Machine Learning Pipeline for Exoplanet Classification and Clustering

Data Mining and Machine Learning Project

Luana Bussu, Luca Caprioli

Introduction

Goal of the project

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, being the analysis of periodicities in star light-curves the most successful so far.

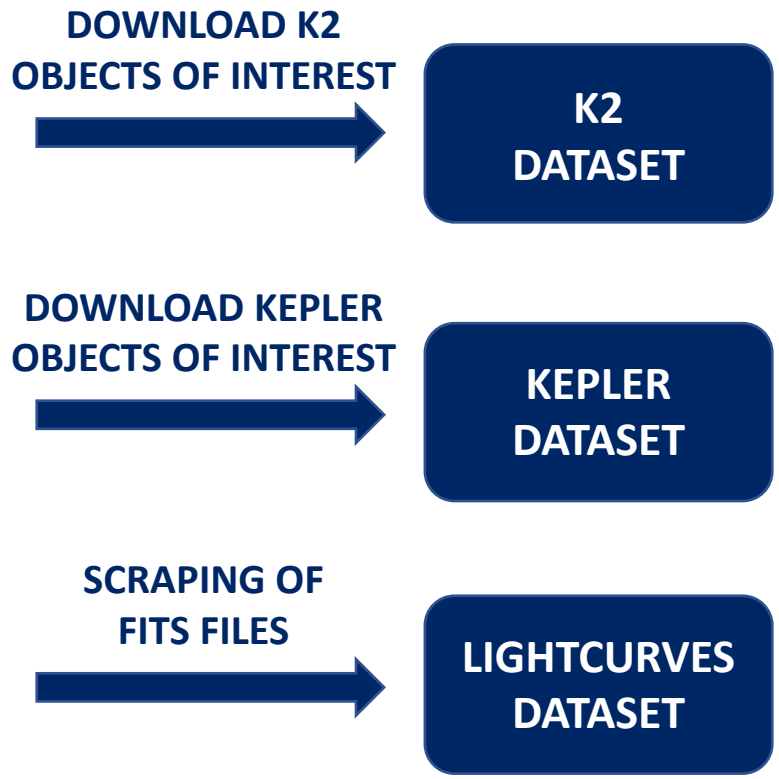
The goal of this project is to build and train a classifier able to recognize an exoplanet starting from the objects of interest of the NASA K2 and Kepler missions and, starting from confirmed exoplanets, to perform different clustering techniques to discover different classes of exoplanets.

Dataset

Dataset Building

Source: https://exoplanetarchive.ipac.caltech.edu/index.html

The dataset used in this application is made up of different datasets related to two NASA missions: Kepler and K2. Considering that most of the discovered exoplanets have been detected through the transit method, 9241 FITS files were collected through scraping



Dataset Description

- **Kepler Dataset**: 9500 rows and 140 columns
- **K2 Dataset**: 2600 rows and 290 columns

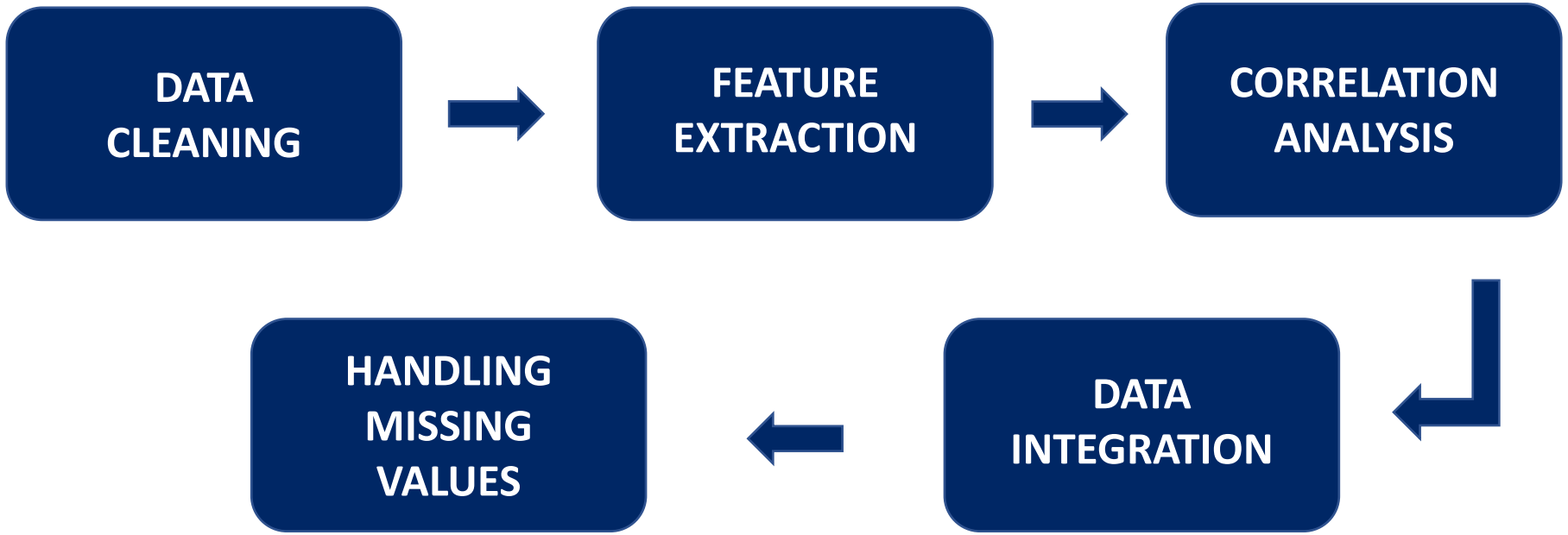
K2 Objects of Interests

	rowid	pl_name	hostname	pl_letter	k2_name	epic_hostname	hd_name	hip_name	tic_id	gaia_id	default_flag	disposition	disp_refname	sy_snum	sy_pnum	sy_mnum	cb_flag	discoverymethod	disc_year
0	1	BD+20 594 b	BD+20 594	b	K2-56 b	EPIC 210848071			TIC 26123781	Gaia DR2 5820093432631513 6	1.0	CONFIRMED	Espinoza et al. 2016	1.0	1.0	0.0	0.0	Transit	2016.0
1	2	BD+20 594 b	BD+20 594	b	K2-56 b	EPIC 210848071			TIC 26123781	Gaia DR2 5820093432631513 6	0.0	CONFIRMED	Espinoza et al. 2016	1.0	1.0	0.0	0.0	Transit	2016.0
2	3	BD+20 594 b	BD+20 594	b	K2-56 b	EPIC 210848071			TIC 26123781	Gaia DR2 5820093432631513 6	0.0	CONFIRMED	Espinoza et al. 2016	1.0	1.0	0.0	0.0	Transit	2016.0
3	4	EPIC 201111557.01	EPIC 201111557			EPIC 201111557			TIC 176942156	Gaia DR2 3596276829630866432	0.0	CANDIDATE	Livingston et al. 2018	1.0	0.0	0.0	0.0	Transit	2018.0
4	5	EPIC 201111557.01	EPIC 201111557			EPIC 201111557			TIC 176942156	Gaia DR2 3596276829630866432	1.0	CANDIDATE	Livingston et al. 2018	1.0	0.0	0.0	0.0	Transit	2018.0
5	6	EPIC 201126503.01	EPIC 201126503			EPIC 201126503			TIC 380166702	Gaia DR2 3598754063687508480	1.0	CANDIDATE	Vanderburg et al. 2016	1.0	0.0	0.0	0.0	Transit	2016.0
6	7	EPIC 201127519.01	EPIC 201127519			EPIC 201127519			TIC 96244568	Gaia DR2 3597255188821238016	0.0	CANDIDATE	Livingston et al. 2018	1.0	0.0	0.0	0.0	Transit	2018.0
7	8	EPIC 201127519.01	EPIC 201127519			EPIC 201127519			TIC 96244568	Gaia DR2 3597255188821238016	1.0	CANDIDATE	Livingston et al. 2018	1.0	0.0	0.0	0.0	Transit	2018.0
8	32	EPIC 201257461.01	EPIC 201257461			EPIC 201257461			TIC 446672940	Gaia DR2 3601830428502237312	1.0	FALSE POSITIVE	Montet et al. 2015	1.0	0.0	0.0	0.0	Transit	2015.0
9	33	EPIC 201257461.01	EPIC 201257461			EPIC 201257461			TIC 446672940	Gaia DR2 3601830428502237312	0.0	FALSE POSITIVE	Montet et al. 2015	1.0	0.0	0.0	0.0	Transit	2015.0
10	34	EPIC 201258341.01	EPIC 201258341			EPIC 201258341			TIC 421194234	Gaia DR2 3601784489532129280	1.0	CANDIDATE	Kruse et al. 2019	1.0	0.0	0.0	0.0	Transit	2019.0

pl_tranmiderr1	pl_tranmiderr2	pl_tranmidlim	pl_tsystemref	ttv_flag	pl_imppar	pl_impparerr1	pl_impparerr2	pl_impparlim	pl_trandep	pl_trandeperr1	pl_trandeperr2	pl_trandeplim	pl_trandur	pl_trandurerr1	pl_trandurerr2	pl_trandurlim	pl_ratdor	pl_ratdorerr1	pl_ratdorerr2	pl_ratdorlim	pl_ratror	pl_ratrorerr1	pl_ratrorerr2	pl_ratrorlim	pl_occdep
				0.0					49	0.0025	-0.0025	0.0					55.8	3.3	-3.3	0.0					
0.0042	-0.0047	0.0	BJD-TDB	0.0													55.8	3.3	-3.3	0.0	0.02204	0.00058	-0.00057	0.0	
0.002801	-0.002798	0.0		0.0													54.721028	5.823183	-12.995528	0.0	0.02259	0.00185	-0.00094	0.0	
0.002087	-0.002129	0.0	BJD	0.0													12.619867	2.838328	-8.018827	0.0	0.01692	0.00674	-0.00148	0.0	
0.0052	-0.0052	0.0	BJD	0.0	0.42	0.33	-0.28	0.0	2.268			0.0	1.9008			0.0	11.8	2.1	-3.0	0.0	0.0144	0.0014	-1	0.0	
		0.0	BJD	0.0	0.0			0.0	448			0.0	1.93			0.0					0.06005			0.0	
0.001116	-0.001211	0.0	BJD	0.0													17.277258	1.123995	-2.04578	0.0	0.11511	0.00492	-0.00336	0.0	
0.0005	-0.0005	0.0	BJD	0.0	0.17	0.14	-0.11	0.0	1.303			0.0	2.5008			0.0	18.1	0.2	-0.6	0.0	0.1058	0.0011	-0.0007	0.0	
		0.0	BJD	0.0	0.95			0.0	184			0.0	12.62			0.0					0.05345			0.0	
0.0397	-0.0397	0.0	BJD	0.0													6.19	0.52	-0.52	0.0					
0.0026	-0.0026	0.0	BJD	0.0					0.0402	0.0016	-0.0016	0.0	3.384	168	-216	0.0	21.4	7.7	-2.8	0.0	0.0183	0.0016	-0.0045	0.0	

Pre - Processing

Pre - Processing



Data Cleaning

Removing irrelevant attributes

'hostname'
'disc_year'
'disc_pubdate'
'pl_pubdate'
'releasedate'
'comments'
…

Removing features having no variance

data.drop(nunique[nunique == 1].index, axis=1, inplace=True)

Removing duplicate rows on 'pl_name'

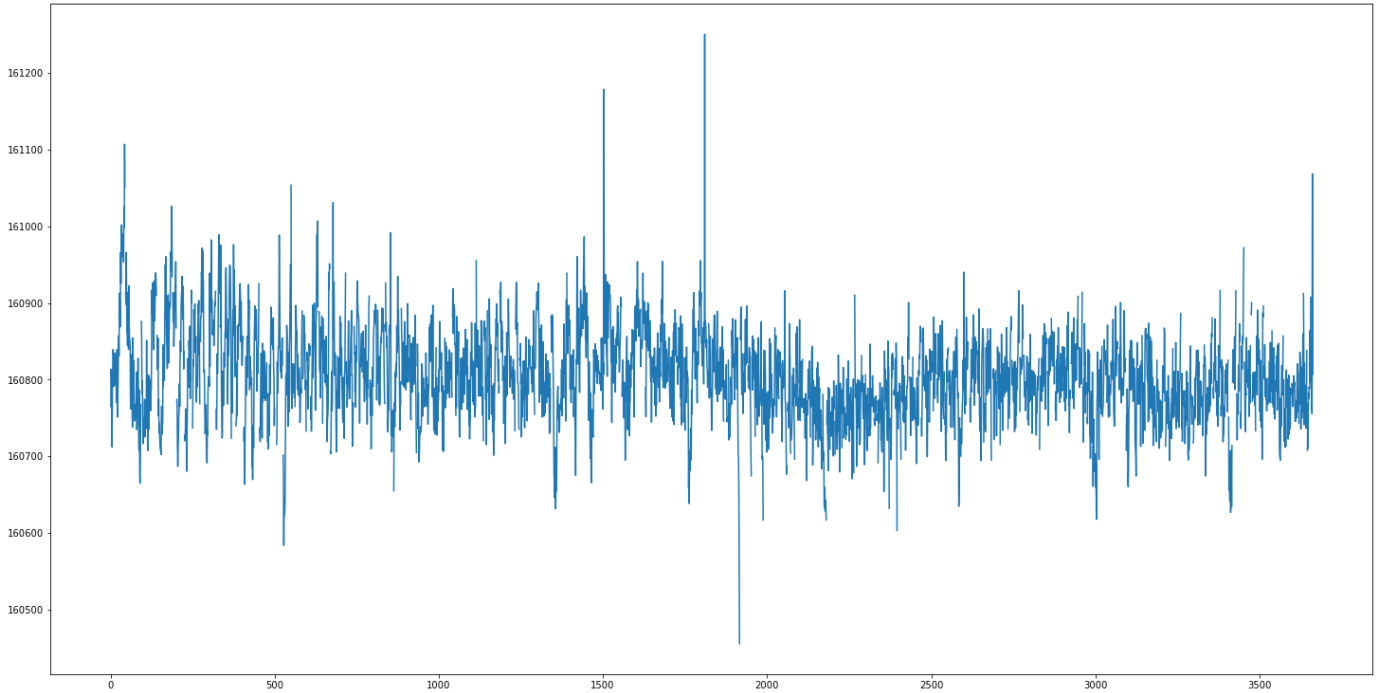
Removing measurement errors

Features named as '_err' or '_lim'

