



Machine-learning Morphological Classification of TESS Light Curves of T Tauri Stars

Astaroth Elizabethson¹ , Javier Serna² , Alejandro García-Varela¹ , Jesús Hernández³ , and Juan F. Cabrera-García^{1,3} ¹ Universidad de los Andes, Departamento de Física, Cra. 1 No. 18A-10, Bloque Ip, AA 4976, Bogotá, Colombia; josegarc@uniandes.edu.co² Instituto de Astronomía, Universidad Nacional Autónoma de México, Ensenada, B.C, México³ Dipartimento di Fisica e Astronomia “Galileo Galilei,” Università degli Studi di Padova, Vicolo dell’Osservatorio 3, I-35122 Padova, Italy

Received 2023 March 16; revised 2023 August 22; accepted 2023 September 4; published 2023 October 10

Abstract

We present a variability and morphological classification study of TESS light curves for T Tauri star candidates in the Orion, IC 348, γ Velorum, Upper Scorpius, Corona Australis, and Perseus OB2 regions. We propose 11 morphological classes linking brightness variation behaviors with possible physical or geometric phenomena present in T Tauri stars, and develop a supervised machine-learning algorithm to automate the classification among these. Our algorithm optimizes and compares the true positive rate (recall) among k-nearest neighbors, classification trees, random forests, and support vector machines. This is done characterizing light curves with features depending on time, periodicity, and magnitude distribution. Binary and multiclass classifiers are trained and interpreted in a way that allows our final algorithm to have single or mixed classes. In the testing sample, the algorithm assigns mixed classes to 27% of the stars, reaching up to five simultaneous classes. A catalog of 3672 T Tauri star candidates is presented, along with their possible period estimations, predicted morphological classes, and visually revised ones. The cross-validation estimated performance of the final classifiers is reported. Binary classifiers surpass multiclass recall values for classes with less representation in the training sample. Support vector machines and random forest classifiers obtain better recalls. For comparison, another performance estimation of the final classifiers is calculated using the revised classes of our testing sample, indicating that this performance excels in singled classed stars, which happens in about 75% of the testing sample.

Unified Astronomy Thesaurus concepts: [Young stellar objects \(1834\)](#); [Light curve classification \(1954\)](#); [Variable stars \(1761\)](#); [Astrostatistics techniques \(1886\)](#); [Protoplanetary disks \(1300\)](#)

Supporting material: machine-readable table

1. Introduction

Stellar and planetary formation is a field that involves a wide range of scales and physical processes; some of these are associated with brightness variability. Particularly in T Tauri stars (hereafter TTS), pre-main-sequence stars with $M < 2 M_{\odot}$, operate different physical mechanisms, such as magnetic activity, accretion, protoplanetary disks evolution, multiplicity, and deuterium burning. These mechanisms produce photometric variability showing an extensive range of light-curve amplitudes and different levels of periodicity and timescales for variability (Herbst et al. 1994; Briceño et al. 2005; Morales-Calderón et al. 2011; Cody et al. 2014; Rebull et al. 2014; Fritzewski et al. 2016). Thus, a large diversity in TTS light curve (LC) morphology is expected.

Following Herbst et al. (1994), different mechanisms of photometric variability can be present in the two types of TTS: the Classical TTS (CTTS) with an accreting protoplanetary disk and the weak-line⁴ TTS (WTTS) in which the accreting mechanisms in the disk are stopped, or the accretion rate is below detectable levels. In CTTS, the gas is transported from the protoplanetary disk to the star, producing accretion luminosity and hot spots on the stellar surface (Hartmann et al. 2016). Thus, temporal variations in the

accretion rates produce changes in stellar brightness. Frequently, these types of brightness variations are named stellar bursts (Stauffer et al. 2014; Cody et al. 2017). Also, an increase in brightness is expected when the hot spot is located in the line of view, producing periodic or semiperiodic variability (e.g., Bouvier et al. 2007; Sousa et al. 2016).

Circumstellar material can also produce brightness variability. For example, dust clumps in the protoplanetary disk can produce extinction fluctuations, decreasing the observed brightness when the clump is in the line of view. Usually, these decreases in the brightness of the LC are called dips (Cody & Hillenbrand 2018; Venuti et al. 2021). In particular, if the dust overdensity is relatively stable, the LC analysis could provide information about the Keplerian rotation of the disk at the location of the dust clump. Regardless of the presence of an accreting disk, magnetic fields on the stellar surface produce cold spots that periodically decrease the flux in the LC when they are located in the line of view. Since this type of variability is regulated by stellar rotation, the LC analysis will provide rotational periods (e.g., Serna et al. 2021). Additionally, magnetic reconnection events on the stellar surface produce isotropic high-energy radiation bursts associated with stochastic spikes in the LC (flares; Davenport 2016; Günther et al. 2020). Stellar or planetary companions can originate periodic variability in the LC modulated by the orbital period of the system. Stellar spots, flares, and stellar/planetary companions can be present in both WTTS and CTTS; however, for the latter, the brightness variability produced by these phenomena could be masked by the protoplanetary accreting disk contributions (e.g., Herbst et al. 1994; Hinton et al. 2022). Taking into account these physical scenarios and the expected

⁴ The term “weak-line” is from the original classification based on the H_{α} intensity (Herbig & Bell 1988).



brightness variability behavior, Cody et al. (2014) applied a morphological LC classification method for young stellar objects in the NGC2264 stellar cluster based on the asymmetry brightness distribution and the periodical level of LCs using optical and infrared observations. The seven morphological classes obtained by these authors can represent different physical processes and geometric effects.

Nowadays, we are in a golden era of large-scale photometric and multiepoch surveys, as CoRoT (Auvergne et al. 2009), Kepler (Borucki et al. 2010), or TESS (Ricker et al. 2014), that generate massive data sets, so machine-learning techniques are required to extract information about astrophysical phenomena present in the diversity of detected signals and classify sources that produce periodic information patterns (Zhang & Bloom 2021). Selecting the features that represent in the best way the information about each kind of astrophysical object is a great challenge, and there are different scopes to do this. Some investigations prefer to automatically extract the features using deep learning algorithms as recurrent neural networks that decide which are the most suitable features to classify the data (Naul et al. 2018). Other investigations study and select the best features that represent the data. For example, in Pérez-Ortiz et al. (2017), robust descriptive statistics of the magnitude density were used as features to detect Be star candidates in the OGLE-IV Gaia south ecliptic pole field. This work allows us to find 50 new objects of this type with only six features. On the other hand, Pashchenko et al. (2017), using principal component analysis, studied the minimum number of features that contain the maximum information to classify variable stars in the OGLE-II field LMC_SC20. They found that the information needed to solve the problem of variability detection is contained in a subset of 10 (of 25) features provided by the Variability Search Toolkit code (Sokolovsky & Lebedev 2018) that they used.

More specifically, machine-learning techniques on TESS data have been used satisfactorily to study brightness variability associated with different physical processes. For example, automatic detection and classification of planets candidates (Pearson 2019; Yu et al. 2019; Olmschenk et al. 2021; Rao et al. 2021; Battley et al. 2022; Ofman et al. 2022), stellar flares (Feinstein et al. 2020; Vida et al. 2021), episodic bright dimming (dippers; Tajiri et al. 2020), and oscillating red giant stars (Hon et al. 2021) have been performed using machine-learning in several available sectors of TESS with different cadences. Moreover, Claytor et al. (2022) used a convolutional neural network and synthetic light curves modulated by stellar spots to infer stellar rotation periods automatically from a TESS sample.

Inspired by these previous studies, in this work, we apply machine-learning techniques to classify TESS LCs of selected TTS candidates located in different star-forming regions in the Milky Way. We propose 11 morphological classes, some of which could be linked to the diverse physical or geometric phenomena present in TTS. Then, we train several supervised machine-learning methods with the purpose of identifying and separating among these morphologies. And finally, we compare the performance of our best classifiers on new data and estimate the efficiency of the proposed automated process. With that objective, the structure of this study is described as follows. Section 2 describes the processes of extracting the TESS light curves and the training and testing samples with the morphological classes adopted in this work. A detailed

description of the machine-learning classifiers, the feature space, and the metrics are given in Section 3. The main results of this work and discussions are given in Section 4. Finally, our summary and conclusions are given in Section 5.

2. Light-curve Extraction and Samples

The TESS mission was mainly designed to detect transiting exoplanets, offering an exceptional opportunity to make variability studies that will allow us to better understand the astrophysical processes of the stars and planets. Nowadays, TESS has surveyed around 85% of the entire sky at least 27 days of continuous observation, and it has been optimized to efficiently search for exoplanet transit signals around more than 20,000 stars simultaneously (Ricker et al. 2014). Each month, the Mikulski Archive for Space Telescopes (MAST) collects and processes observations from the spacecraft and releases data products for the community. However, TESS does not release reduced LCs for every star. Instead, the mission releases the full-frame images (FFIs) for each sector and requires that users perform their photometric extraction, background subtraction, and any systematic correction.

We use the `TESSExtractor` tool⁵ (Serna et al. 2021) to perform LCs from the TESS cycle 1 and 2 (30 minutes of cadence). This tool downloads 10×10 pixel cutouts from the FFIs using the `TESScut` service (Brasseur et al. 2019), selects an optimal aperture, and performs simple aperture photometry (Bradley et al. 2019). The flux was extracted using circular apertures from 1.0–5.0 pixels of radius and properly calibrated to the TESS photometric system (Stassun 2019; TICv8.0). Neighbor stars could contaminate LCs due to the low angular resolution of TESS (21" per pixel squared). Based on the G filter of Gaia-EDR3 (Serna et al. 2021), the sky background level is estimated using the mode flux within the photometric annulus to minimize the effect of nearby sources in our photometric measurements. We reject all data points with a strong sky variability, above 95% of the median background estimation (MED_{sky}). Additionally, we include a contamination level estimator (CLE) based on the ratio between the flux of the target of interest and the sum of the fluxes of other Gaia sources that fall within the apertures (see Table 7). Also, the LCs dominated significantly by two or more signals could exhibit peculiar morphological behaviors in which mixed variability types could be present.

