Kepler Mission: Celestial Bodies Classification

Alberto Monaco s1043826 Edoardo Gervasoni s1043824

Data Mining Course, Radboud University, Nijmegen, January 2020

Abstract. In Astronomy, the study of celestial bodies is based on the data that scientists and researchers receive from satellites and space telescopes in orbit. The observation of the electromagnetic spectrum provides fundamental information about the emitting source, which can belong to galaxies even far away from the Milky Way. Nowadays, the volume of the data collected by space probes and orbital telescopes has become too large to be treated manually by individuals. Therefore, the use of Data mining algorithms and Machine Learning techniques is needed to efficiently carry out Big Data analysis in Astrophysics, e.g. in exoplanets hunting (finding planets outside the solar system). In this report we are going to process data from the NASA Kepler mission, in order to correctly classify observed objects by Kepler orbital telescope as Exoplanets or not. We will describe all the steps carried out throughout the analysis, from data pre-processing and exploration to the modeling techniques applied.

Keywords: Kepler — Exoplanets — Classification — Predictive modelling

1 Introduction

1.1 Application Domain and research problem

The mission of *Kepler* space telescope, launched by NASA in 2009 and operating until 2014, was to collect essential data for the search of exoplanets orbiting around stars in several sectors of the Milky Way[1]. Historically, the first officially recognized exoplanet dates back to 1992 and since then more than 3900 planets have been confirmed. Currently, several detection methods exist, but the one which led to the greatest number of discoveries (77.5 % of confirmations) is the observation of the transit of the planet in front of its star. The Kepler mission demonstrated the effectiveness of this photometric approach by monitoring the brightness of thousands of stars at different times: if the telescope had detected a periodic reduction of the star's brightness, this event could have indicated the transit of a planet, as its passage caused a reduction in the light curve of the star. All the project has been carried out by exploiting the software R, full of helpful libraries to analyse, explore and process the data.

2 Related previous work

For the development of our project, we consulted some sources concerning machine learning techniques applied in exoplanets hunting. In particular two papers helped

[1] https://exoplanets.nasa.gov/the-search-for-life/exoplanets-101/

us in our research, *Automatic classification of kepler planetary transit candidates*[2], Sean D. McCauliff, Jon M. Jenkins, and *Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90*[3], Christopher J. Shallue, Andrew Vanderburg. The first one is important since it describes the application of data-mining and machine-learning techniques to the classification of the exoplanet transit-like signals present in the Kepler light curve data. The use of a random forest model in this paper, suggested us the idea of adopting the same algorithm - among others - for the purposes of our project.
In the second one, a technique for detecting confirmed exoplanets using a deep Convolutional Neural Network is presented.

3 Data overview

The aim of our project is to train an algorithm that is capable to classify a celestial object as an exoplanet or not. For this purpose, it has been chosen the *Kepler Exoplanet Search Results* dataset, which contains the official data from NASA's Kepler mission[4]. The dataset consists of 9564 detected cases, called Kepler Objects of Interest (KOI), which include confirmed exoplanets, false positives or candidate but not confirmed objects.

3.1 Description

For each observation have been collected a total of 49 variables. The first three - kepid, kepoi_name and kepler_name - are identifying attributes of the KOI, which provide an identity number for each belonging star and the name of the various potential exoplanets detected. The variable koi_disposition, on the other hand, is a nominal categorical variable that has the values CONFIRMED, FALSE POSITIVE and CANDIDATE, indicating whether an observed object is a confirmed extrasolar planet, if it is a false positive or if it is candidate. If an exoplanet is confirmed, it is added by researchers to the *Exoplanet Archive Confirmed Planet Table*[5], which keeps track of confirmed discoveries. FALSE POSITIVE and CANDIDATE values, are instead designations taken from "Disposition Using Kepler Data" and are also used for the koi_pdisposition variable. This is another categorical attribute, which describes the most likely physical explanation of a KOI using data analysis techniques.
Koi_score variable gives us a level of confidence about the KOI's disposition and its values range from 0 to 1, where 0 represents a low confidence that the object is an exoplanet, while 1 represents a high confidence that it is. Koi_fpflag_nt, koi_fpflag_ss, koi_fpflag_co, koi_fpflag_ec, moreover, are binary attributes [0,1] that explain false positive cases, i.e. when the star's brightness decrease can have been caused by other phenomena, such as by the transit of another star, in the case of a binary star system, or by the interference of nearby stars. Other relevant attributes are those related to the characteristics of the transit observed by Kepler, such as the interval between the