Changing class labels from string to numeric

'FALSE POSITIVE' → '0'
'CONFIRMED' → '1'
'CANDIDATE' → '2'

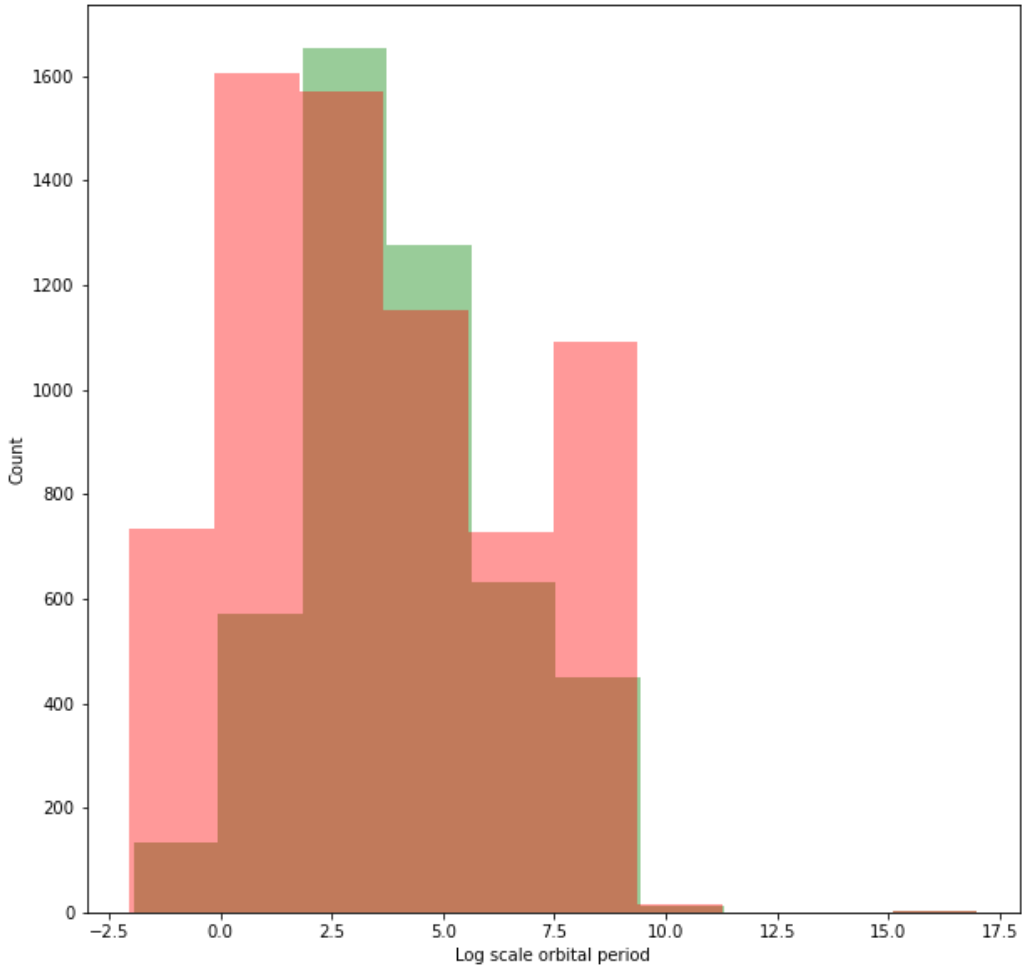
Feature Extraction

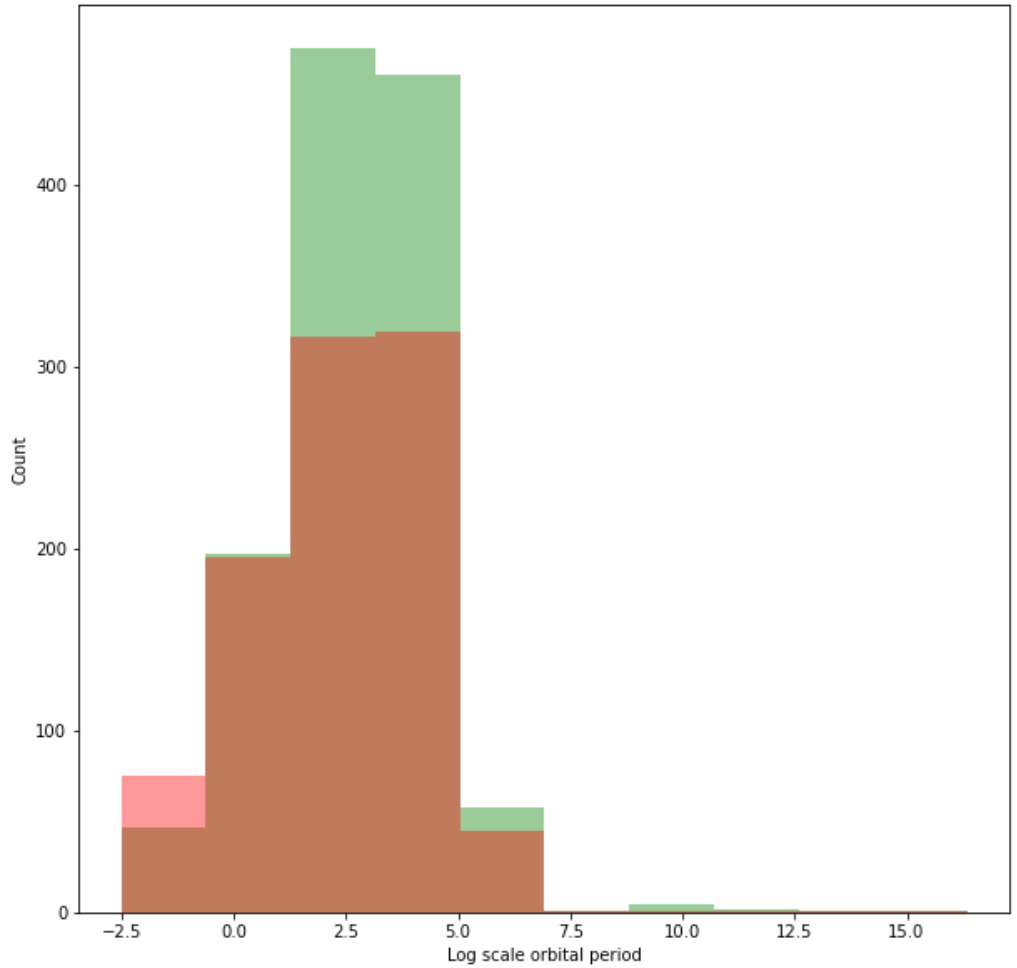


EXTRACT FEATURES → K2 DATASET

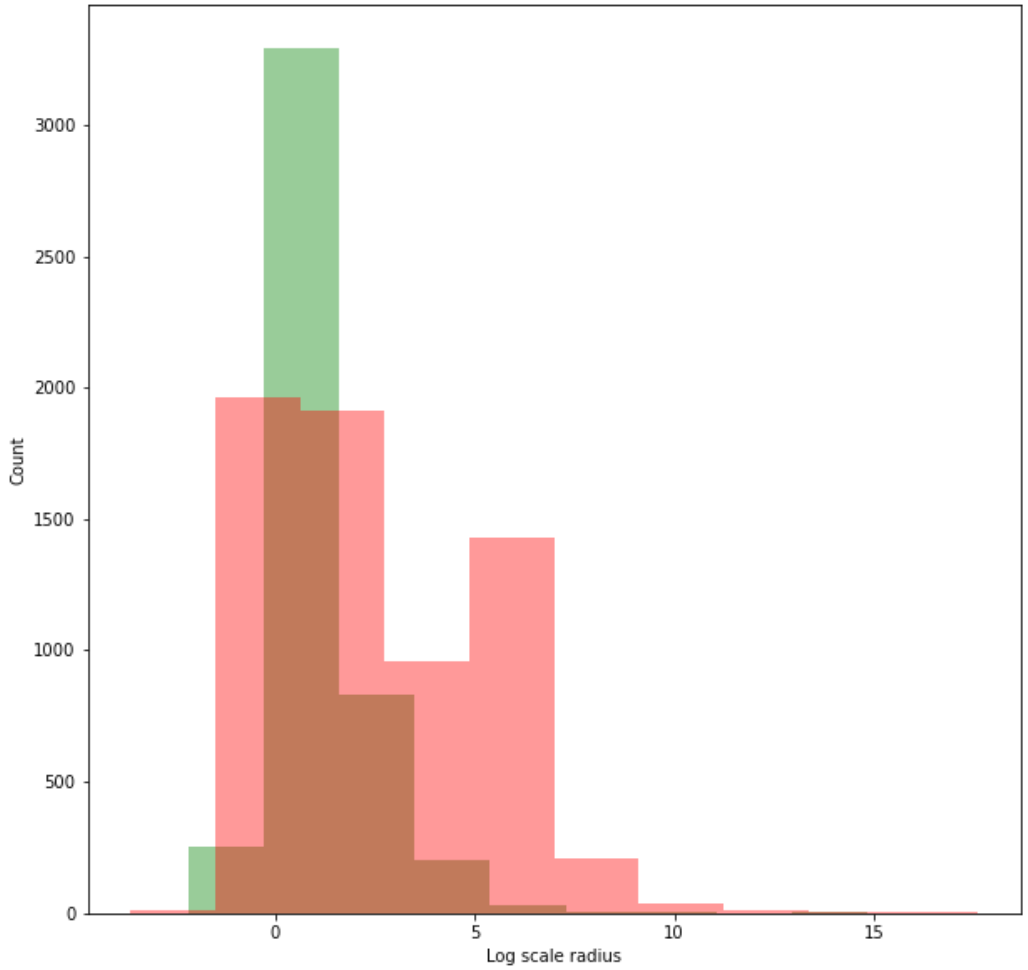
EXTRACT FEATURES → KEPLER DATASET