We adopt the TESS quality flags to reject any anomaly in the photometry (e.g., cosmic rays, popcorn noise, fireworks)⁶ and filter possible bad-quality photometry. We employ the task `kepcotrend` from the `PyKE` package (Still & Barclay 2012) jointly with the cotrending basis vectors (CBVs) provided by TESS to investigate how our sample is affected by systematic effects. This methodology finds the best match between the normalized LCs (F_{obs}) and the representative systematic vector generated by linear combinations of the first four CBVs (F_{sys}), and uses the minimization of $\chi^2 = \sum_i (F_{\text{obs},i} - F_{\text{sys},i})^2 / (F_{\text{obs},i})$, where i indexes over the data points of the LC. As a result, the goodness-of-fit indicates how systematic effects dominate the LCs. In Figure 1, we present the histogram of the χ^2 for our study samples. It reveals a bimodal distribution where the first peak represents the population of LCs mainly dominated by

⁵ An updated version of this tool is available at <http://www.tessextractor.app/>.

⁶ TESS Data release notes: https://archive.stsci.edu/tess/tess_drn.html.

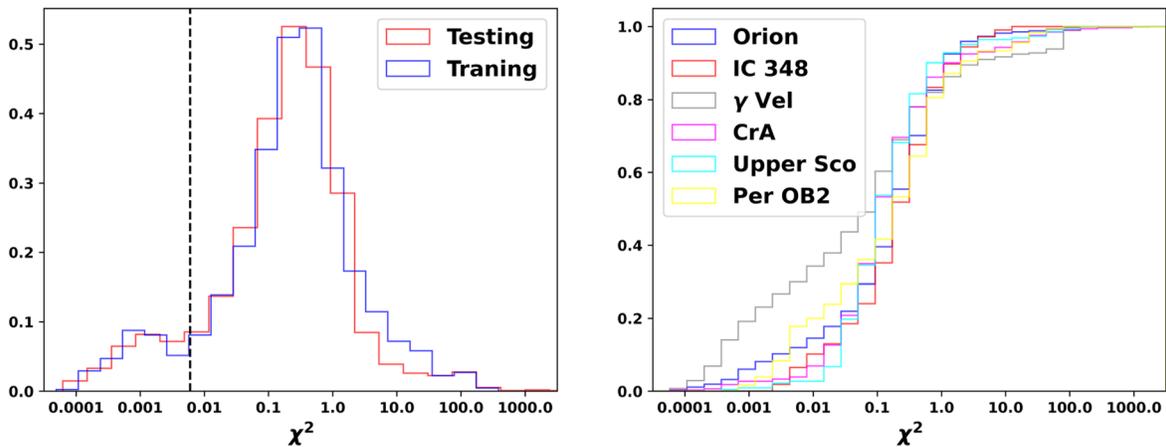


Figure 1. Normalized distributions of χ^2 for our study samples. The left panel shows comparative histograms of χ^2 for the training and testing samples. The dotted vertical line shows the quality cut $\chi^2 = 0.006$, defined as 3σ levels from the minor distribution peak. The right panel shows the cumulative plot of χ^2 per stellar region in the testing sample. The γ Velorum LCs could be more affected by systematic effects than other stellar regions. For example, the quality cut indicates that 30% of γ Velorum LCs are strongly dominated by systematics, while 10% of the LCs in other regions have this problem.

systematic effects ($\chi^2 < 0.006$). The second peak denotes that most of the population sample is not strongly dominated by systematics.

All of our approaches allowed us to define and apply a quality cut in the LCs. We use the MED_{sky} and median absolute deviation of the sky annulus LCs (MAD_{sky}) to clean our sample of stars located at the border of FFIs: $\text{MED}_{\text{sky}} > 16$ mag, and $\text{MAD}_{\text{sky}} > 0.2$ mag. Also, we reject stars dominated significantly by systematic: $\chi^2 < 0.006$. Even after this quality cut has been applied, our sample may have minor systematic effects that we discuss in Section 4.

2.1. Training and Testing Samples

Nowadays, the Gaia mission has provided exceptional measurements of distances and kinematics for stars in the solar neighborhood and opened the possibility of identifying thousands of clusters and young associations as well as confirming their members along the Milky Way (Cantat-Gaudin et al. 2018; Kuhn et al. 2019; Zari et al. 2019). Using clustering algorithms and information provided by Gaia-DR2, Kounkel & Covey (2019) performed a homogeneous search in the Galactic disk, identifying thousands of clusters within 1 kpc, estimating the ages of the individual groups, and providing a comprehensive catalog of clusters and their potential members.

We based this work mainly on the possible kinematic members reported by Kounkel et al. (2018), Kounkel & Covey (2019), and stars confirmed as TTS by Hernández et al. (2014), Briceño et al. (2019), and J. Hernández et al. (2023, in preparation), to build an extensive library of LCs of young stars. Our sample includes stars of regions such as the Orion star-forming complex (hereafter: OSFC), γ Velorum, IC 348 (with age ranging from 1–10 Myr), and more evolved regions such as Upper Scorpius, Corona Australis, and Perseus OB2 (with age ranging from 10–25 Myr). Particularly, OSFC has a wide variety of stellar populations in different environments, spanning different evolutionary stages from 1–10 Myr (Kounkel et al. 2018; Briceño et al. 2019; Zari et al. 2019), making OSFC an ideal star-forming region to drive studies of variability in young stars, especially in the T Tauri phase.

Multiple physical mechanisms in the early evolution of the stars are related to photometric variability observed in their

LCs, e.g., from accretion and mass outflow processes until magnetic activity on the stellar surface, the presence of surrounding material on the star, minor bodies, exoplanets, or even stellar companions (Cody et al. 2014; Venuti et al. 2017; Ansdell et al. 2019; Espaillat et al. 2021; Kounkel et al. 2021). We give a detailed description of each morphological class assumed in this work and possible mechanisms that originated as follows:

Periodic classes (P, Pi, Pd, and Pn). In TTS, these classes are mostly related to stellar rotation (e.g., Serna et al. 2021). The magnetic fields on the stellar surface produce cold spots that periodically decrease the flux in the light curve when they are located in the line of view. Sometimes the periodic light curve (P) has a global increase (Pi) or decrease (Pd) of brightness that could be associated with systematic effects in TESS photometry, or maybe produced by variability contributions from nearby stars, or even associated to long scale brightness variations intrinsic to the star itself. For example, the well-know T Tau star, has shown increasing variability episodes, followed by decreasing ones, on the scale of hundreds of days (Ismailov 2005). Additionally, in some periodic light curves, the noise starts to dominate the variability (Pn), particularly when the star is relatively faint.

Modulated classes (MP, Pia, and Pda). These light curves have periodic behavior, and additional periodic phenomena may modulate systematic variation of brightness amplitude. The multiple periods (MP) could originate from different scenarios that need to be confirmed. For example, multiperiods could originate by spot evolution and/or latitudinal differential rotation (Rebull et al. 2016). Alternatively, rotation plus pulsation could produce multiperiod light curves (Murphy et al. 2021; Venuti et al. 2021). Also, given the large pixel size of TESS, there is a possibility of contamination by nearby periodic stars. We have included periodic light curves that systematically increase (Pia) or decrease (Pda) the variability amplitude. This type of light curve could be a partial curve of an MP class with periodic variation larger than the TESS observational window.

Dipper and/or Bursters classes (DB). The dipper class is characterized by a relatively strong decrease in the brightness on the light curves (Cody & Hillenbrand 2018; Venuti et al. 2021). Extinction fluctuations due to clumpy circumstellar

material could produce a dipping in the brightness. When the dust clump is located in the inner disk, the light-curve analysis could provide information about the Keplerian rotational period, which could be assumed as the stellar rotational period when the clumpy is located within the corotation radius (e.g., AA Tau). On the other hand, the burster class is dominated by brightening events, generally stochastic, that could originate by variable accretion (Stauffer et al. 2014; Cody et al. 2017). The light curve has sporadic bursts of brightness over a quasi-static flux level. In addition, the accretion shock regions produce an increase of fluxes in the light curve when the hot spot is located in the line of view. If the hot spot lifetime is longer than the rotational period, the burster light curve could provide information about the stellar rotational period. Even though protoplanetary physical processes related to bursters and dippers can be different, we expect asymmetries in the brightness distribution in both types. Here we assume a single morphological class that includes dippers and bursters.

Eclipsing binaries class (EB) is an Algol-like light curve that exhibits transits that periodically decrease the brightness when the stellar or planetary companion passes in the line of view, where the beginning and ending time of the eclipse is well defined between relatively stable LC regions. Analysis of the light curve could provide information about the orbital period of the companion (Kounkel et al. 2021).

The long time variability class (L) includes LCs dominated by a systematic increase or decrease in brightness. This class could include stars with variability periods longer than the TESS observational window.

The noisy class (N) includes a stochastic or irregular light curve that shows flux variations with no apparent periodicity and no preference for brightening events (bursters) over fading events (dippers). Noisy light curves include stars with small or no signs of variability. This class includes stars with no apparent brightness change or faint stars with variability amplitude below the photometric uncertainties.

Since TTS show a very rich variety of LC morphologies, the features proposed to find them could be sensitive to classify variables stars with similar behaviors. For example, DB class stars could have similar LCs morphologies to Be or HAeBe stars. Also, periodic behaviors are not exclusive to some TTS, so our methodology could be extended to detect periodic variables such as RR Lyrae, Cepheids, or rotation-modulated variables. Nevertheless, these kinds of variabilities are not part of the scope of our paper, since the algorithm is trained explicitly with candidates and confirmed TTS.

To define our training sample, we randomly select approximately half of the sample reported as OSFC members (Kounkel et al. 2018; Briceño et al. 2019). We inspect each LC and select 956 members for the training sample. We visually classify this sample into the morphological classes described above, and end up with a training sample containing 363 Periodic stars (P), 108 Periodic noisy (Pn), 39 Periodic of increasing brightness (Pi), 45 Periodic of decreasing brightness (Pd), 20 Periodic of increasing amplitude (Pia), 24 Periodic of decreasing amplitude (Pda), 53 Modulated with more than one Periodical signal (MP), 132 Dipper and/or Bursters (DB), 125 Noisy (N), 17 Eclipsing Binaries (EB), and 30 Long timescale variables (L). Examples of each morphology in our training sample are shown in Figure 2. Finally, we have 2716 stars as a testing sample, including the OSFC members not used in the training sample and kinematic members in the other star-

forming regions. The catalog of the both samples, including TIC (TESS input catalog) ID and equatorial coordinates is presented in Appendix B (Table 7).

