algorithm is 4 clusters with "k-means++" centroids selection.

Hierarchical clustering

The second approach was to use a hierarchical clustering algorithm, specifically Agglomerative clustering, in order to further the analysis and compare the results.
The first step was to build the dendrogram representing the clusters to evaluate the existence of first similarities to follow up with the study.

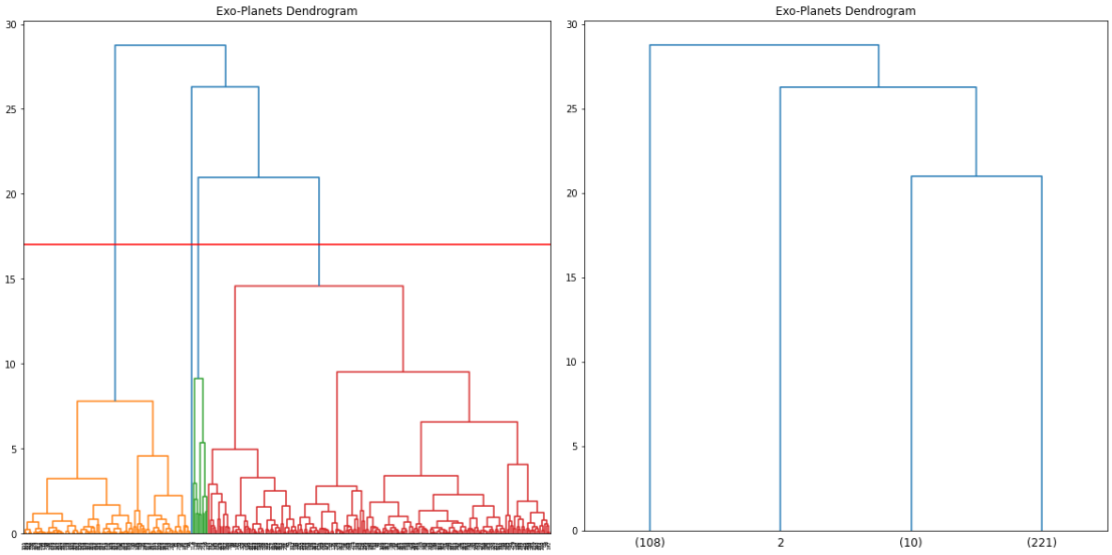

Fig. 29, 30: Dendrogram representing the clusters and tree cut at p=4 (red line)

The dendrogram was built by considering a Euclidean distance between points and using the “ward” method, to minimise the total within-cluster variance.
It’s evident how cutting the tree at a specific level in order obtain 4 different clusters, the numerosity of each group is like the one already seen in the K-Means analysis.
The second step was to find the clusters using the Agglomerative clustering and build the different scatter plot like for the K-Means study.

The results displayed on the graphs are very similar to those obtained with the K-Means algorithm, but the cluster created are much more defined.
While the cluster containing the Earth has still very distant boundaries, albeit less than the previous analysis, the cluster containing the Jupiter-like exoplanets is much more compact, defining a very specific class of celestial objects, the Hot Jupiters.
The remaining two clusters, the Super-massive Giants and the single point one, are basically the same as in the K-Means analysis.