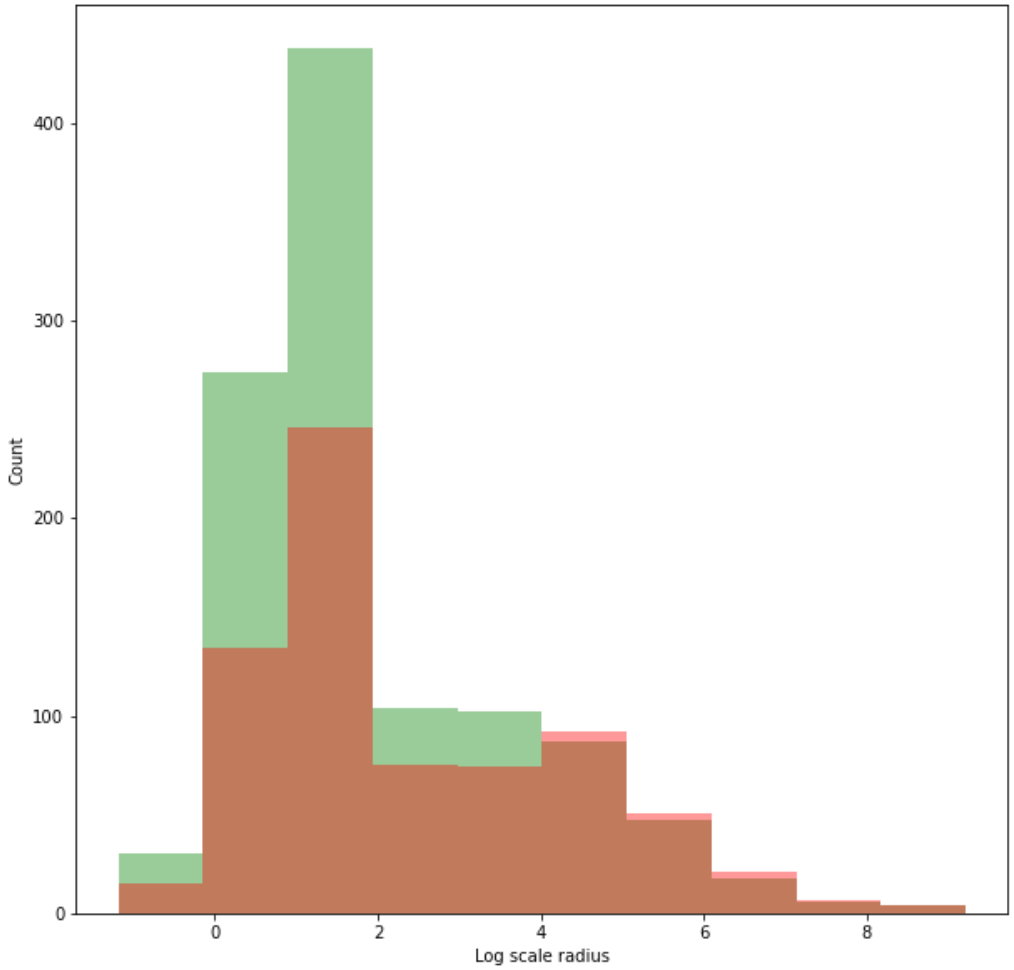
Univariate Analysis



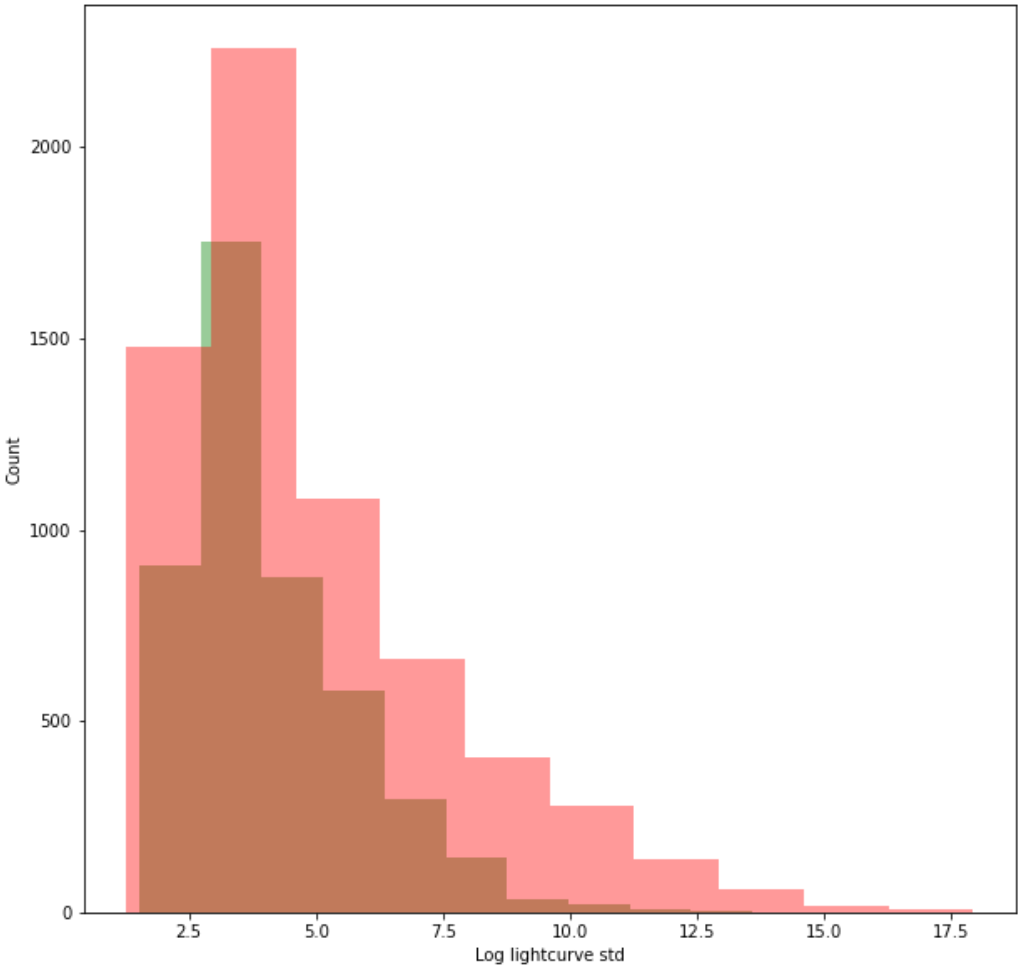


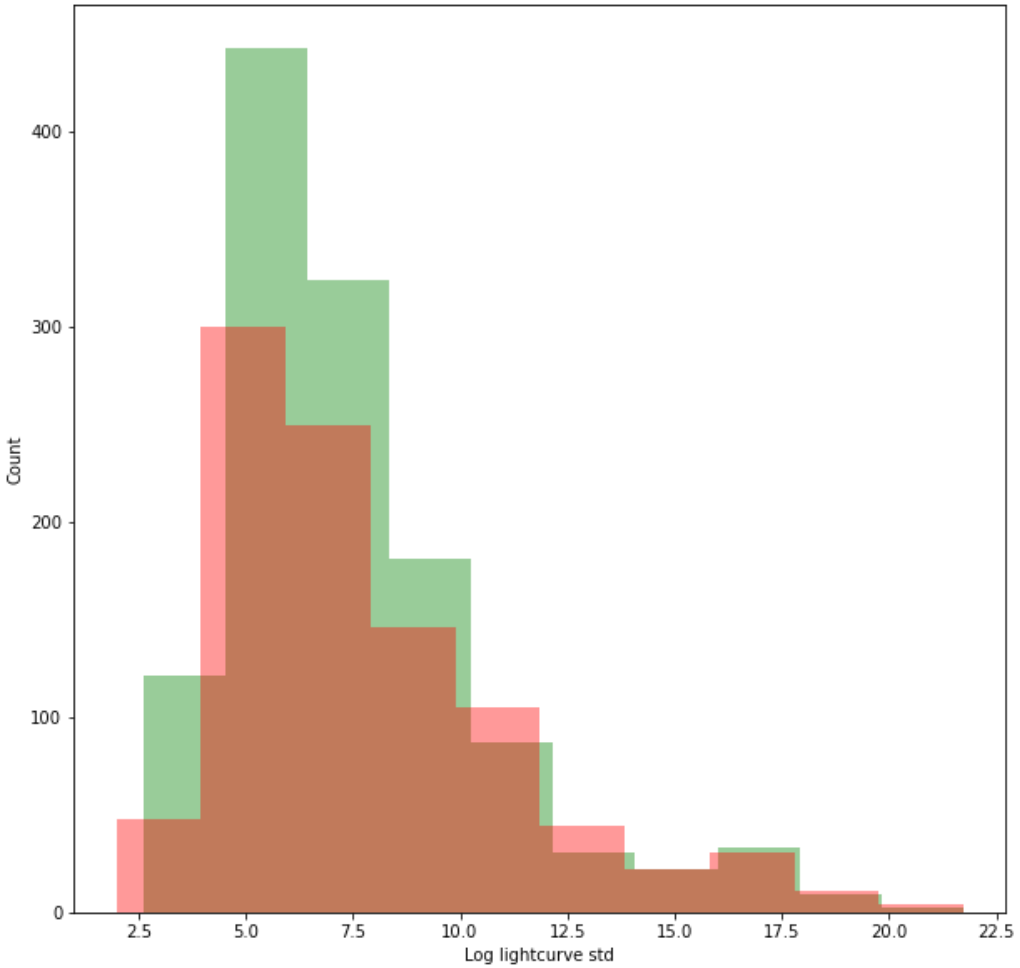
Univariate Analysis (II)



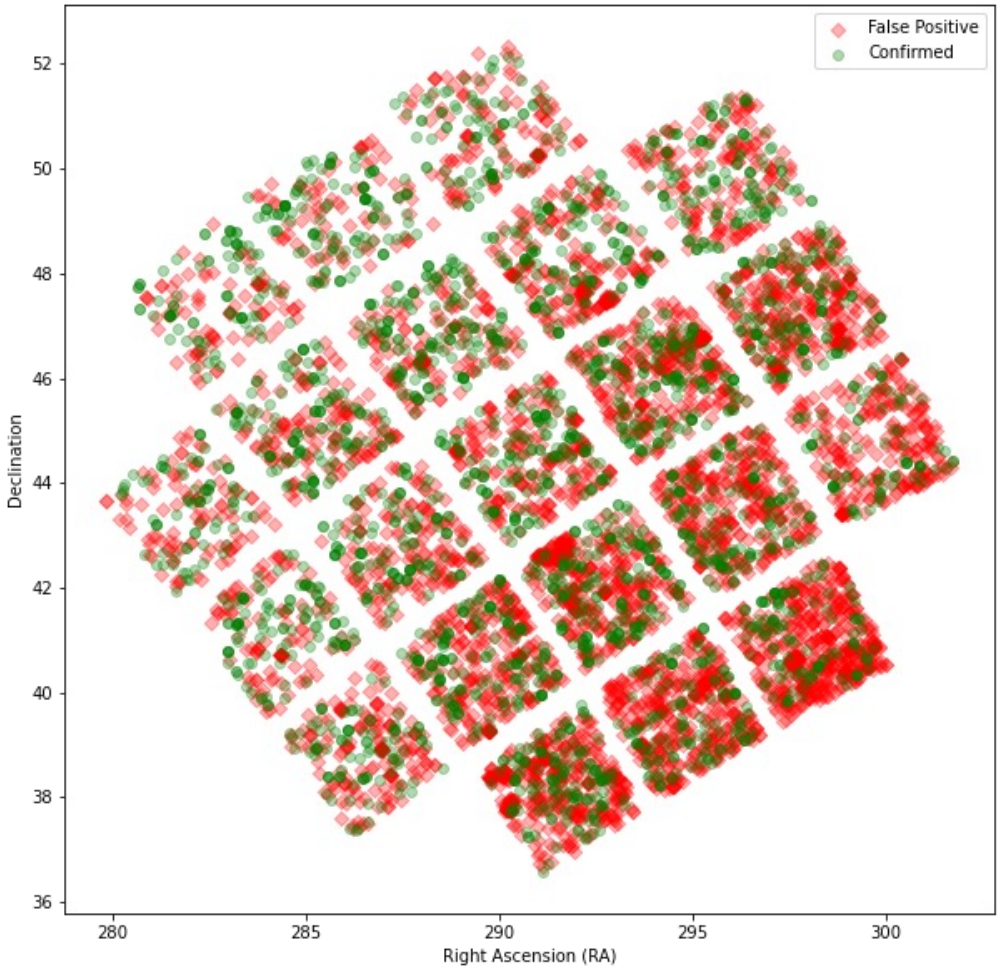


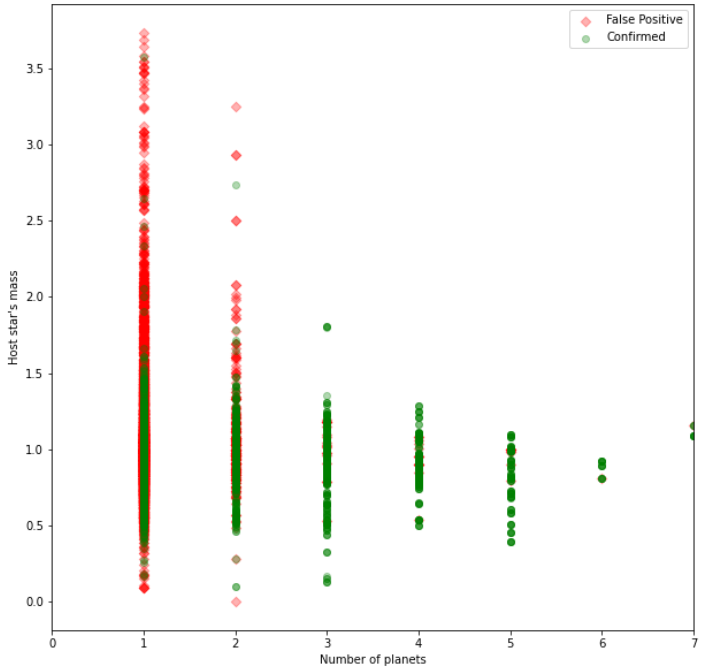
Univariate Analysis (III)