3. Machine-learning Classifiers

3.1. Introduction to Supervised Machine Learning

There are several types of supervised machine-learning algorithms that follow a general process that we describe here. Each individual data is described by a vector of numerical parameters called *features*, selected to represent in the best way stellar properties of interest. These will be the input of a predictive algorithm that will propose the probable class for new data. The set of all features constitutes the *feature space*. In our case, each star is represented by a set of features calculated over its LC (see Section 3.2). The algorithm is developed via a process that constructs a *prediction rule* using a set of previously classified data called the *training sample*. This prediction rule is a function over the feature space, and its output lies in the set of possible classes. The intention is to apply the prediction rule to estimate the classes for a new set called the *testing sample*. To develop the prediction rule using the training sample, each type of algorithm has an associated *loss function* used to quantify the level of misclassification. The optimization of the loss function evaluated over the training sample is the process that produces the classifier: a prediction rule *learned* from the training sample used as input data. The loss function has associated with a set of parameters called *hyperparameters*, which are tuned to optimize the performance of the classifier associated with a settled training sample. So, given a fixed set of values for the hyperparameters and a specific training sample, a sole classifier is obtained when the loss function is optimized. Both the hyperparameters and the feature space dimension affect the complexity of the resulting classifier. Too complex classifiers will *overfit* the training sample, showing a high variance, and will not likely perform as well with new data. On the other hand, a low-complex classifier will not approach the real behavior of the data since it is over-generalizing and has a high bias.

Usually, a grid of hyperparameters is used as input, so different final classifiers are obtained. Among them, the best-performed classifier is selected, which is the one that balances the bias-variance trade-off (Hastie et al. 2009). The notion of “best” depends on the *performance metric* used to compare each classifier. This performance metric is calculated over the output of an *evaluation set*, which is a set of previously classified data that is not used to train the classifier. Usually, to use all of the classified data as the training sample, a process of *cross-validation* is used when developing a classifier. In cross-validation, part of the initial training set is used as an evaluation set to estimate the performance of the produced classifier that uses the whole training set. Particularly, we use a 10-fold cross-validation, in which the training set is divided into 10 partitions. The partition is the evaluation set in each parallel process, while the rest is used to train a classifier. Since these 10 classifiers use 90% of the original training sample, they are expected to perform similarly to the classifier that uses the complete training set. The final estimation of the performance metric is the mean of the 10 metrics calculated over these 10 classifiers. The objective of cross-validation is to have estimations that are not correlated with the training set used,

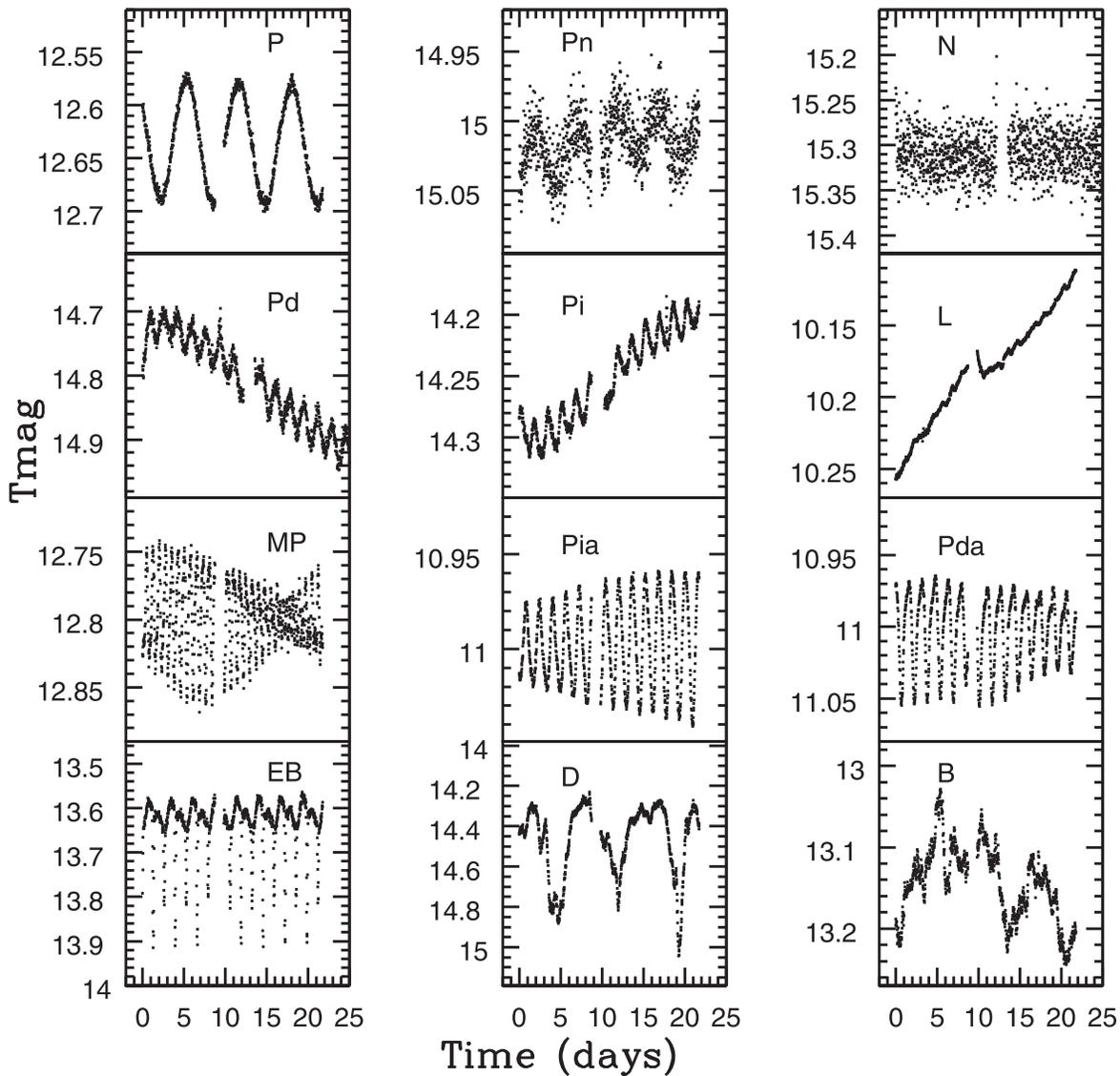


Figure 2. Morphological variability classes. We include some examples of stars representing the classes described in Section 2. (P) The star TIC 50657928 shows a periodic LC. (Pn) The star TIC 11296154 shows a periodic LC with significant noise. (N) The star TIC 249040852 exhibits a noisy LC, in which brightness variability is not detected. (Pi) The star TIC 50792805 exhibits a periodic LC with an overall increase in brightness. (Pd) The star TIC 249070479 shows a periodic LC with an overall decrease in brightness. (L) The star TIC 436013242 has a long-term variability. (MP) The star TIC 50748110 shows an LC with at least two periodic signals. (Pia) The star TIC 50622261 shows a periodic LC in which the variability amplitude increases with time. (Pda) The star TIC 4352899 shows a periodic LC in which the variability amplitude decreases with time. (EB) The star TIC 72829000 exhibits a typical Algol eclipsing binary LC. (D) The star TIC 11453668 shows an LC with sudden decreases (dips) in brightness, and (B) the star TIC 11296297 shows an LC with sudden increases (bursts) in brightness. Since we expect asymmetries in the brightness distribution in both dippers and bursters, we have joined them in a single class called the DB class.

to avoid overestimating the performance of an overfitted classifier.

Given the categorical partition in the train set, two types of classifiers can be identified: *binary classifiers* and *multiclass classifiers*. The binary classifier decides only between two classes, usually referred to as positive and negative. For positive, one of the possible classes is selected. For the negative, there are two approaches. The first approach, called *one-versus-one binary*, selects just one of the other classes. The second approach, called *one-versus-all binary*, selects the rest of the sample. This means that a complete set of binary classifiers must be constructed if a problem with more than two classes wants to be solved. On the other hand, multiclass classifiers decide simultaneously between the complete set of classes.

The performance metrics are defined given the entries of the confusion matrix of a classifier. For a binary classifier, all data consists of those whose real classification is positive (*RP*) and those whose real classification is negative (*RN*).⁷ The entries of the matrix consist of true positives (*TP*), true negatives (*TN*), false positives (*FP*), and false negatives (*FN*), and will follow the conditions $RP = TP + FN$ and $RN = TN + FP$. Correspondingly, proposed positives *PP* and proposed negatives *PN*, follow $PP = TP + FP$ and $PN = TN + FN$.

The most common metrics are accuracy, recall, precision, and the Cohen’s kappa coefficient (or just kappa). Accuracy is the proportion of correctly classified data to the whole data:

⁷ Usually the variables used in literature for real positive and negative cases are *P* and *N*. But we use this notation to avoid ambiguities with our classes “Periodic” and “Noisy”.

$(TP + TN)/(RP + RN)$. Sensitivity, also known as recall, calculates the true positive ratio to all real positive cases: TP/RP . Similarly, specificity, also called selectivity, calculates the true negative ratio to all real negative cases: TN/RN . Kappa is an estimation of the inter-rater agreement of two “observers” that assign categorical values to the same sample (Cohen 1960). In our case, the two observers are the real classification and the proposed classification given by the algorithms. Then, the kappa metric is calculated as $\left(\frac{p_0 - p_c}{1 - p_c}\right)$, where p_0 is the proportion of cases in which the observers agreed, and p_c is the proportion of cases for which agreement is expected by chance (Cohen 1960). For our binary confusion matrix, p_0 is the same as our accuracy, while p_c is calculated as $p_c = \left(\frac{TP}{RP}\right)\left(\frac{TP}{PP}\right) + \left(\frac{TN}{RN}\right)\left(\frac{TN}{PN}\right)$. So, the binary classifier kappa is determined by $\frac{2(TP \times TN - FN \times FP)}{(RN \times (TP + FP) + RP \times (FN + TN))}$. Is worth noting that all of these metrics are the results of calculations of various proportions over a confusion matrix constructed with a test sample, which in themselves represent a posteriori probability estimations.

In the case of multiclass classifiers, the calculation of accuracy and kappa can be extended for the whole confusion matrix. The dimension of a confusion matrix is the same as the number of defined classes, and the elements in the diagonal of the matrix are the correctly classified cases, so the accuracy is calculated by adding these diagonal terms and dividing them by the number of all cases. So, in the multiclass classifier, accuracy stills coincide with p_0 . On the other hand, p_c becomes a sum of terms per class, where each term is the probability of both observers assigning the same class. In contrast, sensitivity and specificity are individually calculated for each class present in the classifier.