[2] https://iopscience.iop.org/article/10.1088/0004-637X/806/1/6/meta
[3] https://iopscience.iop.org/article/10.3847/1538-3881/aa9e09/meta
[4] https://www.kaggle.com/nasa/kepler-exoplanet-search-results
[5] https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html

transits of the planet (koi_period), its duration (koi_duration), or the planetary radius of the KOI (koi_prad) and its surface temperature (koi_teq). The properties of the star are represented by other attributes: koi_steff describes the temperature of the star's photosphere in Kelvin, koi_slogg the gravity on its surface and koi_srad its radius.

3.2 Preprocessing

During the preprocessing phase, the data are prepared and organized in order to train more easily the algorithms. First of all, a **feature selection** of the initial 49 variables was made: columns considered not relevant for our work were removed, including all the columns of errors, as well as the nominal categorical attributes kepler_name, koi_tce_plnt.nmbr and koi_devilname, thus obtaining a dataset consisting of 22 columns.
Secondly, the problem of missing values had to be addressed. We decided to remove about 300 common missing observations for almost all the variables, and then to focus on the koi_score attribute, which has 1510 missing values. Since we are not domain experts and since we didn't want to arbitrarily manipulate and alter the data, we chose to remove these observations, obtaining a dataset consisting of 7944 rows.
As a last preprocessing step, we decided to create our class variable from koi_disposition, performing a **binarization** of it: we considered as positive class (value 1) all the objects confirmed, and as negative class (value 0) all the false positives and candidates.

4 Modeling

4.1 Models description and approach

Once performed all the techniques mentioned and explained above, we have obtained the data suitable for modeling tasks. Given the structure and the patterns showed, the dataset needed a supervised approach in order to extract worthy insights. As a matter of fact, we already had the information and examples input-output pairs through which the researchers were able to classify a celestial body as Exoplanet or not. Our purpose, thus, lies on analysing the data patterns and features and then, based on these examples, producing new values for the target variables of interest. One of the main appealing aspect of the whole project, as often happens in the supervised context, is the discovery and understanding of the most important attributes that are helpful to characterize the choice regarding the dependent variable.
In this respect, we performed several algorithms:

- Logistic Regression
- Step-wise Logistic Regression
- Decision Tree
- Random Forest
- Single-hidden-layer neural network

However, before any model application, we have chosen to apply one more adjustment. Th target variable, in fact, presented roughly only 28% of positive values (equal to 1). Thus, we have considered appropriate the use of a technique known as **Oversampling** to balance the class labels, which basically consists of random picking and generating observations from the minority examples. The method has been carried out by using the function *ovun.sample* of the *ROSE* library. It has to be considered that, for the classification task, we decided to split the data into two different sets: training and test. The first one, equal to the 67% of the records, has been used to training the models to learn the structure of the data and build up an inducer. Subsequently, the remaining set, has been exploited as benchmark to compare and evaluate the predictions achieved by the algorithms.

4.2 Logistic Regression

The first attempt has been realized by a simple Logistic Regression. The idea was to run the algorithm and then set the results from it achieved as baseline for all the next models. This approach tends to consider a following model as acceptable only if it can overcome the outcomes returned by the simplest model.

Once performed the Logistic Regression with all the attributes available, except the target, set as explanatory variables, we decided to run the Logistic Regression with a Step-wise approach. This method, in particular the one with the backward logic, starts from the previous Regression formula and tries to remove some feature which can be considered negligible for classifying the target. It runs many combinations of the model, omitting, step by step, some variables. At the end, all the models built are compared with the **AIC** criterion, in order to establish the best one. The Akaike information criterion is an estimator of the quality of the model under consideration, taking into account the trade-off between the goodness of fit of the model and the simplicity of the same.