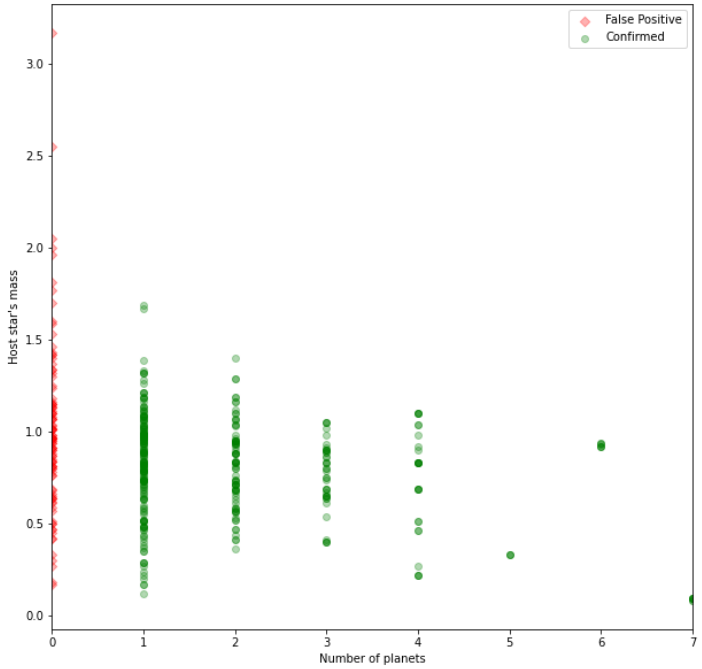




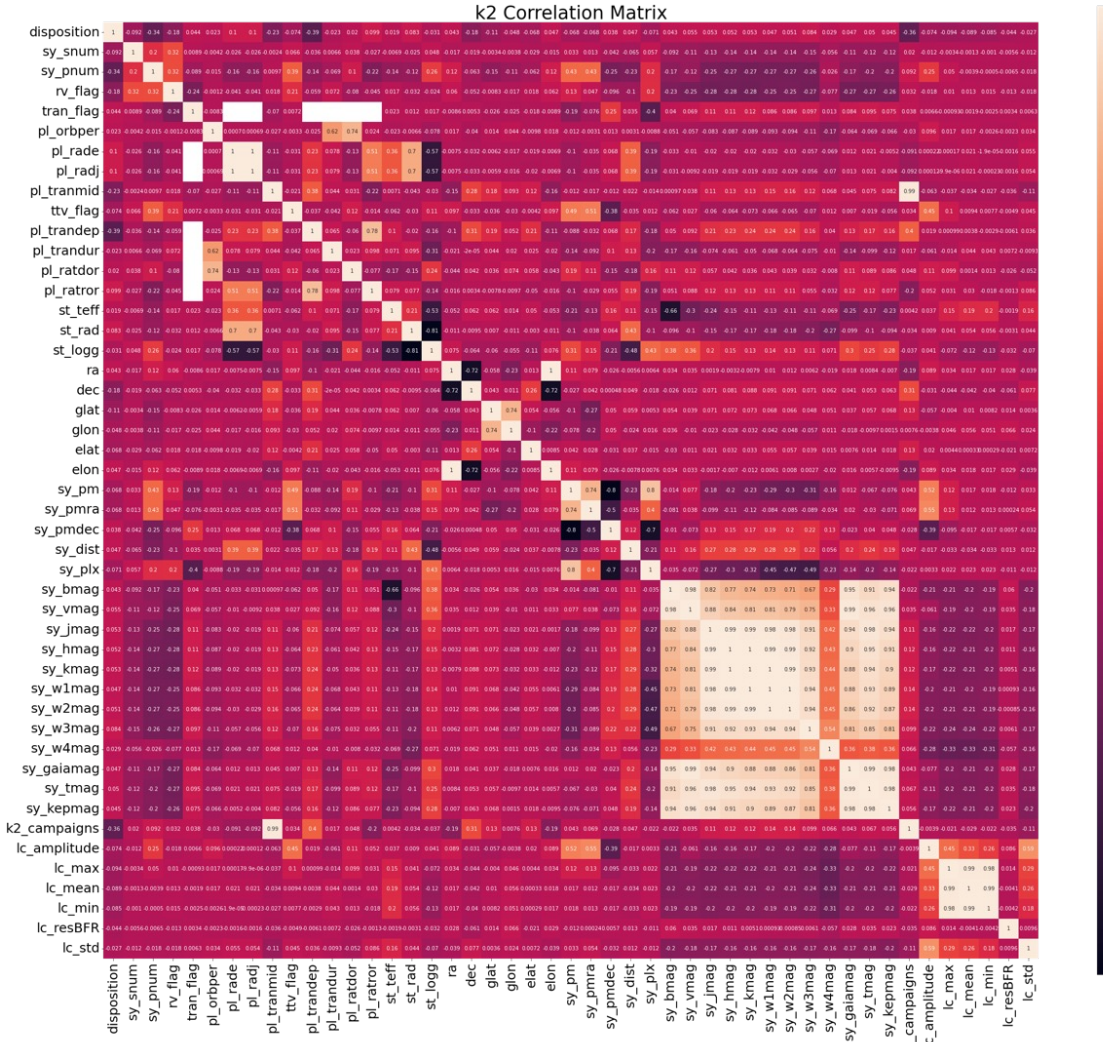
Bivariate Analysis

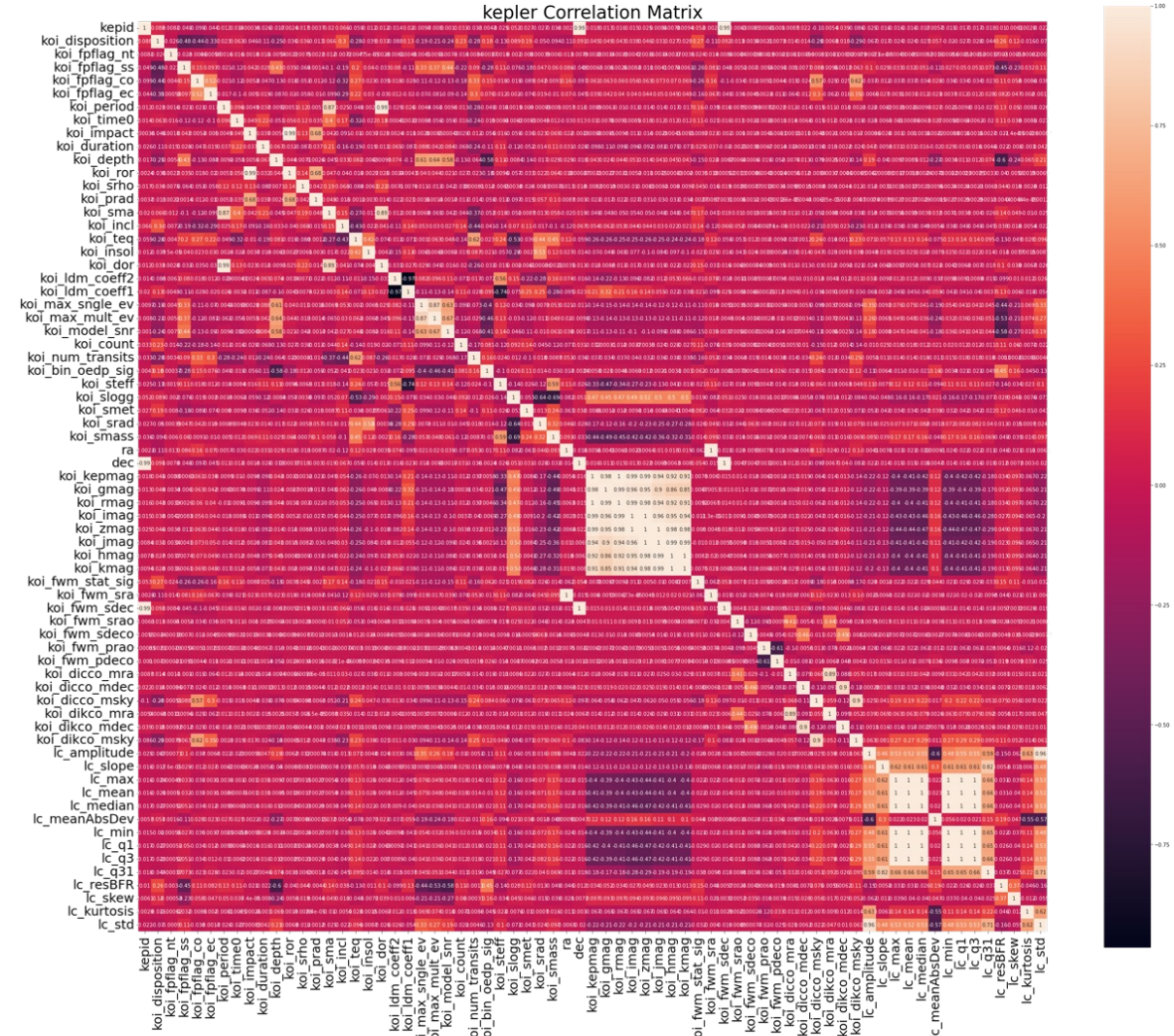






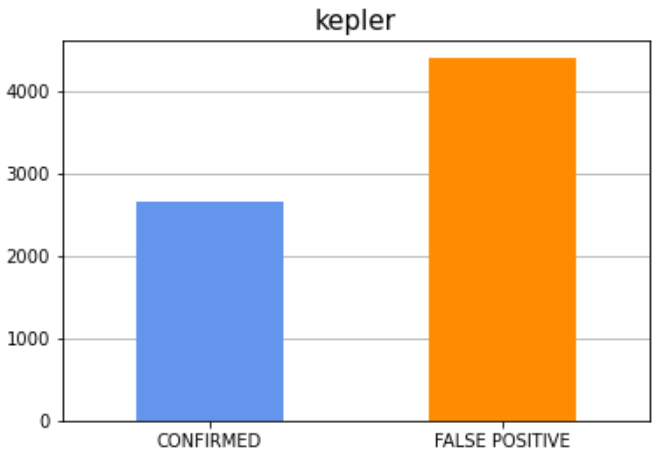
Correlation Analysis

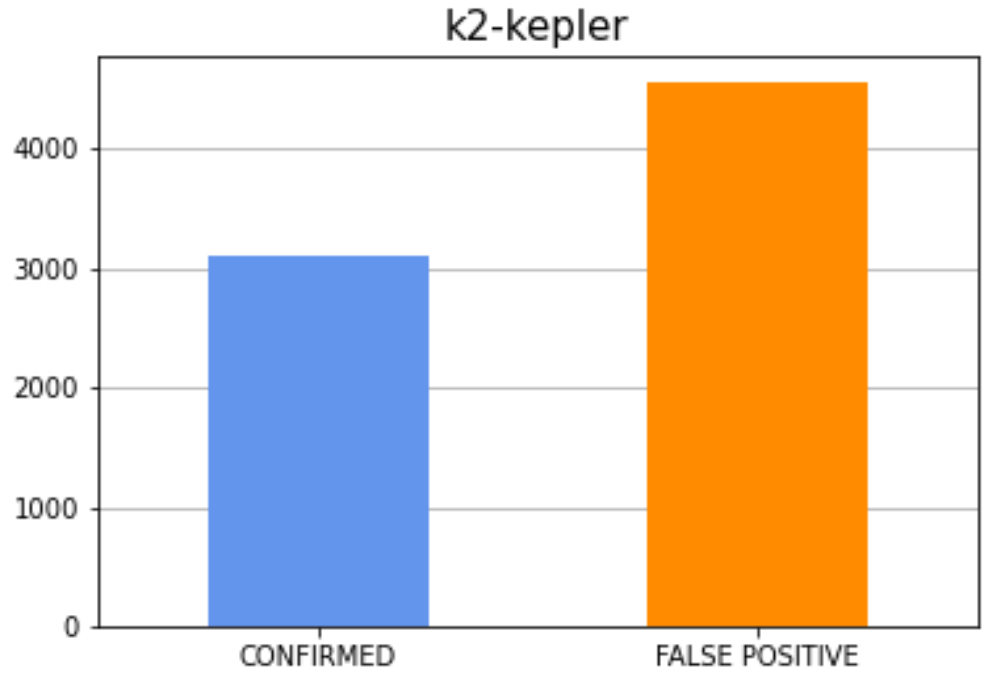




Data Integration

Data Integration (II)





Handling Missing Values

Removing high percentage of missing values

> 30%

K2 - KEPLER DATASET WITH MISSING VALUES

CHOOSE K

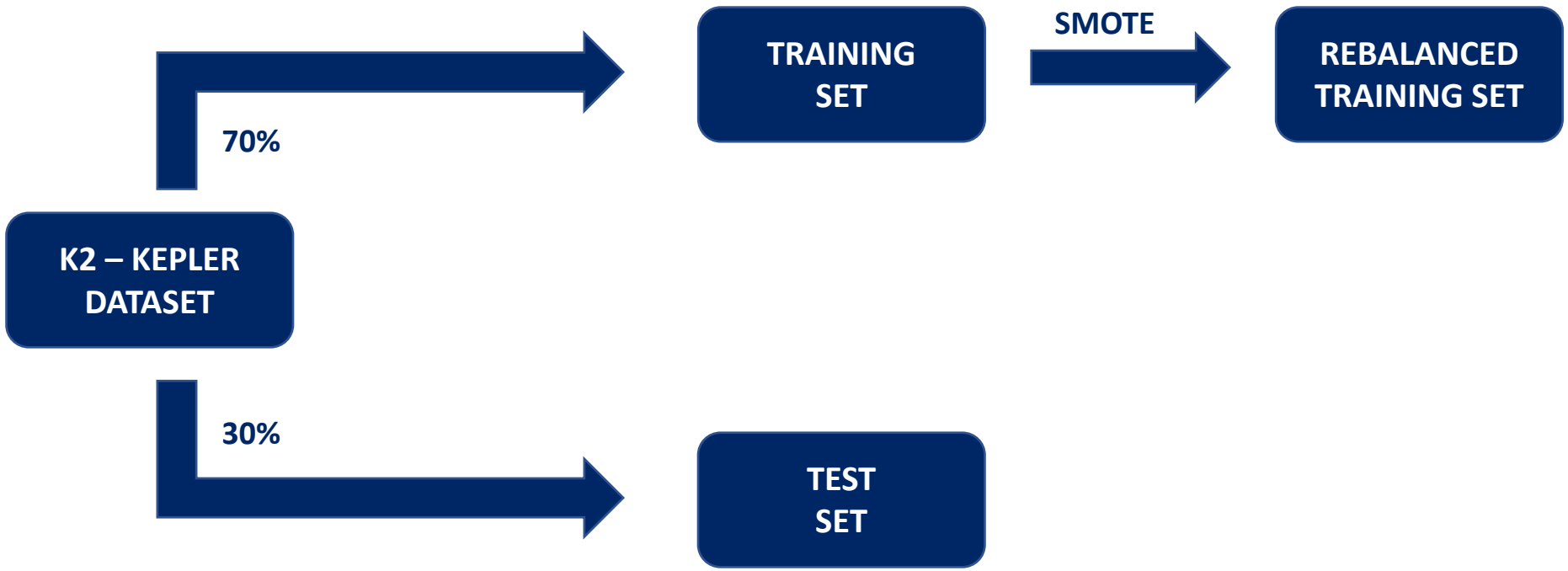
K Neighbors	F1-Score	Std
3	0.910	0.059
7	0.909	0.064
15	0.912	0.049
31	0.913	0.057
63	0.903	0.053
85	0.916	0.051