In this work, we use multiclass classifiers and one-versus-all binary classifiers (hereafter just referred to as binary classifiers). Also, we use four different machine-learning models, *K-nearest neighbors* (KNN), *classification trees* (CART), *random forest* (RF), and *support vector machines* (SVM). More information about these algorithms, construction processes, hyperparameters, and programming is described in Appendix A. Also, both binary and multiclass classifiers are constructed for each type of model in an attempt to find the most suitable classifier to detect every one of the original classes, which implies different optimal hyperparameters, dimensionalities, and even training sets in every case. In the next Sections, further explanation is given.

3.2. Set of Features

Given that an LC is a set of n brightness measurements depending on time, we identify three groups of features. First, statistics estimators of the distribution of magnitude values $\{m_i\}_{1 \leq i \leq n}$, thus independent of time. Then, we have features directly calculated over the times series that are independent of a periodic behavior (i.e., the star’s LC, $\{(t_i, m_i)\}_{1 \leq i \leq n}$). Finally, we use features calculated over the folded LC, meaning that they depend on the periodic properties of the star. A summary of the set of features is given in Table 1.

3.2.1. Magnitude’s Distribution Features

We choose statistical quantities of the magnitude distribution. Some of them are robust in the sense that beyond a fraction of the contaminated points (breakdown point), the

statistic exhibits an arbitrary behavior with high (or low) values (Huber & Ronchetti 2009). As a location measurement, we choose the median of the magnitudes (*med*), which has a high breakdown point of 50%. As scale measurements, we choose the median absolute deviation (*mad*), which is a robust estimator of the dispersion of the magnitudes. We also use the inter-percentile range 2%–98% (*DeltaM*). As a skewness measurement, we use the *octile skewness* (*os*) proposed by Brys et al. (2004). As tail weight measurements, we use the *left octile weight* (*low*), and the *right octile weight* (*row*) proposed by Brys et al. (2006). As smoothness measurements, we use the Abbe value (*eta*) proposed by von Neumann (1941) and its modified version (*robAbbe*) reported by Pérez-Ortiz et al. (2017), which uses a robust measure of the location proposed by Huber (1964).

3.2.2. Time Depending Features

These features estimate magnitude behaviors in the time function without assuming a periodic behavior. This means that the cadence and regularity of the points in the LC highly influence their utility. We consider the following features:

Linear behavior: A simple least-squares linear regression is calculated among the LC obtained using the optimal aperture of TESS photometry (see Section 2). The slope and the r-value (Pearson correlation coefficient) of the linear regression are used as features (*slope*, *r_value*). The same values are calculated in the LC obtained using a 1.0 pixel radius of aperture, the minimal possible aperture used to obtain the photometry of the LC (*slope_min*, *r_value_min*).

Robust average Euclidean distance between successive points (recluiD): This feature uses the median estimation of the set of distances between successive points in the LC, $median(\{\sqrt{(m_{i+1} - m_i)^2 + (t_{i+1} - t_i)^2}\})$, where $1 \leq i \leq n - 1$. A version that takes the sign of the magnitude variation into account is also calculated (*reDSign*), $median(\{\text{sign}(m_{i+1} - m_i)\sqrt{(m_{i+1} - m_i)^2 + (t_{i+1} - t_i)^2}\})$.

Robust average immediate magnitude variation per time unit (rV): This feature calculates the median of the set of magnitude variation per time unit of successive points, $median(\{|(m_{i+1} - m_i)/(t_{i+1} - t_i)|\})$, where $1 \leq i \leq n - 1$, i.e., it is a robust estimator of the behavior of the magnitude rate change.

Robust average linear’s residue between interspersed points: abbreviated as rbLeon. Uses the median of the set of differences among the magnitude of a certain point in the LC, and its linear prediction using the two points that surround it: $median(\{|M_{\hat{t}_i} + B_i - m_i|\})$, where $2 \leq i \leq n - 1$, $M_i = \frac{m_{i+1} - m_{i-1}}{t_{i+1} - t_{i-1}}$ and $B_i = m_i - M_{\hat{t}_{i-1}}$. As with *recluiD*, we also calculate a version that considers the sign of the magnitude variation called *rbLeon-Sign*. Along with *recluiD* and *reDSign*, these features were proposed by León-Figueroa (2020).

Since these last three groups of features use the *median* function as a robust approximation, they are not affected by the gap that usually appears in the LCs due to the transmission window of TESS data.

3.2.3. Period Depending Features

In this work, we implement the Lomb-Scargle periodogram (LSP; Lomb 1976; Scargle 1982) using the `Gammapy` package (Nigro et al. 2019). Within this process, several features can be extracted from the LSP. To compute the periodogram, we use a

Table 1
A Summary of the 28 Features Originally Considered in Section 3.2

Feature	Sample Range	Description
med^R	$(7.62, 1.85 \times 10^1)$	$median(\{m_i\}_{1 \leq i \leq n})$
mad	$(9.09 \times 10^{-4}, 5.84 \times 10^{-1})$	$median(\{m_i - median(\{m_i\}_{1 \leq i \leq n})\}_{1 \leq i \leq n})$
os	$(-6.75 \times 10^{-1}, 8.64 \times 10^{-1})$	$(Q_{0.875} - 2Q_{0.5} + Q_{0.125}) / (Q_{0.875} - Q_{0.125})$
low	$(-8.09 \times 10^{-1}, 8.38 \times 10^{-1})$	$(Q_{0.375} - 2Q_{0.25} + Q_{0.125}) / (Q_{0.375} - Q_{0.125})$
row	$(-6.93 \times 10^{-1}, 9.32 \times 10^{-1})$	$(Q_{0.875} - 2Q_{0.75} + Q_{0.625}) / (Q_{0.875} - Q_{0.625})$
$DeltaM^R$	$(5.0 \times 10^{-3}, 1.56)$	$Q_{0.98} - Q_{0.02}$
$slope$	$(-7.18 \times 10^{-2}, 4.39 \times 10^{-2})$	Slope of the linear regression—optimal aperture
$slope_min^R$	$(-8.04 \times 10^{-2}, 4.70 \times 10^{-2})$	Slope of the linear regression—minimal aperture
r_value	$(-9.96 \times 10^{-1}, 9.98 \times 10^{-1})$	R-value of the linear regression—optimal aperture
$r_value_min^R$	$(-9.95 \times 10^{-1}, 9.98 \times 10^{-1})$	R-value of the linear regression—minimal aperture
eta^R	$(2.40 \times 10^{-4}, 2.26)$	$(n \sum_{i=1}^n (m_{i+1} - m_i)^2) / (2(n-1) \sum_{i=1}^n (m_{i+1} - \bar{m})^2)$
$robAbbe$	$(9.83 \times 10^{-5}, 2.52)$	$loc_{hub}(\{m_{i+1} - m_i\}_{1 \leq i \leq n}) / loc_{hub}(\{m_i - loc_{hub}(\{m_i\}_{1 \leq i \leq n})\}_{1 \leq i \leq n})$
rV	$(8.10 \times 10^{-3}, 6.18)$	$median(\{ m_{i+1} - m_i / (t_{i+1} - t_i)\}_{1 \leq i \leq n-1})$
$reucID$	$(2.08 \times 10^{-2}, 1.31 \times 10^{-1})$	$median(\{\sqrt{(m_{i+1} - m_i)^2 + (t_{i+1} - t_i)^2}\}_{1 \leq i \leq n-1})$
$reDSign^R$	$(-2.13 \times 10^{-2}, 2.13 \times 10^{-2})$	$median(\{sign(m_{i+1} - m_i) \sqrt{(m_{i+1} - m_i)^2 + (t_{i+1} - t_i)^2}\}_{1 \leq i \leq n-1})$
$rbLeon$	$(1.25 \times 10^{-4}, 1.06 \times 10^{-1})$	$median(\{ M_i t_i + B_i - m_i\}_{2 \leq i \leq n-1})$
$rbLeonsign^R$	$(-5.30 \times 10^{-3}, 4.03 \times 10^{-3})$	$median(\{(M_i t_i + B_i - m_i)\}_{2 \leq i \leq n-1})$
per	$(1.50 \times 10^{-1}, 2.46 \times 10^1)$	Period with the highest value in the LSP
per_2	$(1.25 \times 10^{-1}, 2.50 \times 10^1)$	Period with the second highest value in the LSP
per_3^R	$(7.50 \times 10^{-2}, 2.50 \times 10^1)$	Period with the third highest value in the LSP
fm	$(3.0 \times 10^{-3}, 9.91 \times 10^{-1})$	Highest value in the LSP
sm	$(1.0 \times 10^{-3}, 7.69 \times 10^{-1})$	Second highest value in the LSP
tm^R	$(1.0 \times 10^{-3}, 3.82 \times 10^{-1})$	Third highest value in the LSP
$tail$	$(0.0, 9.21 \times 10^{-1})$	Value of the LSP at $P = 25$ days
Env	$(-1.73 \times 10^{-1}, 9.89 \times 10^{-2})$	$USlope - LSlope$
$USlope$	$(-1.75 \times 10^{-1}, 9.25 \times 10^{-2})$	Slope of the linear regression using peaks of the LC
$LSlope^R$	$(-1.0 \times 10^{-1}, 9.96 \times 10^{-2})$	Slope of the linear regression using troughs of the LC
dtw	$(2.35 \times 10^{-1}, 3.08 \times 10^2)$	$\min(\sqrt{\sum_{k=1}^K w_k})$

Notes. For os , low , row and $DeltaM$, the p -percentiles (Q_p) are calculated over the magnitude distribution. For $robAbbe$, the Huber (1964) location measure loc_{hub} is calculated. The term w_k of the dtw is the distance matrix between the synthetic LC and the original LC.

^R These features were removed from the final feature space as described in Section 3.3.

linear grid of 1000 periods within an interval of $0.04 < P < 25$ days (hereafter, window-search). Our minimum period is twice the TESS cadence, and the maximum period of 25 days is approximately the covered observing time of TESS. Each periodogram was processed to estimate the LSP power for the three highest peaks. The first one (fm) is usually associated with the rotation period (per), and the last ones (sm , tm) to secondary and tertiary components of the periodogram (per_2 , per_3). In some instances, the LSP does not converge to a reliable period. This could be due to a couple of different scenarios: long-term variability not covered in the window-search or possible remaining systematic effects in LCs. We use the LSP power at 25 days ($tail$) as a feature to detect those cases.

Assuming a periodic pattern can be useful to morphologically classify LCs, since some stars could change the periodic behavior on time. The following features take into account this scenario.