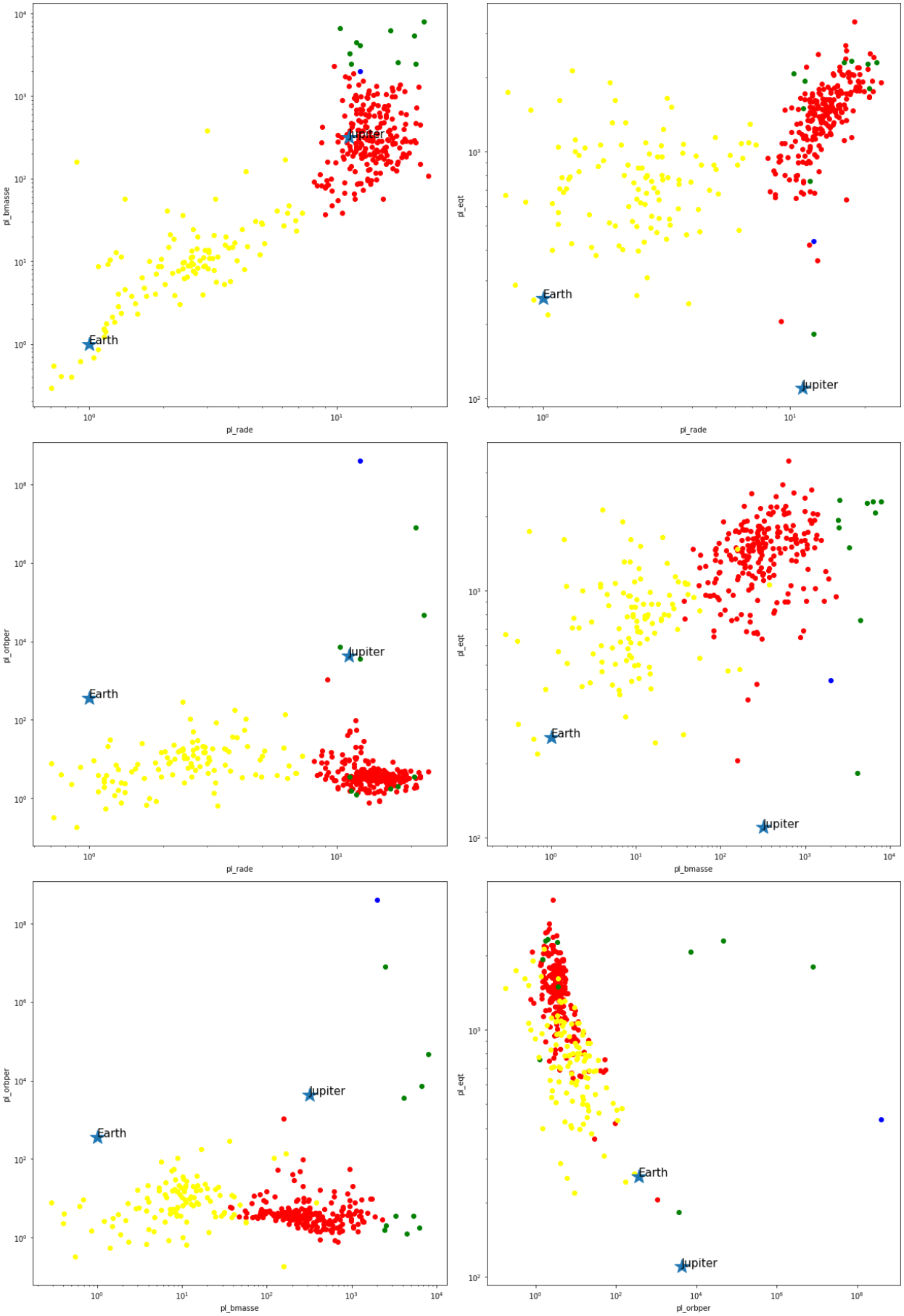

Fig. 31, 32, 33, 34, 35, 36: AGNES results displayed on all the combinations of attributes

Other than the “ward” method the “complete” linkage method was tested, but it returned worse results of the former:

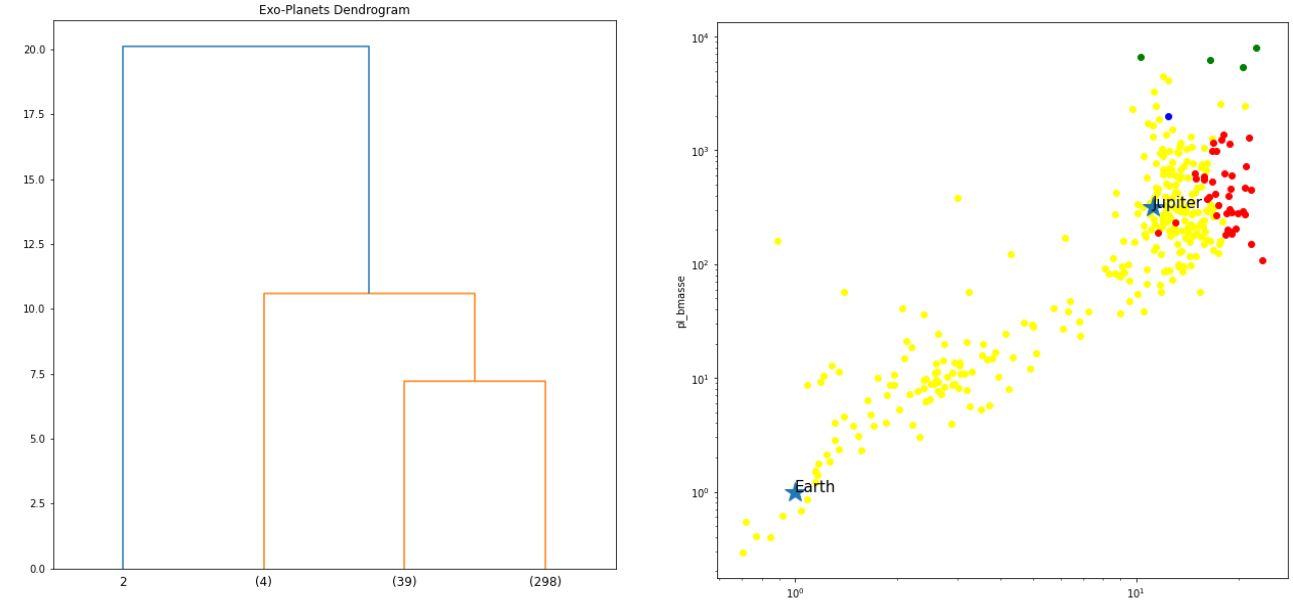

Fig. 37, 38: Dendrogram and AGNES results with “complete” linkage

From the scatterplot becomes evident that with this kind of linkage multiple clusters are merged in less defined ones, resulting in an excessive loss of information.
As done before the Agglomerative clustering algorithm was tested with different number of clusters in input, but the results were very similar to K-Means, although more precise

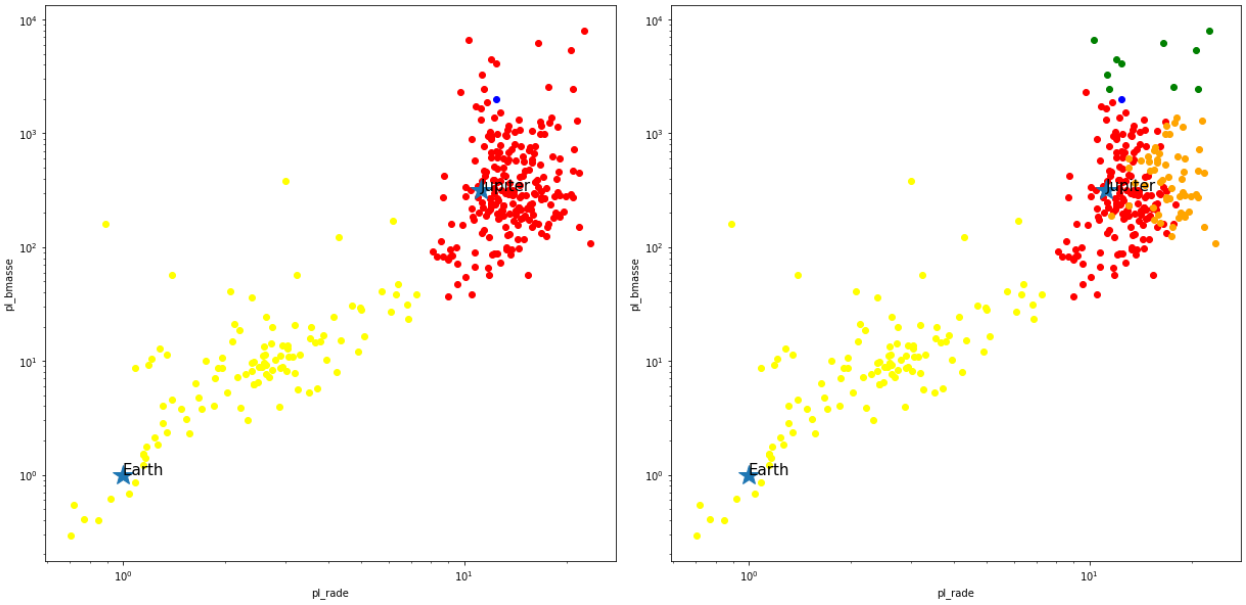

Fig. 39, 40: AGNES results with 3 and 5 clusters

Due to the similarities between the two different clustering studies, we can confirm that all the inference made for the K-Means algorithm are valid for the Agglomerative clustering algorithm too, further improving the trustworthiness of the analysis.
With the results obtained regarding the definitions of the clusters we can affirm with high confidence that the AGNES algorithm works better than K-Means on generating meaningful groups on the NASA dataset of confirmed exoplanets.