K-NN FILLING

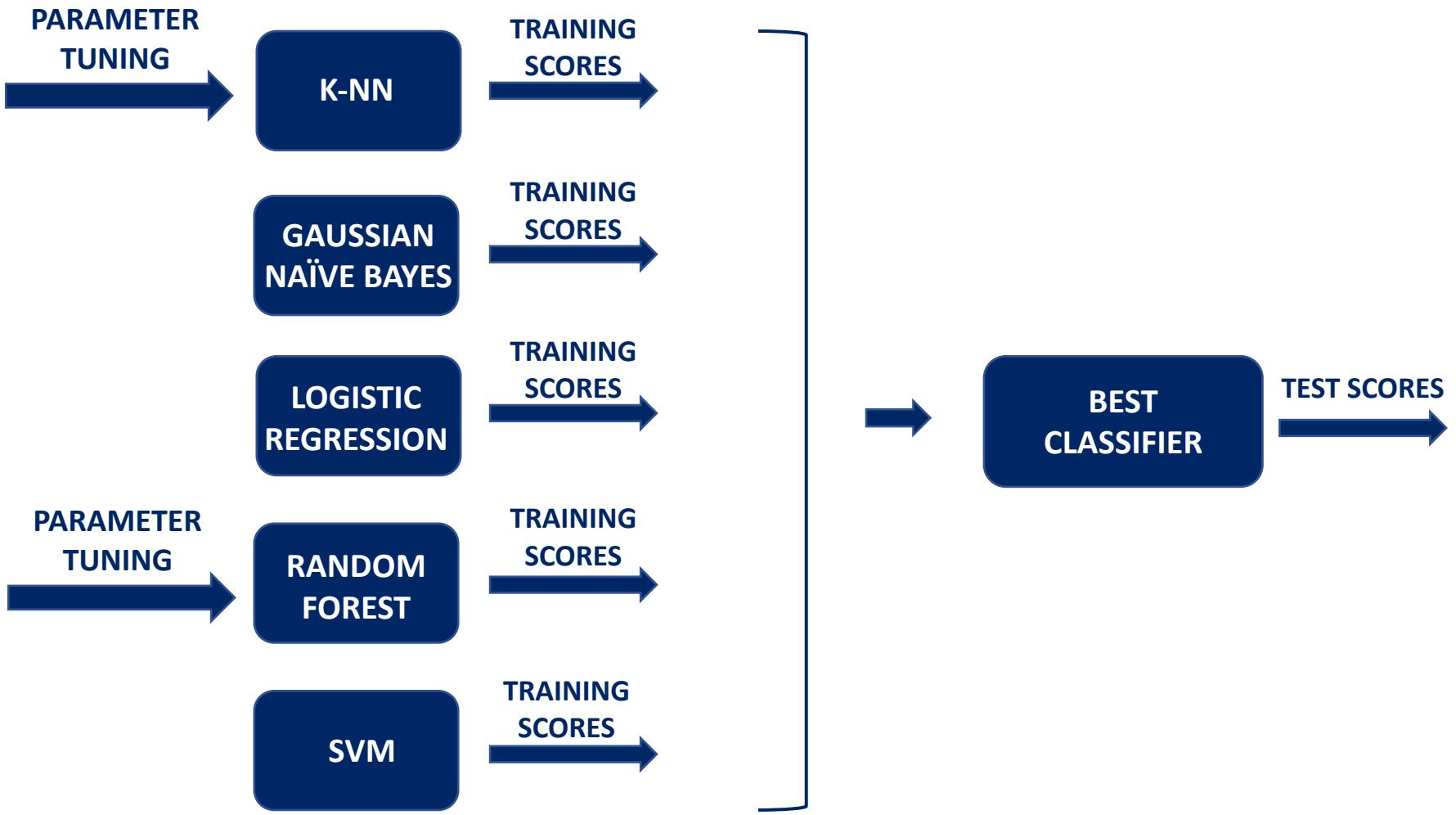
K2 – KEPLER FILLED DATASET

Classification

Training and Test splitting



Learning Phase

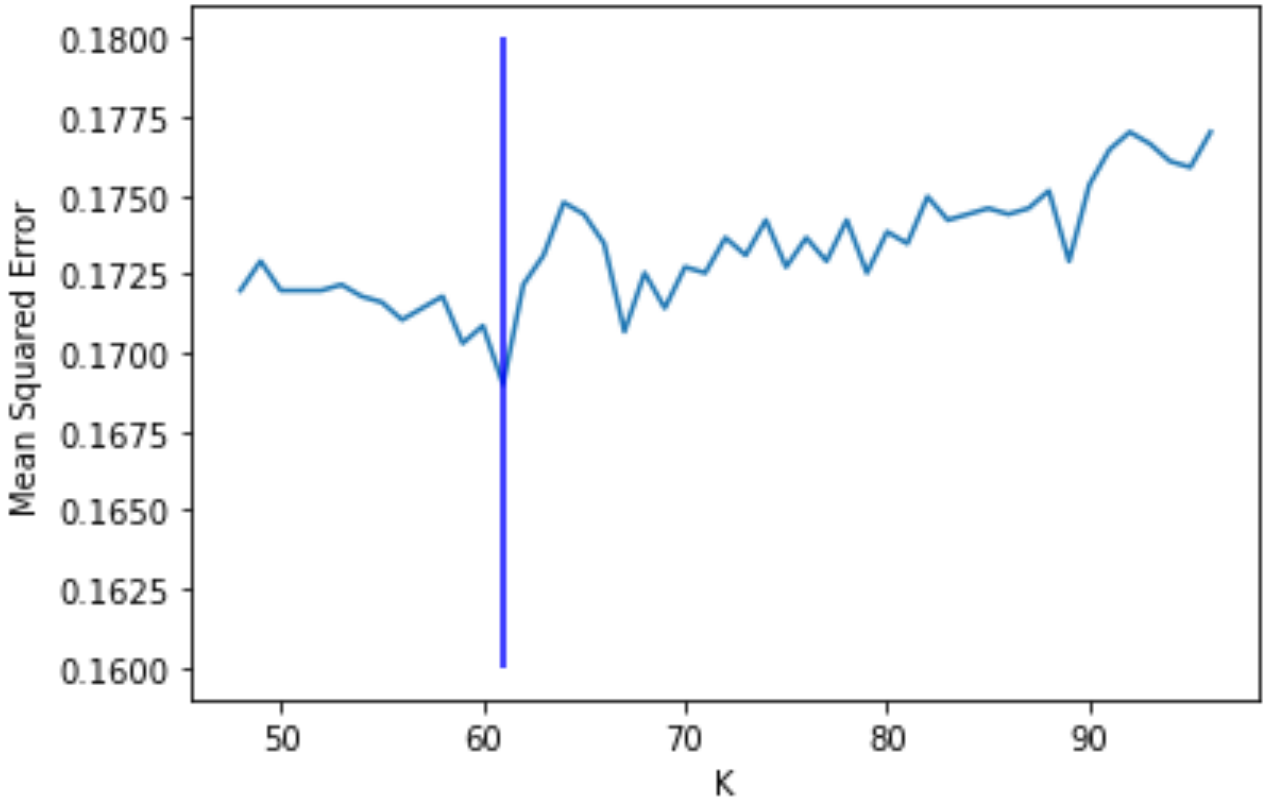


K-NN Parameter Tuning

K-NN is a very simple algorithm but it's important to choose a good value for *k.*

*W*e started with a value of k equal to the square root of the number of samples as done for handling missing values.

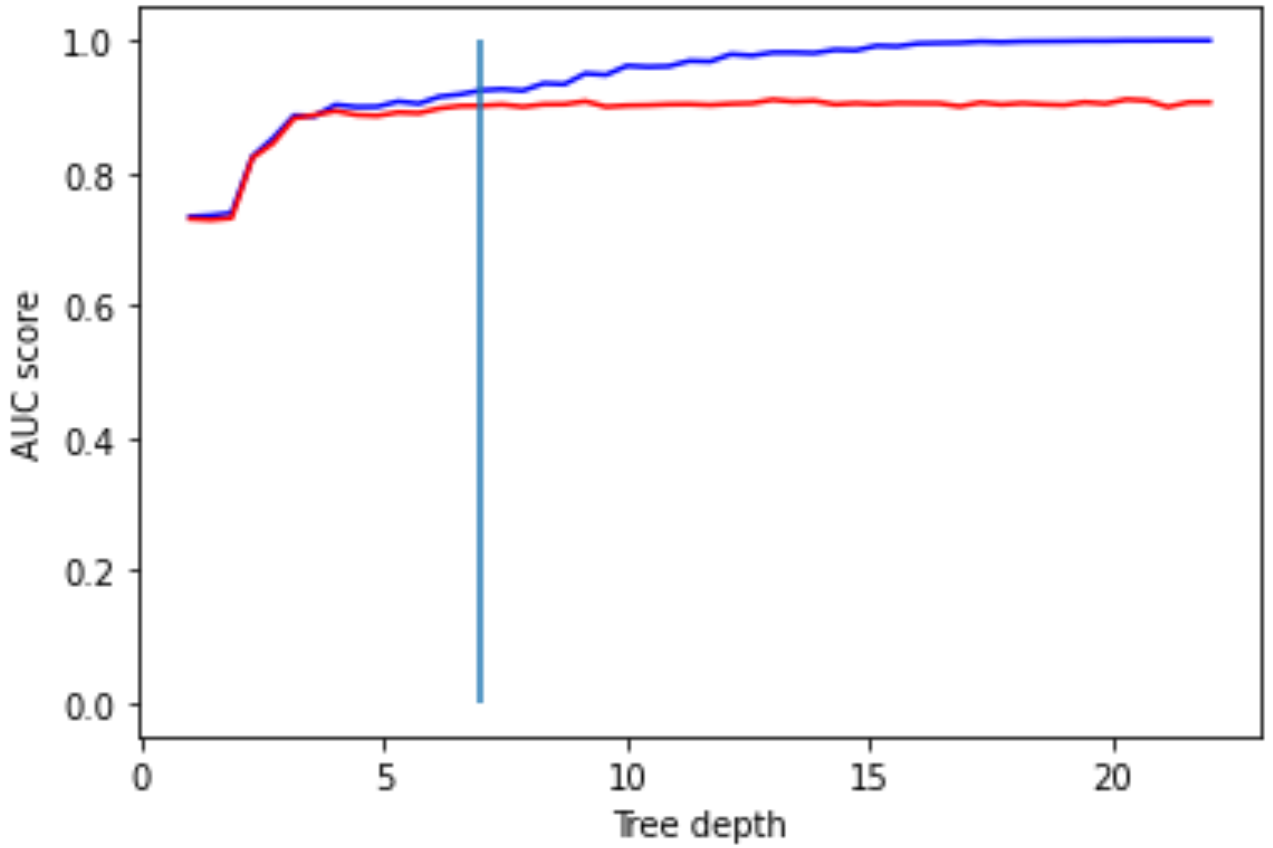
We tested k values belonging to [sqrt - sqrt/3, sqrt + sqrt/3] and we choose the k that gives the lowest error.



Random Forest Parameter Tuning

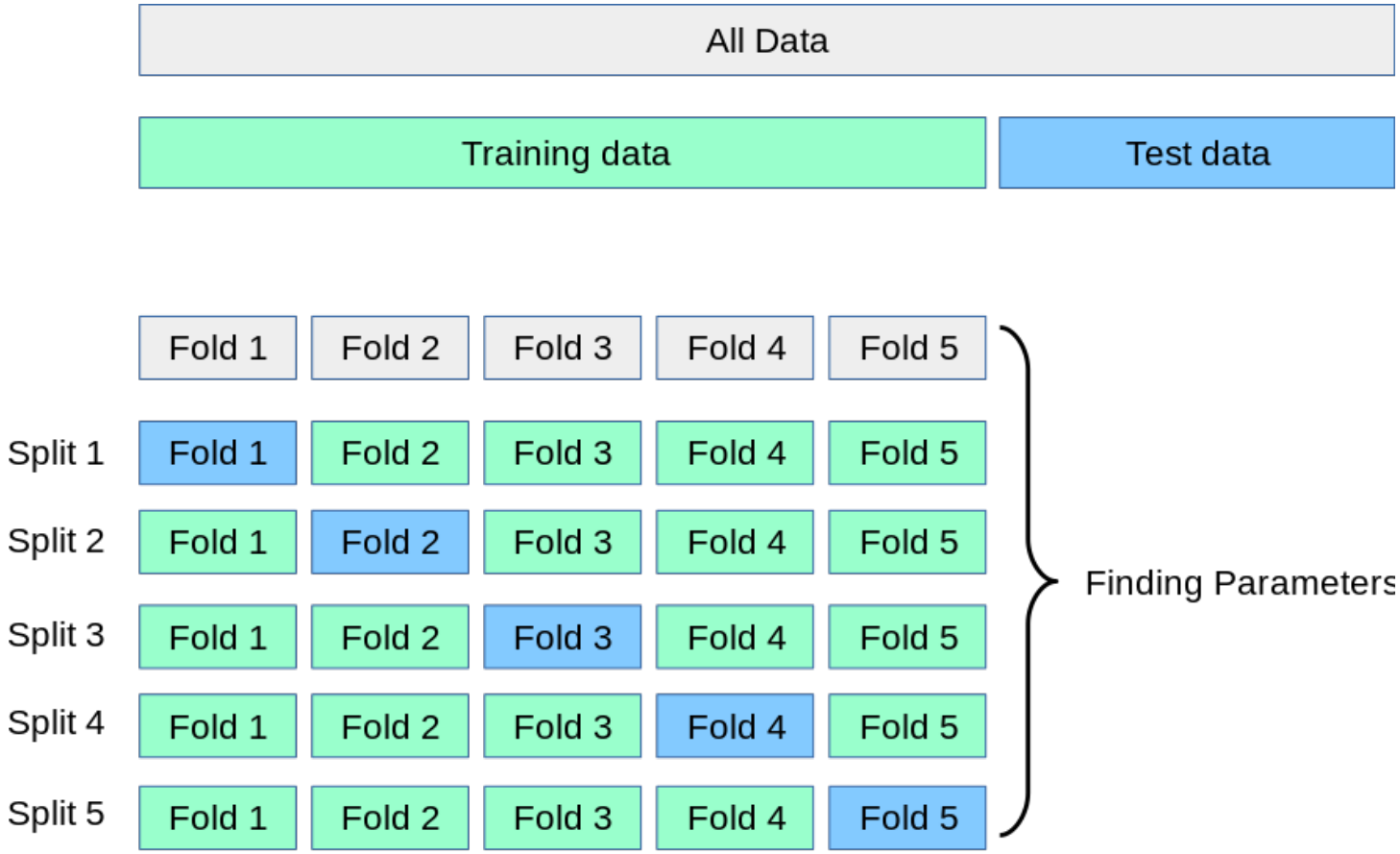
Important attributes for Random Forest:

- Number of trees = 40 ((Sturrock et al., 2019)
- Maximum depth: we chose the maximum depth such that the difference between the AUC value of training and testing was at most 0.3



Cross Validation

All the classifiers were evaluated with 10-fold cross validation so we obtain, for each classifier, 10 different scores for each metric.