Dynamic time warping (dtw): is a technique used to measure similarity between a pair of LCs or signals, recognizing similar shapes, even if both time-series do not have the same length or have different amplitude and timescales (Giorgino 2009). Given two LCs (T and S), where $T = \{t_1, t_2, \dots, t_n\}$ and $S = \{s_1, s_2, \dots, s_m\}$, of length n and m , respectively. The dtw method computes the distances of each point of T and points from S , generating a distance matrix $distMatrix(t_i, s_j)$, with $1 \leq i \leq n$ and $1 \leq j \leq m$. The final objective is to find a warping path of

the contiguous elements $W = \{w_1, w_2, \dots, w_l, \dots, w_K\}$ associated to the $distMatrix(t_i, s_j)$, such that $\max(n, m) < K < m + n - 1$; $w_l = dist(t_i, s_j)$, and it minimizes the function $dtw(T, S) = \min(\sqrt{\sum_{i=1}^K w_i})$. Where the last element of the warping path, w_K , corresponds to the distance calculated with the dtw method (Cassisi et al. 2012). We use the *FastDTW* package (Salvador & Chan 2007) to estimate the DTW similitude between an LC of our sample and a synthetic template. Each template was created using the phase-folded LC, duplicating the recovered structure to the same temporal length of the observed LC (e.g., $t_i = s_i$). Finally, we use the output distance as the feature called dtw , which measures how similar is the periodic pattern over time.

Envelope trend (Env): The purpose of this feature is to measure how the amplitude of an LC changes over time. To do this, we identify both the peaks and the troughs on the LC, and then we perform a linear regression over each of these sets of points. The values for the slopes are then presented as separate features, i.e., Upper slope (Uslope for the peaks), and Lower slope (Lslope for the troughs). The Env feature is defined as: $Env = Uslope - Lslope$. Identification of the peaks and troughs is obtained using the *SciPy* package `scipy.signal.find_peaks` (Virtanen et al. 2020). In order to accurately find the peaks and troughs, the calculated period is included as a parameter for the required minimum horizontal distance between neighboring peaks.

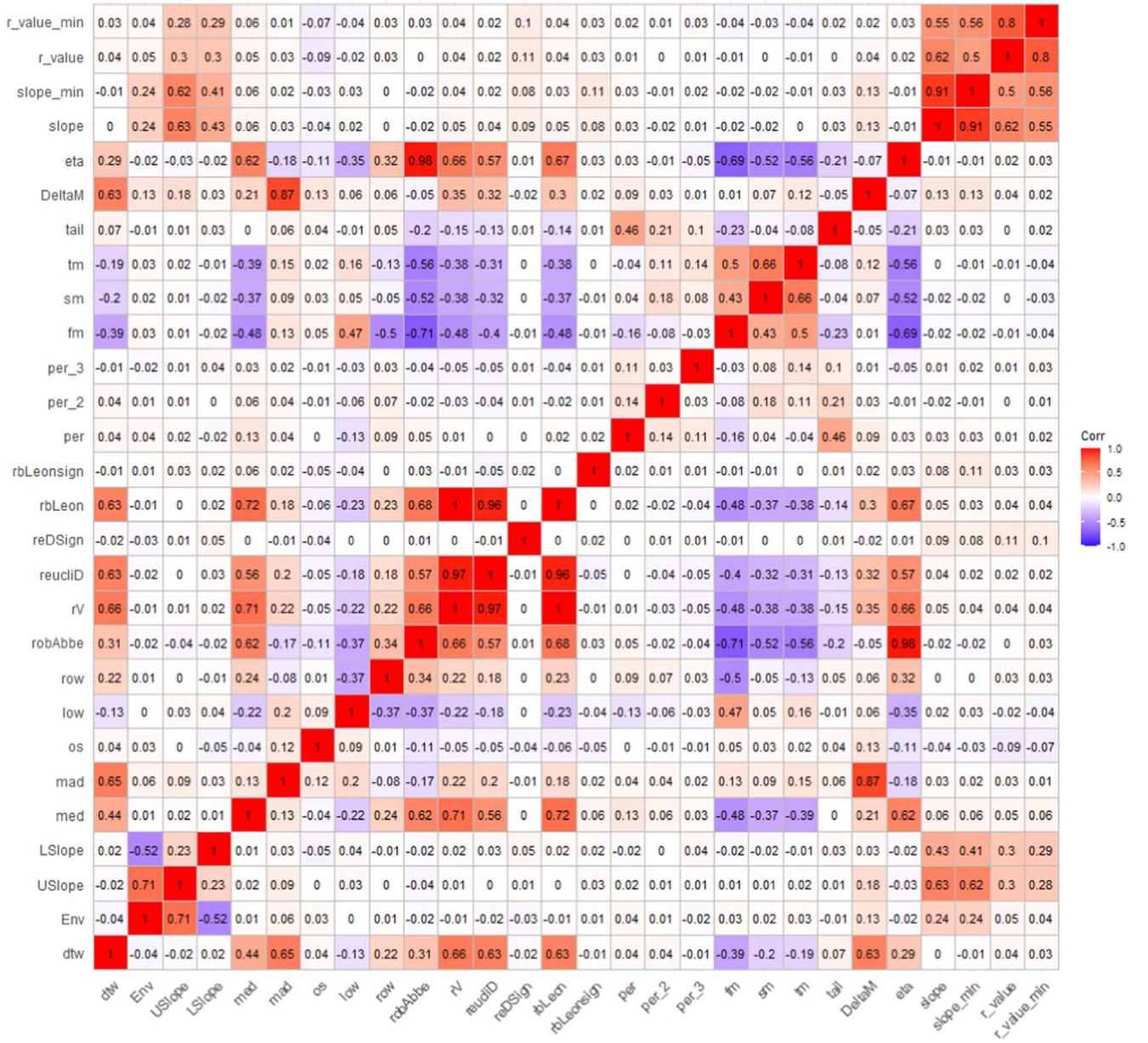


Figure 3. Correlation matrix of the initial 28 features calculated over all of our data (training and testing samples). Since there are few NAs values in *DeltaM*, those correlations are calculated over slightly less data.

3.3. Space Feature Dimension Reduction

We use two approaches to reduce the size of the initial 28 feature sets: correlation analysis and RF importance analysis. In this reduction process, we also consider that some of our features are mathematically related or trace the same characteristics in LCs. Gabruseva et al. (2020) showed that using the Pearson correlation matrix as a simple method to discard redundant features can be more effective than automated feature selection methods. The initial correlation matrix of all of the data is shown in Figure 3. Additionally, a correlation matrix of each sample was calculated since observing the individual matrix’s behavior in each one is especially important in cases where the training and testing samples come from different sky regions. Even though stellar formation is expected in the studied regions, the observed properties of the LCs in each region could change due to

different distances, extinctions, or observing conditions (e.g., see Figure 1). In our case, the correlations among features did not change significantly between the training and testing samples.

Numerically, optimal and minimal aperture photometry generate two different LCs. However, the high correlation between *slope* and *slope_min*, and between *r_value* and *r_value_min*, means the same first-order behavior. So we keep only the features *slope* and *r_value*. As expected, the Abbe value (*eta*) and its robust version (*robAbbe*) give the same information, so we keep the latter one as a noise indicator. Finally, a high correlation exists among *DeltaM* and *MAD*, which is expected as both are dispersion measures. Also, due to the high percentiles required to estimate *DeltaM*, it is not possible to calculate this feature for some few LCs (27 stars in training and 10 in testing samples), and are reported as *NA*. So *MAD* is kept, and *DeltaM* is discarded. In addition, there are

features that, even though they do not have such high values of correlation between them, their removal will avoid redundant information in the feature space. For instance, the triad *Env*, *USlope*, and *LSlope* are linearly dependent since the first one is calculated using the other two (see Section 3.2.3). Since *LSlope* shows the less important value of this triad in the iteration of RF using the original 28 features (see Section 3.4), it was extracted from the final feature set. Similarly, the values of the peaks of the LSP have ordered values by definition, meaning that *tm* will have lower values than the other two (*fm*, *sm*). Moreover, the importance in the previous iterations of RFs of *tm* and the corresponding period (*per_3*) is low enough to be extracted from the feature space. Finally, we remove the feature *med* because it depends on the distance of the star, and the algorithms are trained using only data from the OSFC, which could have different distances from the other young stellar populations in the testing data.

In order to further understand the role that our features play in the classification problem, we also conduct the following RF importance analysis. We generate an additional feature containing a random value and normalize the entire set using their minimum and maximum values. Then, we calculate the RF importance of each of these features, and compare it to the importance of the added random variable, estimating its relevance to our problem. By performing this experiment over our training sample, we find that the features *reDSign* and *rbLeonSign* are less important than the randomly generated feature, indicating their irrelevance to this problem. Comparing features using this method helps to identify nonessential features that correlation analysis will overlook. In this case, *reDSign* and *rbLeonSign* are random enough to have almost zero correlation with all other features, but not informative enough to add new information for an RF classifier. So, after this process, we end up with 18 final features.

Initially we do multiple iterations, training several classifiers using different subsets of the training sample. These previous iterations of classifiers showed a trade-off between general performance and good detection of less represented classes, even when they were trained using resampled versions of the training sample to deal with the class imbalance problem. Another phenomenon affecting the training process is that classifiers could detect similar properties that naturally appear in different classes, reflecting usual confusion between specific subsets of classes.

To address these issues, we use an approach that develops an independent classifier optimized for each class. That means one prediction for each class is made for every single star. Ideally, if the classes completely represent a partition of all possible behaviors of the testing sample, only one classifier will accept a star, and the rest will reject it. As presented later in Section 4, this ideal behavior occurs in the majority of our data. Nevertheless, crossed acceptances also occur in up to five classes (commonly two), meaning that our classifiers can identify different behaviors. Furthermore, few stars (9.1%) are rejected by all classifiers, supporting the idea that the proposed morphologies cover most part of the LC behaviors. However, they will not completely cover all of them.

3.4. Optimal Classifier Selection per Class and Performance Evaluation

Once we have the 18-dimensional feature space, we go through several stages of optimization for each classifier

developed per class. Every time we train a classifier, we do a hyperparameter grid search to obtain the best performance. In this first stage, the optimal hyperparameter values are selected for each classifier type. The results of the multiclass classifier are compared using the kappa metric, and those from the binary classifier are compared using the sensitivity metric. This stage contributes most to the total computational time required to complete the training algorithm because multiple classifiers are trained for each possibility in the hyperparameter grid, and we use cross-validation to estimate the performance values.