However, despite the large number of attributes, all the features were considered useful by this approach. Thus, the model expressed by the Step-wise coincides with the Logistic Regression we have talked about.

4.3 Decision Tree

So, fixed the baseline, we switched over the Decision Tree. It was carried out for two different purposes:

- Feature selection
- Classification

As a matter of fact, the *caret* library allows us to evaluate the importance of the variables that lead the model to learn the right patterns of the data and perform correct predictions.

From the figure 1 it is possible to look at the score achieved by any feature. As we can see, four of them were considered as not very useful for our purpose, and then not taken into account for the next algorithms.

The second aim, conversely, consisted of applying the learned model to forecast new examples, i.e. the regular task required to a classification model.

Both the goals has been completed by means of **parameters' tuning** and a **Three**

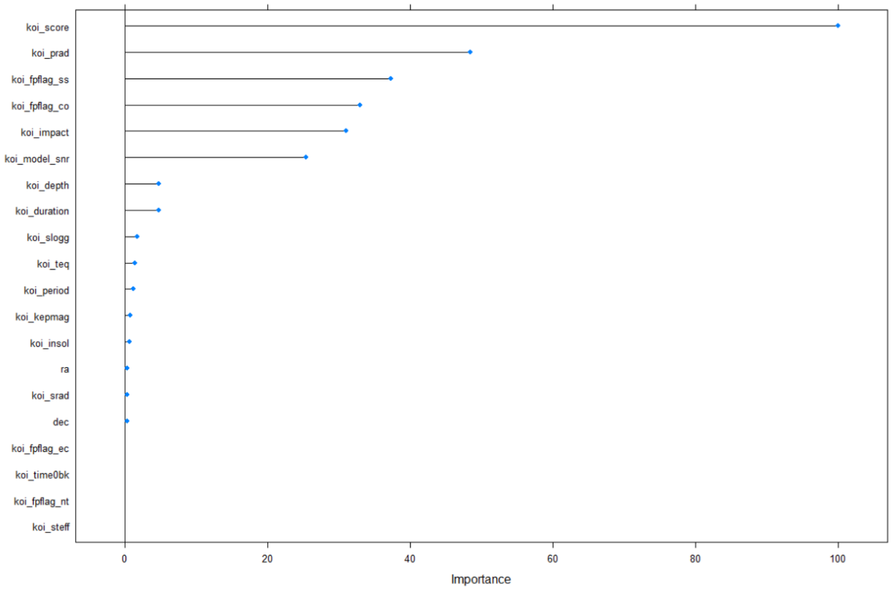

Fig. 1. Variable importance according to the Decision Tree

Fold Cross Validation to reduce the model's bias. This last method, in fact, divides the data into k subsets and then repeats the holdout method k times, such that each time, one of the k subsets is used as the test set and the other k-1 subsets are put together to form a training set.

4.4 Random Forest

As done for the Decision Tree, we have tuned the parameters of the Random Forest thanks to the opportunity given by *caret* to customize the algorithm. In particular *mtry* and *ntree*, after the tuning, were fixed equal to 4 and 500, respectively. The model has been trained with a Three Fold Cross Validation, the same technique mentioned before.

4.5 Single-hidden-layer neural network

At the end, despite the great results obtained so far, we believed that our work would have been complete only after the implementation of a Neural Network. In particular, we exploited a Single hidden layer Neural Network, through the use of the *nnet* method contemplated in the caret package. For this purpose, we decided to perform some preprocessing techniques, PCA, Normalization and Standardization. Among these we kept the last one, since it was able to return the best value in terms of ROC curve. Here, again, we applied the Three Fold Cross Validation.

5 Results

To jump into the Results section it is necessary to define the **Confusion Matrix** and all the measures associated to.

5.1 Confusion Matrix

	Actual: No	Actual: Yes
Predicted: No	TN	FN
Predicted: Yes	FP	TP

This is a table often used to describe and compare the performance of classification models on a set of test data. It contains predictions of such a model, together with the actual values contained in the data. By crossing this information, it is possible to define some indicators:

- TP: The predicted value is positive and it is true.
- TN: The predicted value is negative and it is true.
- FP: The predicted value is positive but it is false.
- FN: The predicted value is negative but it is false.