Performance Evaluation

Mean scores of 10-fold values for each metric.

In the next slide we can see the F1-score value for each fold

	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

Performance Evaluation

	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

Being the original dataset unbalanced we choose the best classifier using F1-score as metric

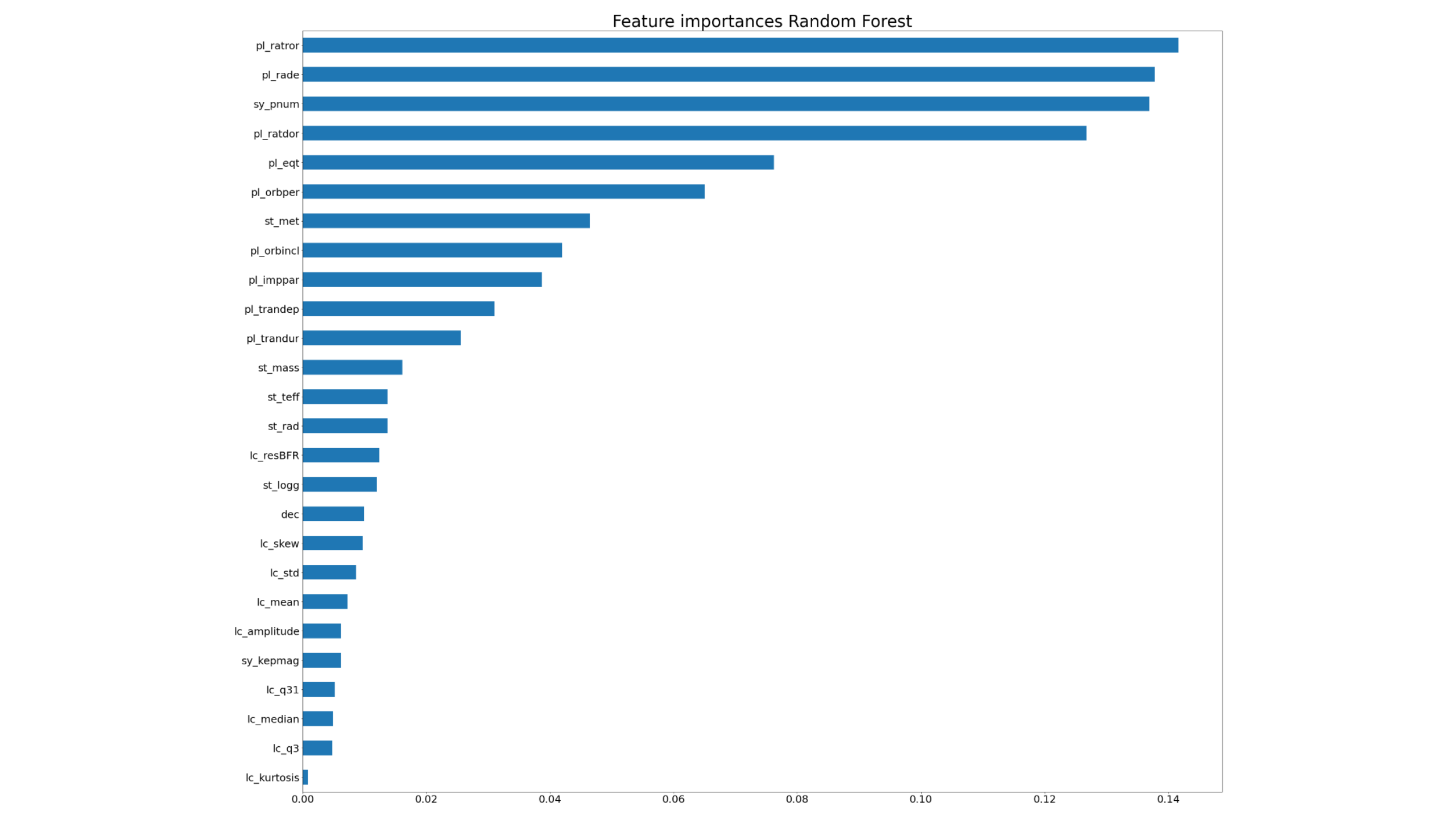
In the table we can see that Random Forest is the best classifier for our purpose

Random Forest Trials

Comparison of training and test scores in different cases.

$\Longrightarrow$ The model is not overfitted and the best performaces are obtained using both planet metadata and lightcurves features.

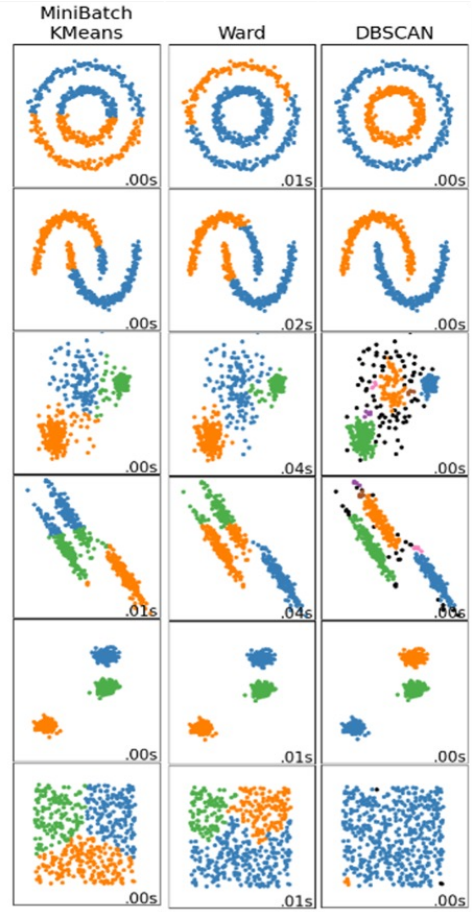
	F1-Score (Training Set)	F1-Score (Test Set)
Full Dimensionality (Planet Metadata + Light curves features)	0.931	0.90804
PCA on Planet Metadata + Light curves features	0.899	0.86645
Planet Metadata	0.837	0.9011
PCA on Planet Metadata	0.901	0.8853


Feature importances Random Forest
pl_ratror
pl_rade
sy_pnum
pl_ratdor
pl_eqt
pl_orbper
st_met
pl_orbincl
pl_imppar
pl_trandep
pl_trandur
st_mass
st_teff
st_rad
lc_resBFR
st_logg
dec
lc_skew
lc_std
lc_mean
lc_amplitude
sy_kepmag
lc_q31
lc_median
lc_q3
lc_kurtosis
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14

Clustering

Pre-Analysis

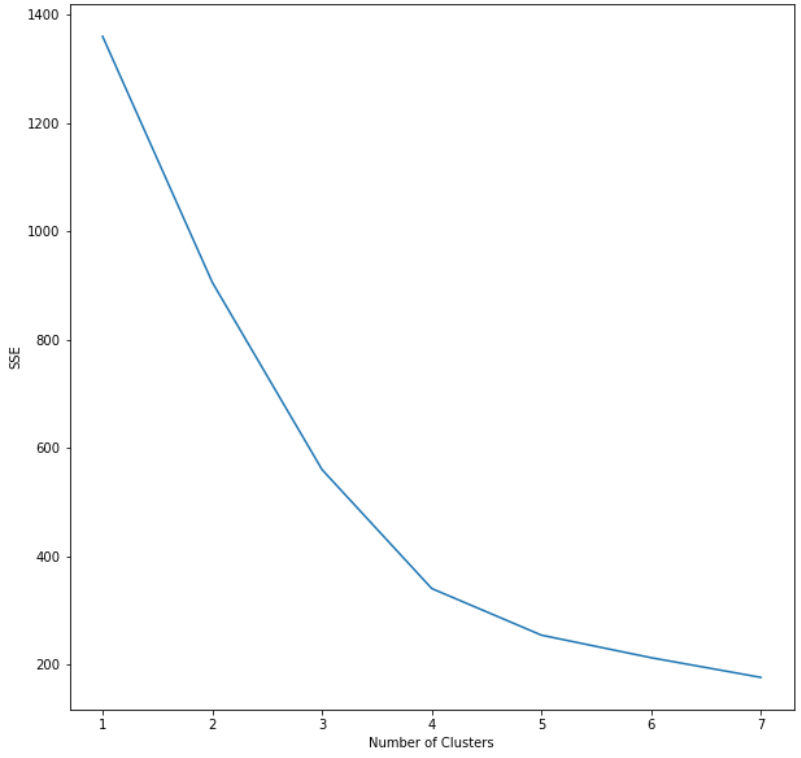
The clustering analysis started with determining whether the dataset could contain meaningful clusters through the Hopkins test.

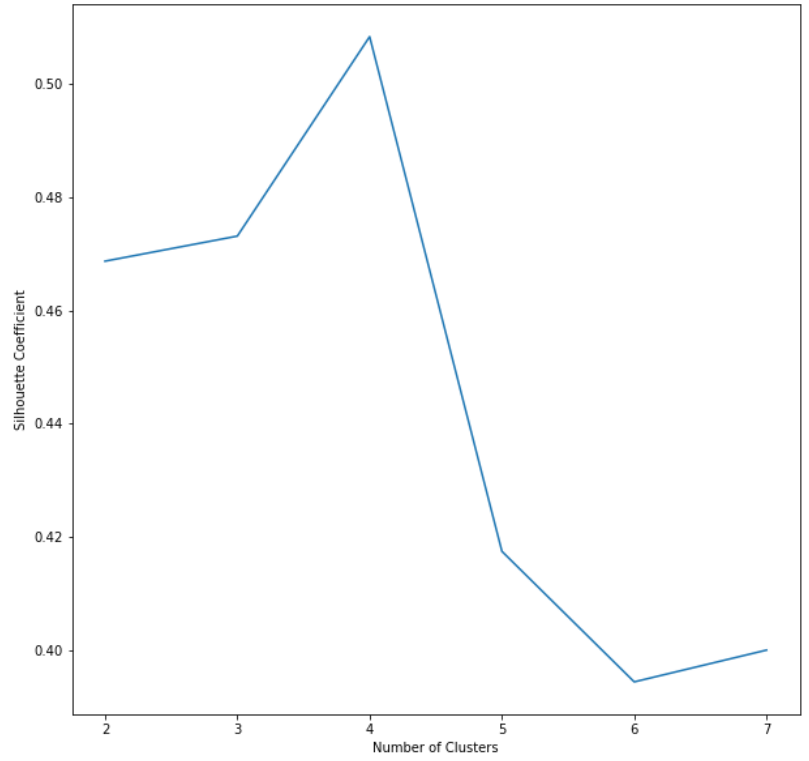



```
['pl_rade ', 'pl_bmasse ', 'pl_eqt ', 'pl_orbper ', 'st_mass ', 'st_rad ', 'st_met ']
0.9881837530203653

['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
```

K-Means (I)

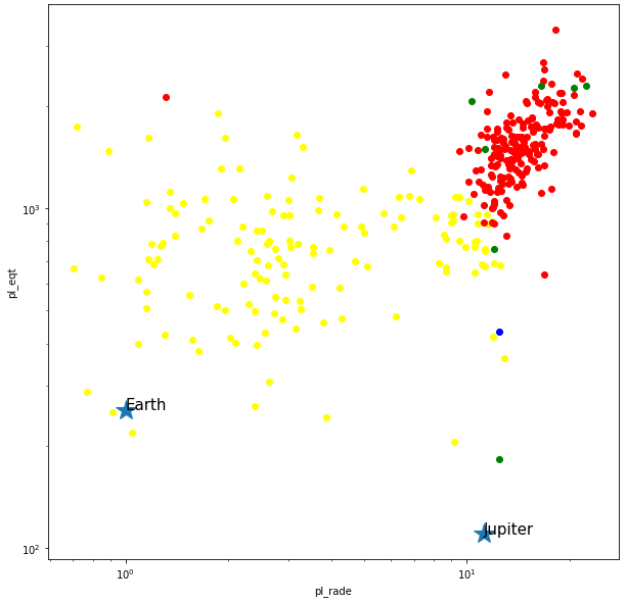
The first step was to find the number of clusters to consider in order to have a good analysis.
Both elbow method and silhouette score were used so to compare them and have multiple sources from which choose the optimal number of clusters.