Class imbalance is tackled in this stage. We use under-sampled subsets of the training sample, reducing the presence of the most represented classes. This is a common and straightforward approach to solve the unbalance problem, at the cost of possible information loss during the process (Sun et al. 2009). Nevertheless, for our data, better results were obtained using this technique, and no further or complex solutions were necessary. Specifically, for multiclass classifiers, only class P was rebalanced, reducing from 363 LCs to 200 LCs selected randomly. It is worth mentioning that optimization results from our multiclass classifiers have no significant differences whether we use the balanced sample or not, even when we use the usual performance metric accuracy to select along the hyperparameters grid instead of using the kappa metric.

The unbalance in the training data is more evident for binary classifiers, and it is drastic for the poorly represented classes, such as Pda or Pia. In each binary model we train, all classes but the respective positive class are randomly reduced to the size of the positive class to help balance members that will be part of the negative class in a corresponding binary classifier. For example, there are 45 Pd LCs in our original training sample and 911 non-Pd stars. So, when the Pd binary classifier is constructed, only 45 P random LCs are added to the non-Pd training sample. The same reduction is made for classes with sizes larger than 45 LCs (Pn, MP, DB, and N). In this case, for classes with sizes smaller than 45 LCs (Pi, Pia, Pda, EB, and L), their complete samples are added to the non-Pd class in the current training sample. Ultimately, the current training sample consists of 45 Pd versus 355 non-Pd stars. This process does not completely balance the training sample but greatly improves the performance of the model we train, especially when sensitivity is used as a performance metric to optimize the hyperparameters.

Then, we have another stage that optimizes the initial dimension of the space feature, along with the specific features in it. For binary classifiers, we begin by training a set of RF binary classifiers in the 18-dimensional feature space. The performance of those RF classifiers sorted by the importance of each feature is given in Table 2. Then, the order of importance is used to progressively reduce the set of features that each trained classifier initially receives, excluding the less important features first. We must not confuse this reduction process of the feature space dimension with the reduction of feature dimension in each split in the bagging process of RFs (see Appendix A.3).

For the multiclass case, a similar but simpler process is performed. Only one RF classifier is trained using all 18 features and the complete training sample. So, the order in which features are reduced is always the same. Table 3 shows the calculated importance values. Nevertheless, since we want later to compare the performance of different binary and multiclass classifiers, recall is calculated for each class in each

Table 2
Binary RF Feature Importance Order (i_n) and Their Reported Mean Impurity Decrease, Using Gini Index (see Appendixes A.2 and A.3)

Class	Kappa Sensitivity	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8	i_9	i_{10}	i_{11}	i_{12}	i_{13}	i_{14}	i_{15}	i_{16}	i_{17}	i_{18}
DB	0.925	mad	DTW	Env	USlope	robAbbe	sm	os	row	fm	rV	per	slope	reucID	r_value	rbLeon	tail	low	per_2
	0.917	74.799	28.685	17.654	15.612	14.634	10.102	9.087	8.831	6.756	4.492	3.981	3.937	3.758	3.158	3.078	2.696	2.526	1.452
MP	0.580	sm	per	per_2	fm	os	low	tail	rbLeon	DTW	robAbbe	mad	rV	reucID	Env	row	USlope	slope	r_value
	0.566	23.159	19.626	6.634	4.950	4.461	3.839	3.098	3.064	2.946	2.916	2.785	2.768	2.681	2.453	2.256	1.874	1.329	1.225
Pia	0.664	Env	per_2	DTW	fm	USlope	rV	per	reucID	rbLeon	row	os	sm	tail	robAbbe	r_value	slope	mad	low
	0.550	7.751	5.258	3.795	3.448	2.068	2.011	1.835	1.691	1.555	1.039	0.969	0.871	0.699	0.644	0.590	0.532	0.476	0.401
Pda	0.665	Env	DTW	fm	per	reucID	r_value	mad	rbLeon	rV	tail	low	sm	robAbbe	per_2	row	slope	os	USlope
	0.667	16.916	9.984	3.178	2.186	1.352	1.204	1.141	1.062	0.980	0.844	0.821	0.805	0.785	0.743	0.720	0.577	0.481	0.471
Pi	0.850	r_value	per	slope	USlope	tail	fm	per_2	Env	sm	row	mad	DTW	robAbbe	reucID	rV	os	rbLeon	low
	0.821	21.617	11.112	7.633	4.697	4.530	3.308	3.038	1.898	1.860	1.680	1.500	1.157	1.086	0.972	0.938	0.807	0.806	0.779
Pd	0.637	r_value	fm	sm	USlope	per	slope	DTW	per_2	mad	robAbbe	tail	os	row	Env	low	rV	rbLeon	reucID
	0.622	22.876	11.355	7.061	5.301	4.777	4.753	3.149	3.093	2.740	2.583	2.436	2.224	1.688	1.563	1.430	0.922	0.898	0.875
P	0.821	fm	row	DTW	low	tail	robAbbe	sm	USlope	r_value	reucID	mad	rV	rbLeon	slope	per	os	Env	per_2
	0.901	56.524	37.251	32.844	31.570	28.862	28.068	24.452	21.980	21.663	21.247	21.156	20.995	20.700	19.189	17.022	16.594	14.788	14.246
Pn	0.911	fm	robAbbe	per	rbLeon	mad	r_value	tail	sm	rV	slope	DTW	reucID	low	per_2	row	USlope	Env	os
	0.917	70.121	69.311	9.847	9.061	3.673	3.353	2.606	2.253	1.512	1.461	1.219	1.114	0.976	0.932	0.778	0.726	0.639	0.493
N	0.922	fm	robAbbe	sm	per	mad	rbLeon	reucID	rV	low	os	tail	r_value	row	DTW	per_2	USlope	slope	Env
	0.928	61.209	38.901	32.393	16.656	7.703	7.371	6.466	6.416	3.803	3.702	3.602	2.943	2.870	2.851	2.620	2.458	2.301	1.971
L	0.926	tail	r_value	per	sm	per_2	slope	robAbbe	fm	row	Env	USlope	low	rV	reucID	os	DTW	rbLeon	mad
	0.933	22.709	8.120	7.017	3.087	2.565	2.162	2.000	0.909	0.878	0.718	0.713	0.710	0.601	0.542	0.437	0.375	0.356	0.304
EB	0.336	Env	rV	row	low	fm	os	tail	reucID	rbLeon	mad	per	USlope	slope	per_2	DTW	robAbbe	sm	r_value
	0.235	3.309	3.242	2.707	2.492	2.453	2.358	1.705	1.484	1.386	1.342	1.329	1.319	1.228	1.172	0.959	0.918	0.869	0.540

Note. Since a binary RF is trained for each class, we have a different feature importance order and values per each one.

Table 3
RF Feature Importance for a Multiclass Classifier

Feature Importance	fm	robAbbe	Tail	MAD	Per	SM	r_value	DTW	Row
	0.160687	0.121865	0.070729	0.061459	0.061281	0.053784	0.053445	0.051867	0.045698
Feature Importance	USlope	slope	rbLeon	rV	low	Env	os	reucID	per_2
	0.040866	0.037967	0.037147	0.037044	0.034761	0.034011	0.033671	0.031921	0.031796

Note. The values are the reported normalized mean impurity decrease, using Gini index (see Appendixes A.2 and A.3).

multiclass classifier developed using a different feature space. This also means that sometimes the same classifier is selected as optimal to detect different classes when the highest recall is obtained by that classifier in two or more classes (see Table 4).

In this stage of optimization, an interesting pattern occurs. Typically, a classifier developed in a feature space with larger dimensions is usually more complex than one developed with a smaller feature space. This does not necessarily mean better performance, even in cases where the used model is less prone to overfit. The performance behavior of the multiclass and binary classifiers usually obtains a maximum value with lesser dimensions. This behavior is the motivation to conduct this stage of feature space’s dimension optimization. For example, Figure 4 shows a method to estimate the performance of multiclass and binary classifiers. Particularly, we calculate the recall for each class against the dimension of the feature space. The right panel of Figure 4 illustrates the compared behavior of the recall of the SVM binary classifiers for each class. Some classes, such as Pd or Pia, clearly obtain maximum performance values at lower dimensions. In this case, a recall of 0.8 is obtained using four features for Pd and a recall of 0.96 using three features for Pia. On the other hand, classifiers for N and L classes maintain similar performance values through most dimensions. Left panel of Figure 4 shows only the recall behavior of SVM multiclass classifiers for DB and N classes and RF multiclass classifiers for the P class since these are the ones relevant in the final selection performed in Section 4.1.

We choose recall for comparing performance metrics between multiclass and binary classifiers. This is because accuracy or kappa can be biased when calculated over the training sample’s confusion matrix in binary classifiers and can get higher values than those calculated in the multiclass ones. These higher values do not necessarily imply better performance due to class imbalance being more noticeable in binary classifiers than in multiclass classifiers.

4. Results and Discussion

4.1. Model Selection and Final Algorithm Evaluation

All of the processes described in Section 3.4 are executed for each of the four different classifier models considered in this work (KNN, CART, RF, and SVM). Until this point, we have independently obtained the best binary and multiclass classifier per each class and each model. SVM and RF models stand out; their binary classifiers usually have almost the same performance values, with SVM always being better (see the left side of Table 4). For multiclass, some RF models actually surpass SVM. Even though RF models have a better recall, in some cases the values obtained by the classifiers are below 0.7. For example, Pia and Pda classes have the lowest performance value among them (see the right side of Table 4). Generally, CARTs are the third best-behaved models, sometimes even exceeding RF binary models when the characterized class has

Table 4
Best Performed Classifiers among Binary (Left Columns) and Multiclass (Right Columns) per Class, Showing Models and Feature Space’s Dimension

Recall	Binary Model	Dim	Class	Recall	Multiclass Model	Dim
0.947	SVM	9	DB	0.977	SVM	5
0.774	SVM	9	MP	0.642	SVM	6
0.950	SVM	3	Pia	0.350	RF	18
0.917	SVM	6	Pda	0.542	RF	12
0.897	SVM	6	Pi	0.872	SVM	18
0.800	SVM	4	Pd	0.667	RF	16
0.948	SVM	8	P	0.972	RF	6
0.963	SVM	4	Pn	0.944	RF	6
0.976	SVM	5	N	0.976	SVM	6
1.000	SVM	5	L	0.967	RF	6
0.824	SVM	17	EB	0.647	RF	4

Note. The ones with the highest recall (in bold) are chosen to give the final predictions to the testing sample. The classifier with higher accuracy for the tied values in the N class was selected.

low representation in the training sample. Finally, KNN models are the least effective, mostly affected by class imbalance. As a summary, in Table 4, we show the final comparison between binary and multiclass classifiers presenting the best model and feature space’s dimension per class.