From these values it is possible to infer some useful measures. In particular:

- Accuracy:
$$A = \frac{TP + TN}{TP + TN + FP + FN}$$
- Precision:
$$P = \frac{TP}{TP + FP}$$
- Recall:
$$R = \frac{TP}{TP + FN}$$
- F Measure:
$$F1 = \frac{2P \cdot R}{P + R}$$

While Accuracy can be seen as the fraction of records rightly classified, the Precision and Recall have a specific meaning. In a classification task, the precision is the fraction of positives records rightly classified among all the observations labeled as positive from the model. Conversely, Recall is defined as the fraction of positives records rightly classified among the total number of observations that actually belong to the positive class.

5.2 Models Results

Once defined the measures able to evaluate the models, we can now show the results:

1. Logistic Regression

	Actual: No	Actual: Yes
Predicted: No	1404	74
Predicted: Yes	449	1824

- Accuracy: 87.62%
- Precision: 94.99%
- Recall: 75.77%
- F1 Measure: 84.30%

2. Decision Tree

	Actual: No	Actual: Yes
Predicted: No	1621	136
Predicted: Yes	232	1762

- Accuracy: 90.19%
- Precision: 92.23%
- Recall: 87.48%
- F1 Measure: 89.81%

3. Random Forest

	Actual: No	Actual: Yes
Predicted: No	1678	48
Predicted: Yes	207	1818

- Accuracy: 93.20%
- Precision: 97.22%
- Recall: 89.02%
- F1 Measure: 92.94%

4. Nnet

	Actual: No	Actual: Yes
Predicted: No	1388	95
Predicted: Yes	497	1771

- Accuracy: 84.21%
- Precision: 93.59%
- Recall: 73.63%
- F1 Measure: 82.42%

6 Conclusions

The goal of this project was to identify the best classifier in order to define a celestial body as an Exoplanet or not, based on the NASA's Mission Kepler data. The data originally available contained several problems that we tried to solve, despite the lack of domain experts. In particular, we decided to delete all the missing values of the *koiscore* variables, since all the common missing replacement approaches seemed to be inappropriate. Furthermore, the dependent variables was divided into three different categories. We decided, in this regard, to keep the confirmed Exoplanets as the positive class and assemble as not confirmed the remaining categories. Moreover, we found two more critical points, one concerning the numerous set of variables, and the second regarding the classes imbalance for the target variable. Throughout the analysis, we have chosen to lighten the computational burden for the model, avoiding, in the meanwhile, possible loss of information. This approach led to perform features selection, keeping, though, all the necessary information about stars and Exoplanets. Conversely, since the records amount were not so extensive, we thought that Oversampling would have been the best method to balance the classes. As showed from the Results section, the Random Forest can be considered as the best model for these data, in almost all the measures used for the analysis. In particular, it seems to retrieve well the records actually positive and to not make many mistakes about the positive class classification. In conclusion, despite the Kepler mission is now over, the NASA continues to send data to the scientist in order to find possible new Exoplanets. The use of Data Mining techniques has demonstrated, for this purpose, effective results and potential great insights. Our suggestion, thus, is to keep use and implement these and more techniques to develop reliable analysis in this field.

7 Further Developments

In this context, the project we have carried out can be improved and expanded especially by exploiting some more complex models. In particular, it could be appropriate the implementation of Multi-Layers Neural Networks to learn the data pattern and then perform predictions. Moreover, the k-fold Cross Validation could be operated by a greater k value.

8 References

[1] https://exoplanets.nasa.gov/the-search-for-life/exoplanets-101/
[2] Automatic classification of kepler planetary transit candidates, Sean D. McCauliff, Jon M. Jenkins.
[3] Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90, Christopher J. Shallue, Andrew Vanderburg.
[4] https://www.kaggle.com/nasa/kepler-exoplanet-search-results
[5] https://exoplanetarchive.ipac.caltech.edu/docs/API_kepcandidate_columns.html
[6] Introduction to Data Mining, Pang-Ning Tan, Michael Steinbach, Vipin Kumar.

Link to the code: https://github.com/AlbertoMonaco/DataMining