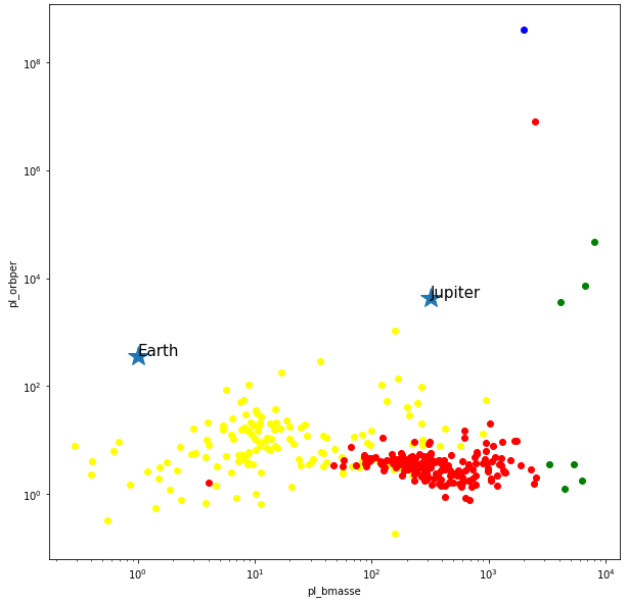


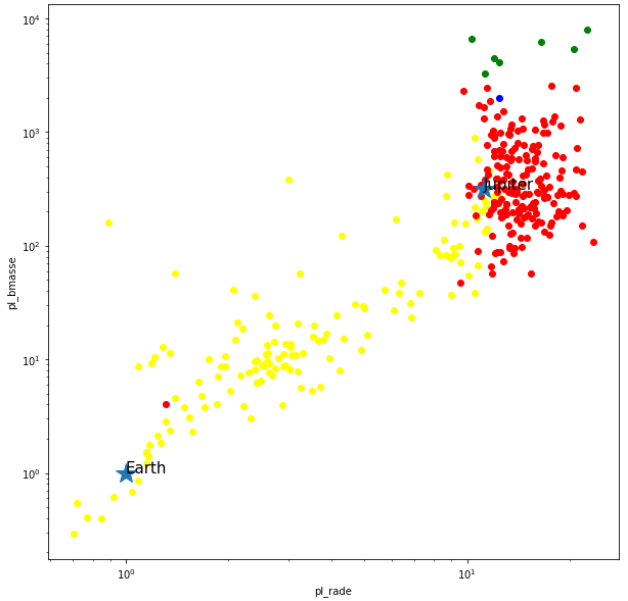


K-Means (II)

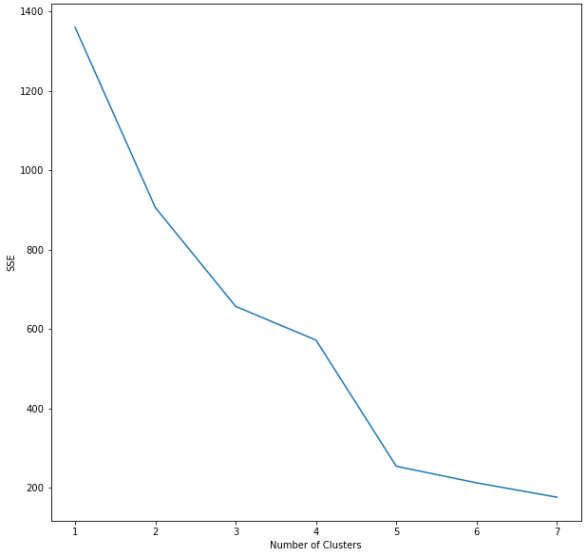
Cluster	Most important attributes	
Yellow	**pl_rade**	**pl_eqt**
	0.9956512124212715	0.8505620077170917
Red	**pl_rade**	**pl_eqt**
	0.7054046549603565	0.6103352701157794
Green	**pl_orbper**	**pl_bmasse**
	18.462492828330653	1.8238282403683042
Blue	**pl_bmasse**	**pl_rade**
	5.796564321588109	0.7988947409228376





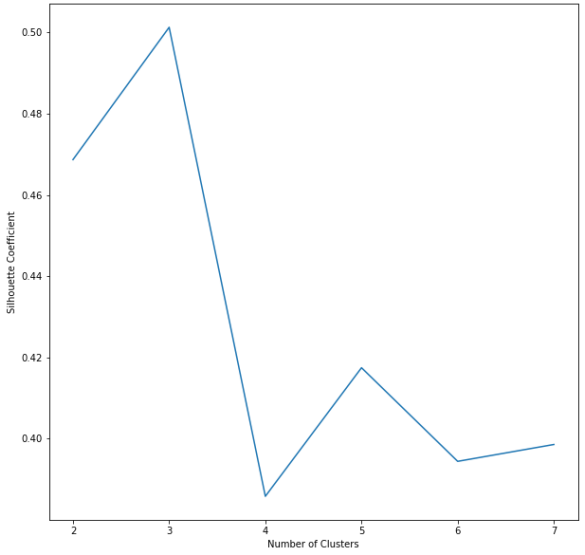


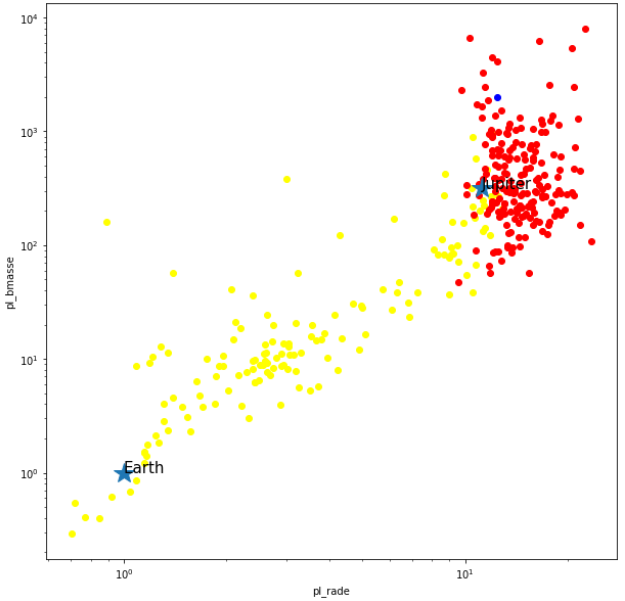
K-Means (III)



Random «Elbow»

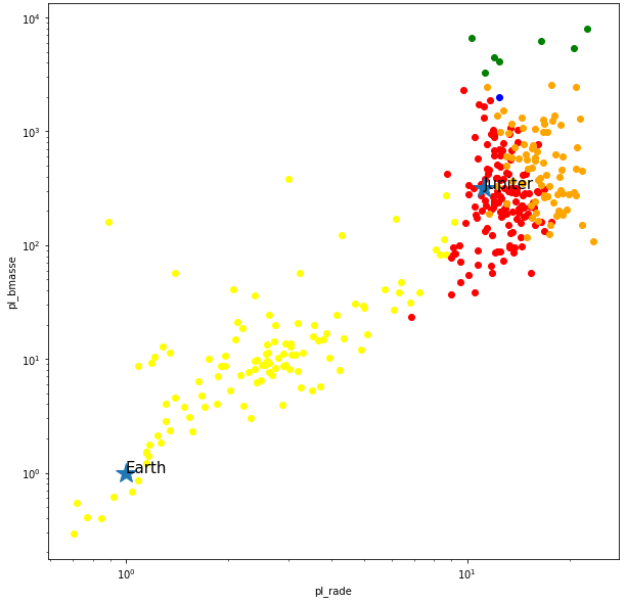
Random «Silhouette





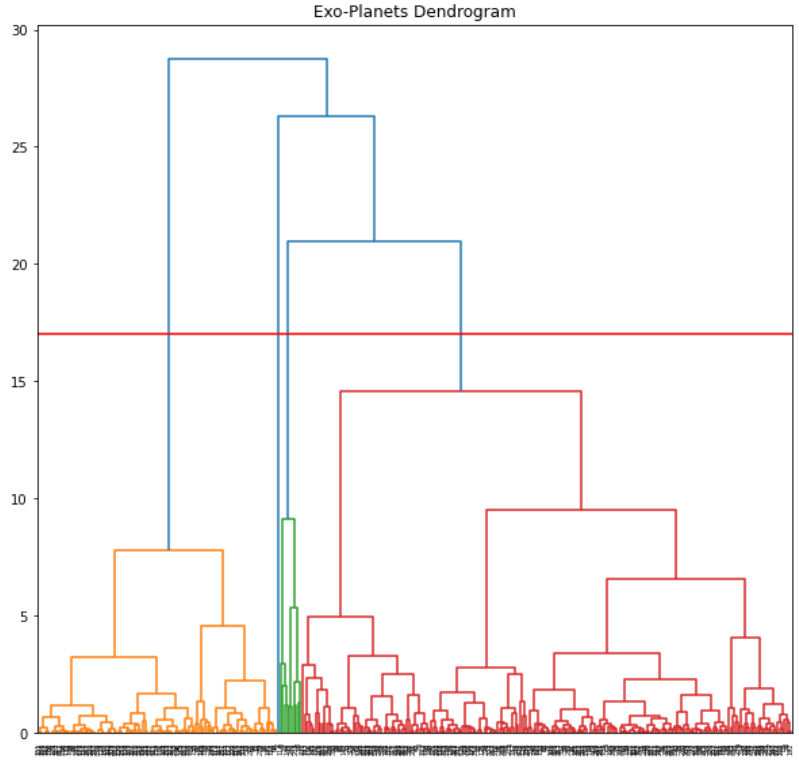
3 Clusters

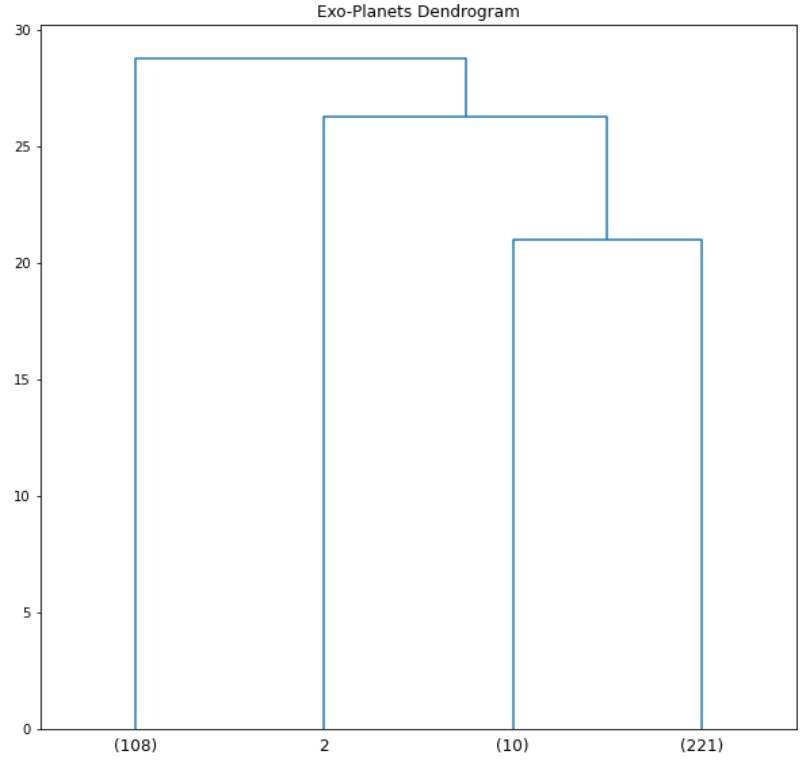
5 Clusters



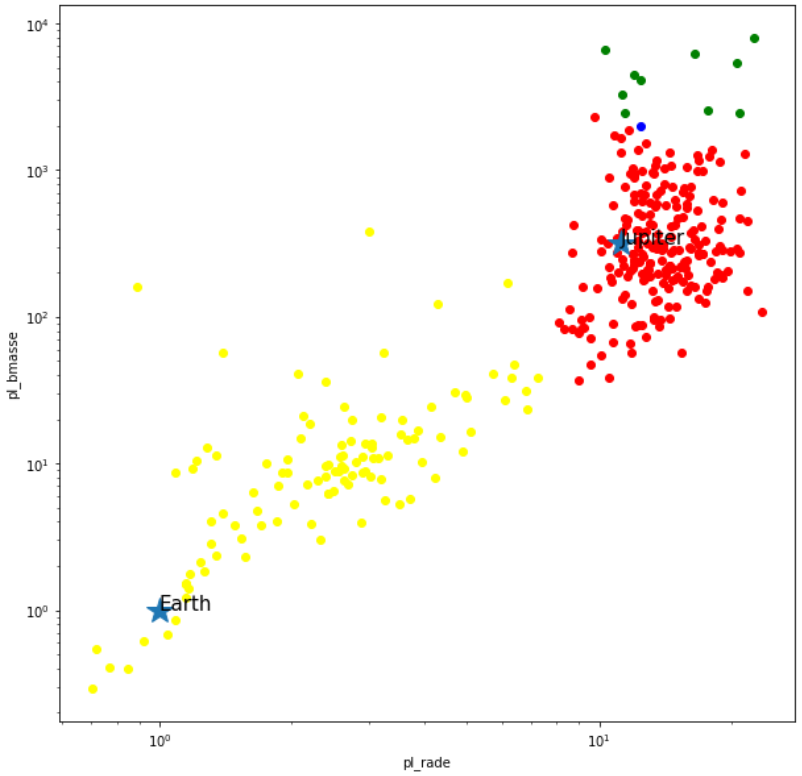
AGNES (I)

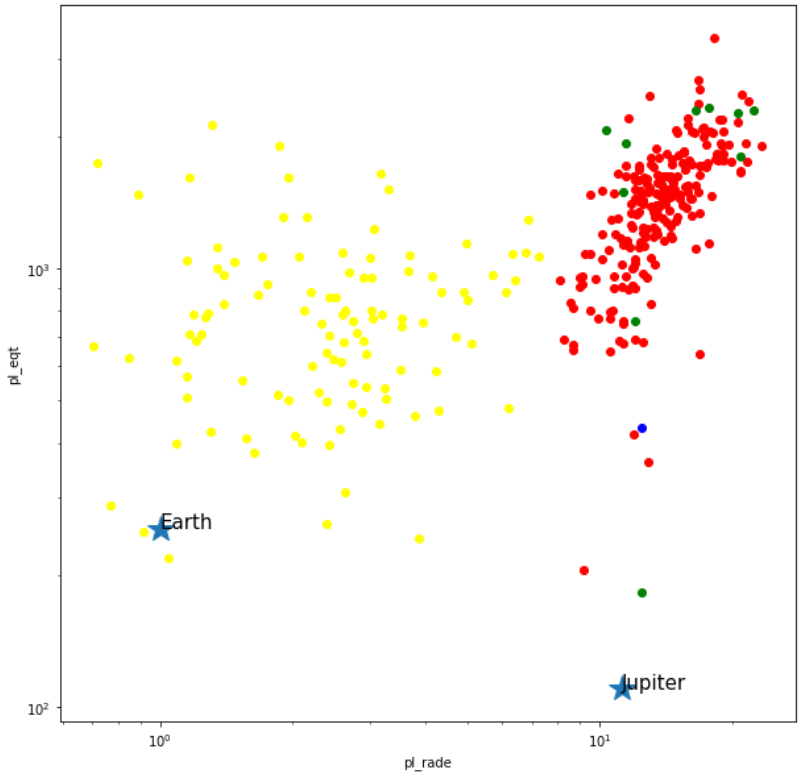
The second approach was to use a hierarchical clustering algorithm, specifically AGNES. The first step was to build the dendrogram representing the clusters. The dendrogram was built by considering the Euclidean distance between points and using the “ward” method, in order to minimise the total within-cluster variance.