Interestingly, Table 4 shows that for classes DB, P, and N, which have a large representation in the training sample, the multiclass classifiers have higher sensitivity even though binary classifiers are individually optimized for each class. Thus, these multiclass classifiers are included in the final prediction algorithm. For the remaining classes, we choose the binary classifiers. This final prediction algorithm consists of a set of the best-performed classifiers for each of the 11 classes. Therefore, 11 predictions are obtained for each star in the testing sample, proposing whether or not the star is a candidate for that morphology class. Those classifiers, the set of features, and the hyperparameter values they end up using are presented in Table 5. This last characteristic is an uncommon approach in machine learning. Traditionally, a single final class is predicted for new data. Clearly, this happens when a single final multiclass classifier is selected. On the other hand, when binaries classifiers are used, a best-probability criterion allows for the final decision. Nevertheless, our approach intentionally permits mixed classifications since the physical phenomena associated with brightness variability behaviors can occur simultaneously. Even when the training sample was initially being classified by visual inspection, a subset of LCs with uncertain classification were discarded from the training sample and added to the testing sample. Some of these LCs have characteristics of two or more morphological classes. Far from being a problem, this allows us to select and develop a pipeline that takes into account the possible presence of mixed classes (Section 3.4). The final

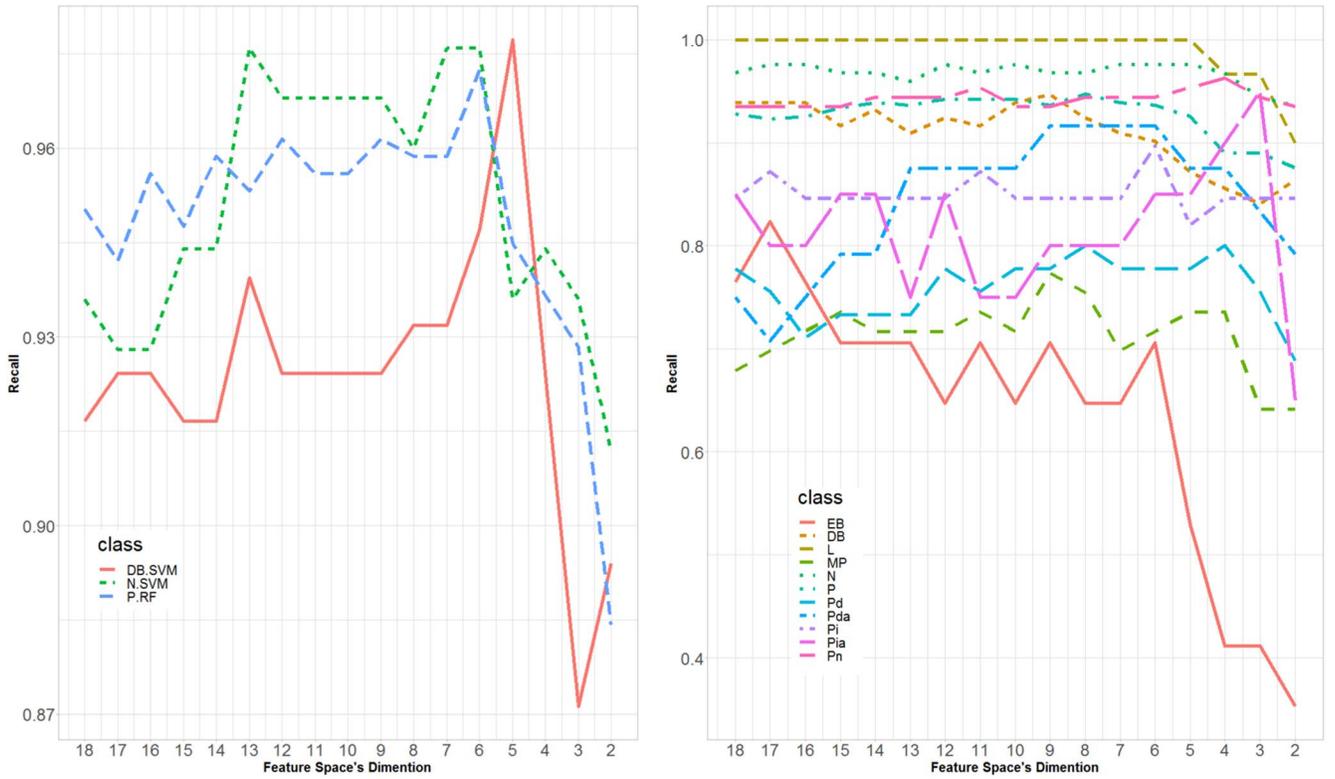


Figure 4. Recall values of multiclass (left panel) models as a function of the feature space’s dimension, for DB, N, and P classes. The order in which the features are decreased is given by the importance in Table 3. Recall values of SVM binary models (right panel) in function of the feature space’s dimension. For each class, the order in which the features are decreased is given by the importance in Table 2.

Table 5
Best Performed Classifier for Each Morphological Class, Showing Models and Feature Space’s Dimension

Class	Recall	Model	Dim	Features Used	Hyperparam
DB	0.977	SVM multiclass	5	<i>fm,robAbbe,tail,mad,per</i>	$C = 10^5, \gamma = 10^{-2}$
MP	0.774	SVM binary	9	<i>sm,per,per_2,fm,os,low,tail,rbLeon,dtw</i>	$C = 10^{8.5}, \gamma = 10^{-4}$
Pia	0.950	SVM binary	3	<i>Env,per_2,dtw</i>	$C = 10^{4.5}, \gamma = 10^{-1}$
Pda	0.917	SVM binary	6	<i>Env,dtw,fm,per,reucliD,r_value</i>	$C = 10^{4.5}, \gamma = 10^{-1.5}$
Pi	0.897	SVM binary	6	<i>r_value,per,slope,USlope,tail,fn</i>	$C = 10^{4.5}, \gamma = 10^{-1.5}$
Pd	0.800	SVM binary	4	<i>r_value,fm,sm,USlope</i>	$C = 10^{6.5}, \gamma = 10^{-2.5}$
P	0.972	RF multiclass	6	<i>fm,robAbbe,tail,mad,per,sm</i>	$mtry = 1$
Pn	0.963	SVM binary	4	<i>fm,robAbbe,per,rbLeon</i>	$C = 10^{5.5}, \gamma = 10^{-1.5}$
N	0.976	SVM multiclass	6	<i>fm,robAbbe,tail,mad,per,sm</i>	$C = 10^6, \gamma = 10^{-2}$
L	1.000	SVM binary	5	<i>tail,r_value,per,sm,per_2</i>	$C = 10^{0.5}, \gamma = 10^{-1}$
EB	0.824	SVM binary	17	<i>Env,rV,row,low,fm,os,tail,reucliD,rbLeon, mad,per,USlope,slope,per_2,dtw,robAbbe,sm</i>	$C = 10^3, \gamma = 10^{-2.5}$

Note. The different features used in those final classifiers are shown along with their hyperparameters values. It is important to emphasize that a perfect recall score does not imply a complete well-classified evaluation sample, given that recall only counts for true positive cases. For our data, the L final classifier has a recall of 1, but their specificity is 0.9889 since it does not reject perfectly the non-L stars.

predicted classes (even mixed classes) for each star in the testing sample are presented in the sixth column of Table 7. A summary of the stars proposed in the test sample by each final classifier is shown in the first row of Table 6.

A visual revision of the LCs in the testing sample is conducted to confirm the morphology proposed by the final classification algorithm. The revised classes of all LCs in the testing sample are shown in the seventh column of Table 7. Also, the number of stars in the test sample by class, after their revision, is shown in the second row of Table 6. This validation process allows us to directly estimate the performance of our final classifier using the proportions of each class in the testing

sample, i.e., the a posteriori probability of belonging to each class. This, in turn, lets us compare this validation process with the estimations of efficiency made by the algorithm using cross-validation over the training sample (Section 3). Figure 5 compares the recall estimations for each final classifier among three sets: the entire testing sample, the subset of the testing sample containing LCs with only one revised class, and those with two revised classes.

The visual inspection also permits the identification of extreme outliers that can affect the morphological classification of the LCs. First, we have stochastic physical phenomena such as flares or a single transit in the observational window.

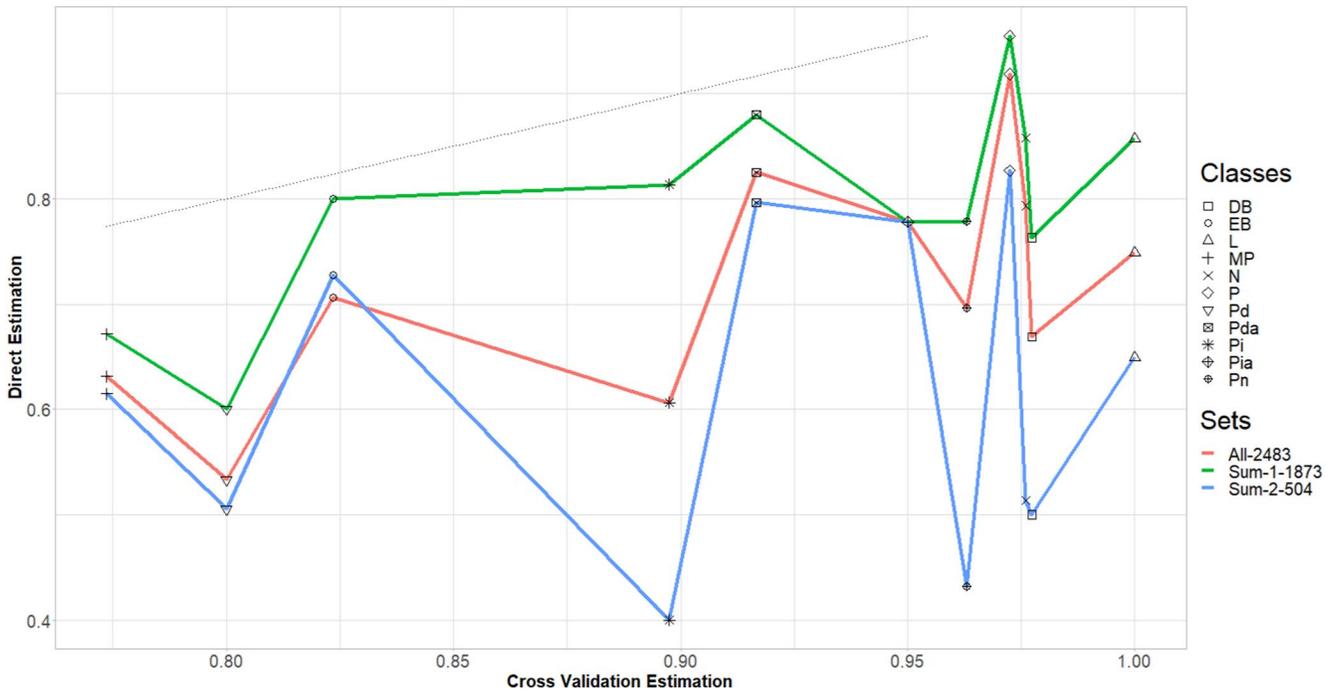


Figure 5. Comparison of the recall estimations for each final classifier. The dotted line represents the identity. The x-axis shows the recall estimation done by the classifiers using cross-validation over the training sample. The y-axis shows the recall estimated using as an evaluation set the complete testing sample (All), the LCs in the testing sample with only one final class verified by visual inspection (Sum-1), and those with two final verified classes (Sum-2). Since each class has the same recall from cross-validation using different sets, they are located in the same position on the x-axis.