DBSCAN

The elbow method was conducted in a similar way to find the optimal epsilon for the DBSCAN algorithm:

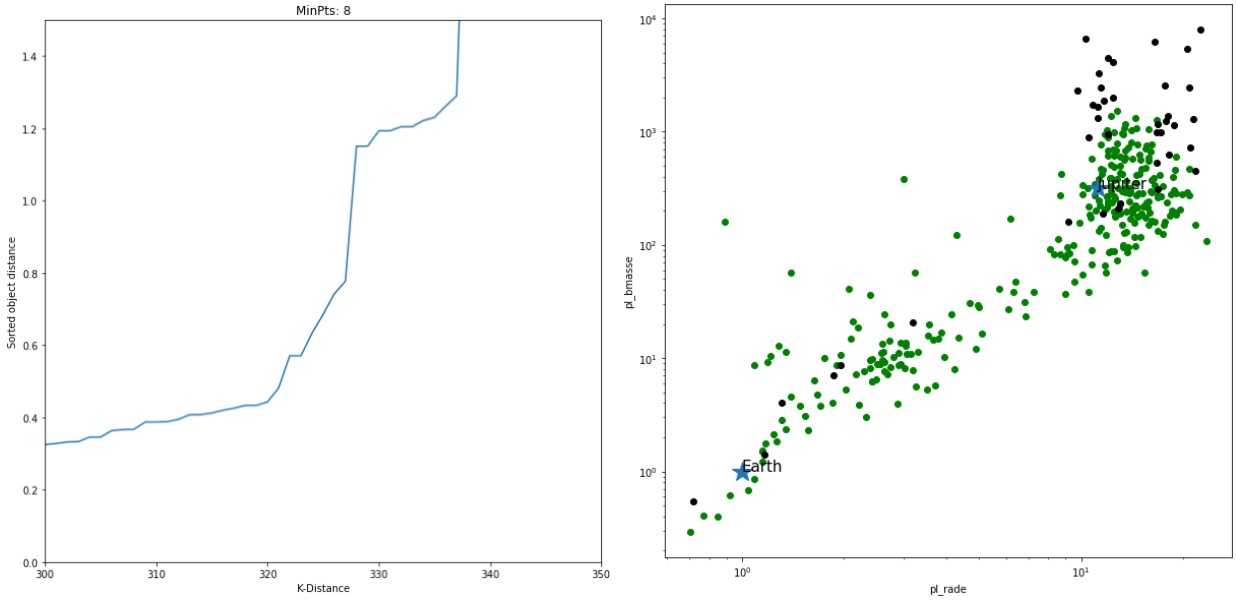

Fig. 41, 42: AGNES results with 3 and 5 clusters

From Fig.41, by using as minimum points per cluster double the number of attributes, as generally described in the literature, the optimal epsilon is [0.5, 1.3]. Even then, an algorithm like DBSCAN is not suited for the domain of exoplanets classification problem.

This is because the dataset is sparse and doesn't follow a particular pattern, so a density-based clustering algorithm that don't use centroids is not a good choice. A test was conducted (Fig.42) but in line with what just analysed, the algorithm returned a high percentage of outliers (black points) and basically a single class.

Conclusions

The primary objectives of this study were to create a machine learning model to automate the classification of Kepler and K2 cumulative object of interest data features and light curves, and to analyse related groups of objects classified as exoplanets through a clustering analysis. To achieve this goal, a comprehensive machine learning pipeline was created to engineer the data, train, and test models. This process attempted to produce a set of candidate features after accounting for missingness, statistical inconsistencies, correlations, and bias. These candidate features were used to train K-Nearest Neighbours (KNN), Support Vector Machine (SVM), Gaussian NB, Logistic Regression and Random Forest classifiers. Based on model performance and explainable feature importance, the Random Forest classifier was chosen as the primary model for testing and deployment. The Random Forest classifier identified planet radius, planets in the host system, orbital distance and period as features with the highest importance for classifying objects-of-interest. Objects-of-interest with a radius between Mars and Neptune, a transit light curves like other confirmed planets, and other planets in their host system are most likely to be classified as exoplanets. Finally, thanks to K-Means, but more importantly to the Agglomerative clustering algorithm, 3 main classes of exoplanets were identified, in line with the previous literature. AGNES identified Long Period Giants, Hot Jupiters and Super-massive Jupiters with high precision, while grouping other exoplanets with measurements lower than Jupiter all in a macro-cluster.

References

1. "Kepler and K2 Missions". *NASA Official Website*. Online. Retrieved from https://astrobiolo-gy.nasa.gov/missions/
2. "NASA Exoplanet Archive". *NASA Official Website*. Online. Retrieved from https://exoplanetarchive.ipac.caltech.edu/index.html Digital Objects identifiers: **DOI 10.26133/NEA4, DOI 10.26133/NEA19**
3. Barbara A. "Mikulski Archive for Space Telescopes" Online. Retrieved from https://archive.stsci.edu/
4. Lightkurve v2.3 API. Online. Retrieved from https://docs.lightkurve.org/#documentation
5. Astroquery Documentation. Online. Retrieved from https://astroquery.readthedocs.io/en/latest/mast/mast.html