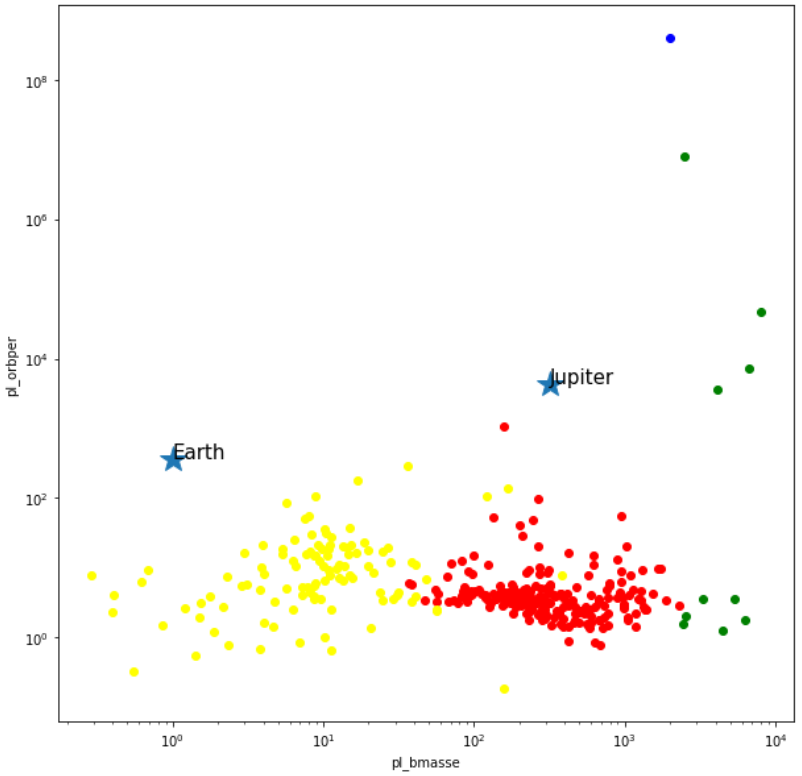




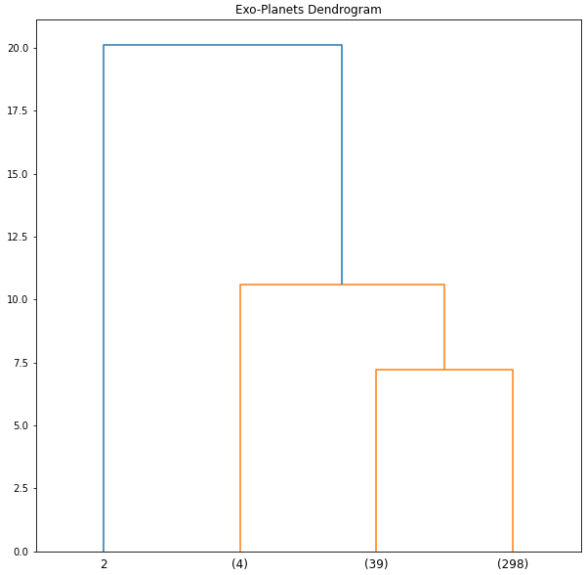
AGNES (II)


pl_bmasse
pl_rade
Earth
Jupiter


pl_eqt
pl_rade
Earth
Jupiter

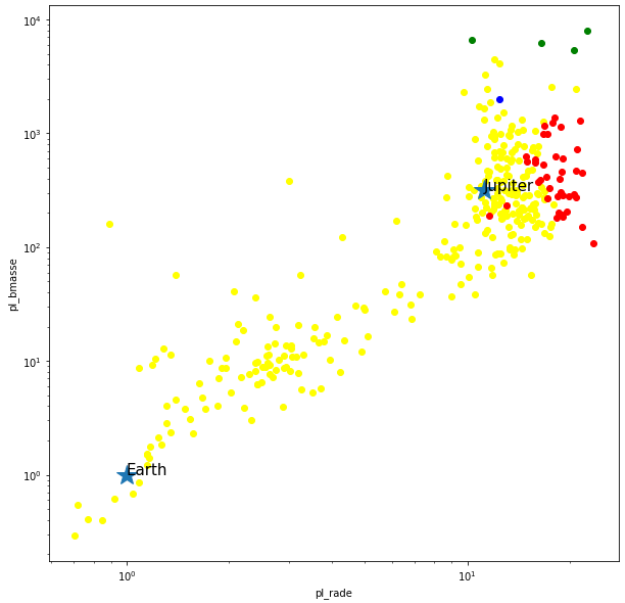

pl_orbper
pl_bmasse
Earth
Jupiter

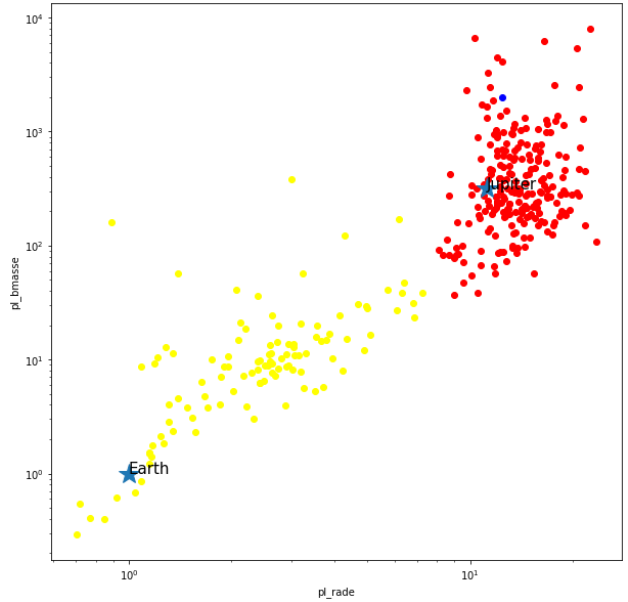
AGNES (III)



«Complete» Dendrogram

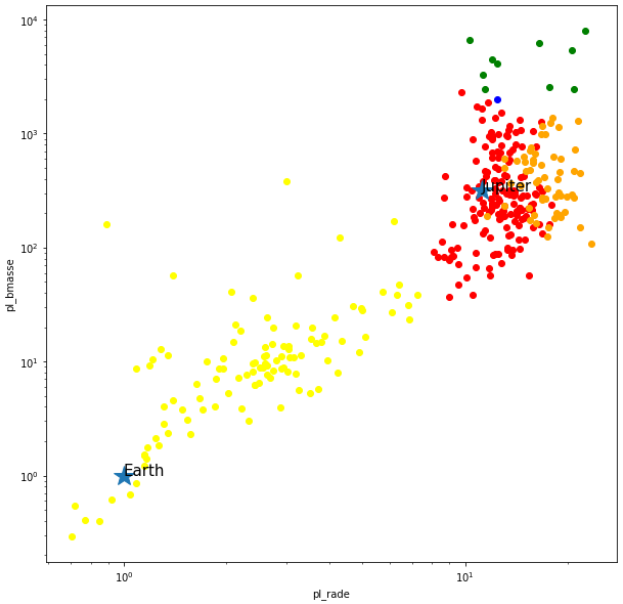
«Complete» linkage results





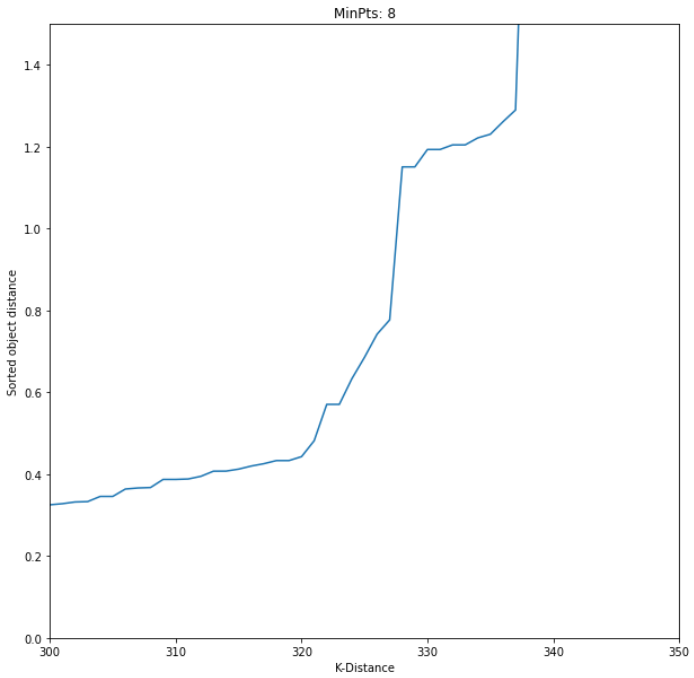
3 Cluster
«Ward»

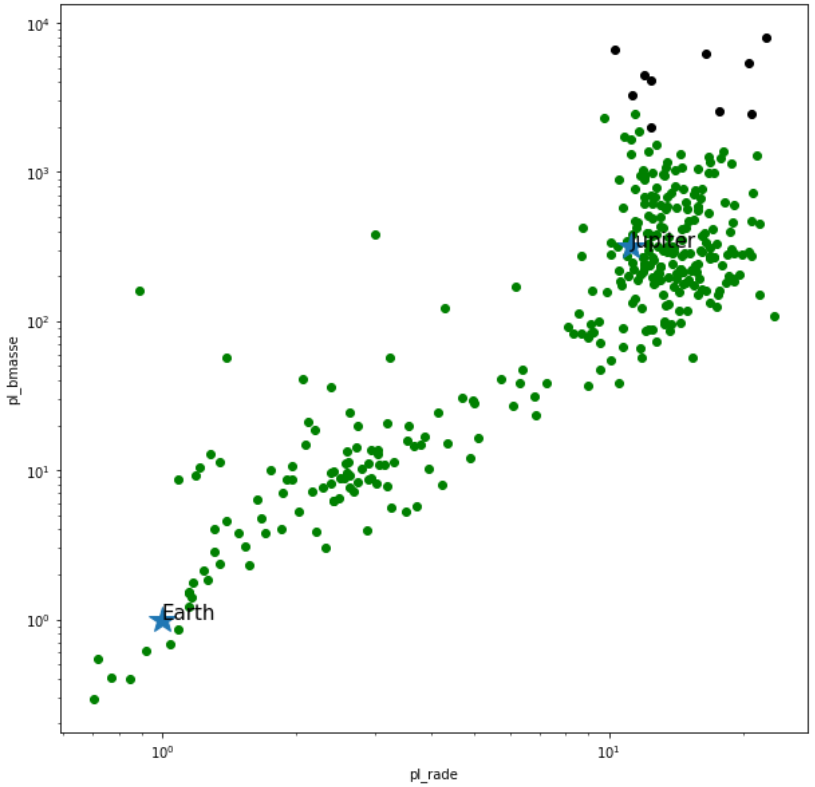
5 Cluster
«Ward»



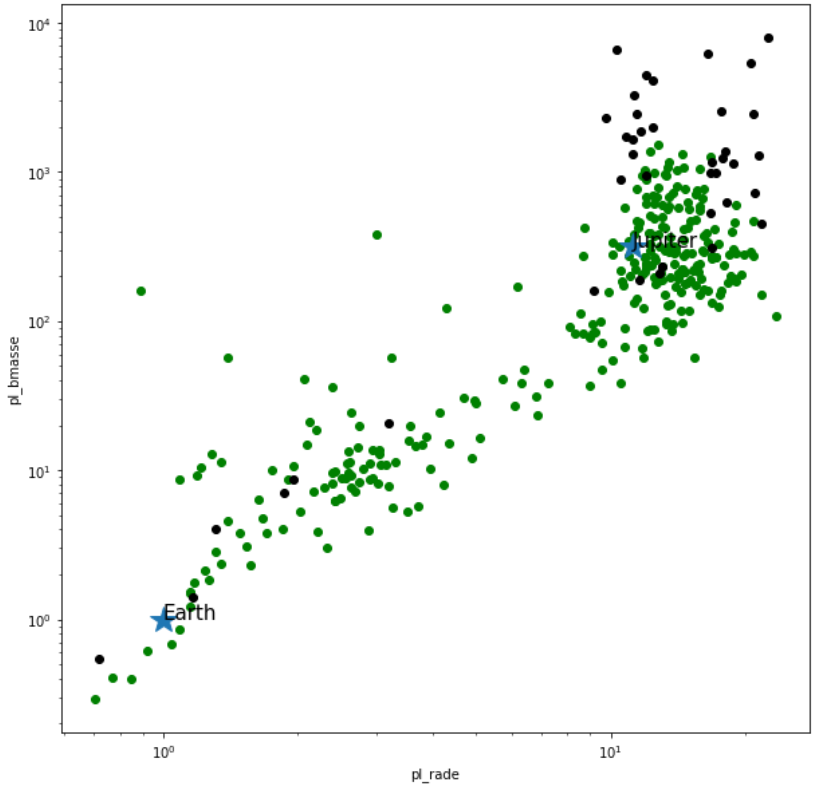
DBSCAN

The optimal epsilon is [0.5, 1.3]. 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.





Eps = 1.3



Eps = 0.5

Conclusions

Conclusions

The Random Forest classifier was chosen as the primary model for testing and deployment.
Objects-of-interest with a radius between Mars and Neptune and with a transit light curves like other confirmed planets, and other planets in their host system are most likely to be classified as exoplanets.

Thanks to K-Means, but above all to 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.