Table 6
Number of Stars Presented by Class, or Total Assigned Classes, Respectively

	Stars by Class											Stars by Total Classes						
	DB	MP	Pia	Pda	Pi	Pd	P	Pn	N	L	EB	0	1	2	3	4	5	
Predicted	252	281	131	147	136	157	940	359	569	202	237	260	1658	661	118	18	1	
Inspected	159	215	58	90	176	179	905	409	688	239	18	131	2049	521	15	0	0	

Note. The “0” designation implies that no class was assigned to the star since its LC fits none of the classes established in Section 2.1. The “1” designation means only one class is selected. We have stars with mixed classes for 2, 3, 4, and 5 designations. The first row shows the data predicted by the final classification algorithm, and the second row shows the data assigned by the visual revision. Due to mixed classes being allowed, the numbers on the left side do not add the 2716 stars in our testing sample. On the other hand, the numbers on the right side do.

Additionally, we have systematic errors that still persist in a few LCs, regardless of the quality cut described in Section 2. Also, very few LCs have extreme atypical values that are likely produced by problems in the photometry (e.g., bad pixels, cosmic rays). These changes in magnitude due to extreme outliers should not affect the magnitude distribution features estimations due to their robustness (see Section 3.2.1). However, these outliers could affect the estimation of the period depending features (see Section 3.2.3). This also means that in these stars the classifier’s predictions could also be affected. Particularly, in the testing sample we identify 233 stars presenting at least one of these behaviors and since our training sample does not contemplate these kinds of variability, these stars are removed before computing the recall estimations using a posteriori probability, as shown in Figure 5. Nevertheless, these stars are still reported in Table 7, along with a flag in the eighth column, associated with the undesirable characteristic. Therefore, it is important to remark that these flags are derived from human visual inspection and only indicate those LCs strongly affected in their machine-learning classification by these unwanted characteristics.

For the remaining 2483 stars (hereafter the clean testing sample), about 63.9% are predicted to have characteristics of only one class. For the rest of the testing stars, about 22.8% are accepted by two of the final classifiers simultaneously, meaning that they have two predicted classes, and less than 4.2% are assigned to three or more classes (3.5% for three classes, 0.6% for four, and 0.1% for five). On the other hand, less than 9.1% of the testing sample is rejected by all classifiers, they are labeled No-Class (NC) in Table 7. All of these percentages imply that about one in every four stars is predicted for more than one class. As expected in Section 3.3, the set of final classifiers mostly predicts only one class for the new data, and even though mixed classifications are allowed, these are less common.

Subsequently, after the visual inspection, the proportion of mixed classes is reduced, over 75.4% of the clean testing sample is assigned to a single class, less than 20.3% to two simultaneous classes, only about 3.7% have no class, and under 0.6% have three classes. The fact that after visual inspection mixed classes are still present indicates that several physical phenomena associated with the proposed classes can occur

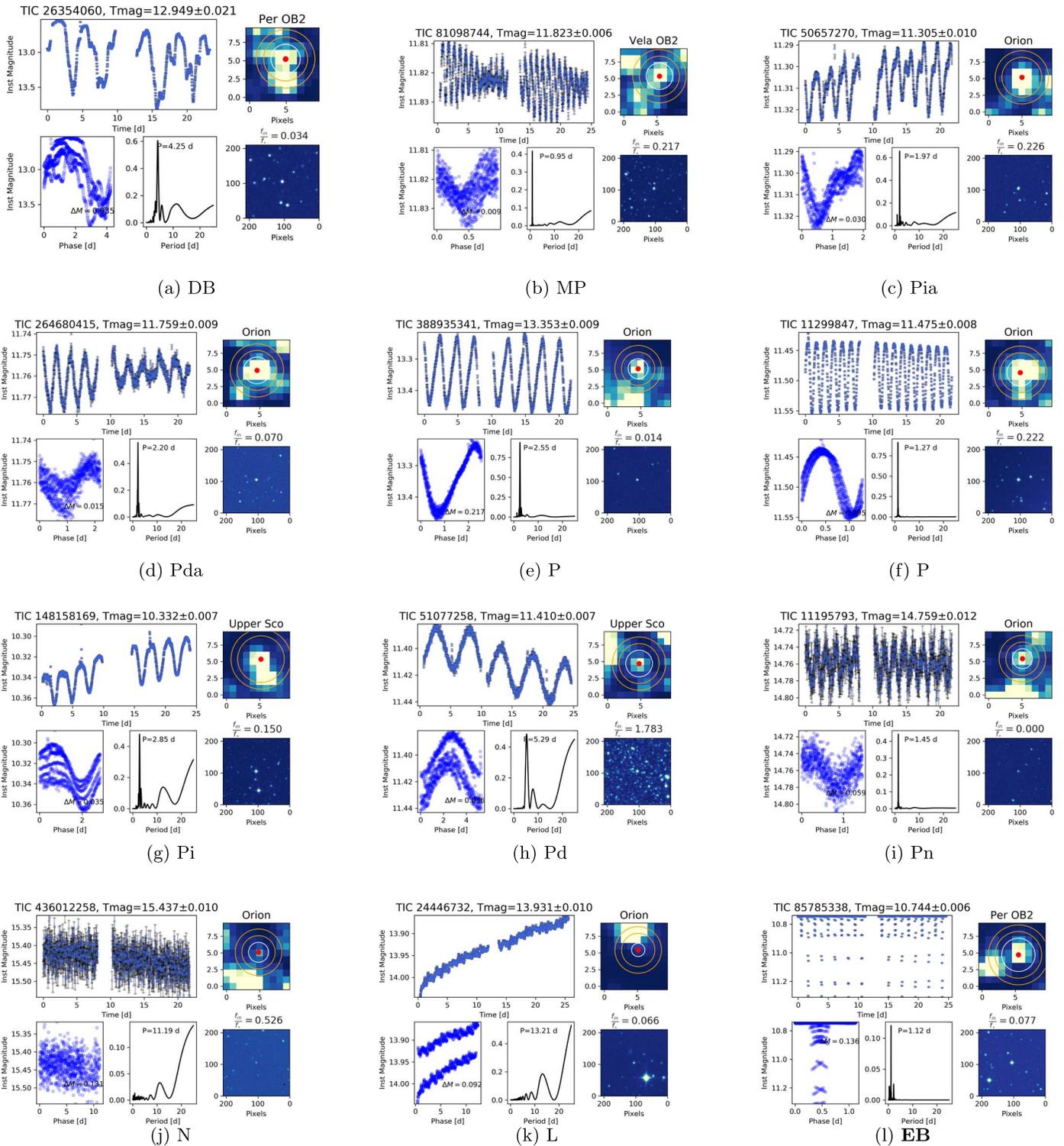


Figure 6. Selected LCs of single-classed stars in the testing sample. The proposed morphology class is made by the final algorithm and confirmed by the visual revision. For each plot from left to right at the top: (A) Light curve in TESS magnitude with a representative error bar. The legend at the top refers to the star identification, the mean TESS magnitude, and the uncertainty. (B) Field of view ($210 \times 210 \text{ arcsec}^2$) corresponding to a TESS image of 10×10 pixels. The white circle shows the photometric aperture, and the orange circles the sky annulus. The red dot marks the centroid of the star. In the bottom panels from left to right: (C) Phase-folded light curve to the estimated best period. ΔM shows the amplitude. (D) Lomb-Scargle periodogram, with the estimated period. (E) 210×210 pixels Digital Sky Survey (DSS2) thumbnail, same field of view as (B). f_{fin}/f_s shows the contamination level estimator (CLE).

simultaneously in a star (approximately one in every five cases). The presence of mixed classes is less probable, showing the adequacy of the proposed classes listed in Section 2. Figure 6 shows LC examples with one predicted class that was

confirmed by the validation process, meaning that it has the same revised class in visual inspection.

The P class final classifier is one of the best-behaved in the testing sample. Also, star candidates proposed by this classifier

are the most common in the clean testing sample, having the most probable single class prediction for 520 stars and also being the most common mixed class predictions: Pd+P, Pda+P, and MP+P with 57, 59, and 60 candidates, respectively. When compared to the visual revision, the P class is the most common morphology in the clean test sample, with 807 stars and 234 of those presenting mixed classes. The two most common combinations are MP+P with 73 stars and Pd+P with 43. Our final algorithm correctly predicts about 74% of the stars that have a P morphology in the testing sample, even predicting correctly mixed morphologies that include the P class.

Figure 5 also shows how a posteriori recall estimation can be heavily influenced by the presence of mixed classes. For example, for the Pi class, its a posteriori recall dramatically decreases when the testing sample includes stars with two mixed classes. The expected percentage for stars with mixed classes in the whole clean testing sample is 21%, but over 77% of the misclassified Pi stars have mixed revised classes. Pi stars with mixed classes are usually mistaken by the final algorithm with other periodic behaviors as P, MP, and Pn or are classified as L having the amplitude of their periodic behavior overshadowed by their mean increasing one.

Less drastic but still considerable decreases in the recall estimations occur for N, Pn, and DB classes, having, respectively, 43.5%, 43.7%, and 54% of stars with mixed classes among misclassified ones. Pn stars with mixed classes are usually not periodic enough to be accepted by the Pn final classifier and are predicted as just N or even NC. Other final classifiers sometimes pick up the periodicity of these Pn stars and are predicted as MP, P, or Pi. Parallely, for some N stars, magnitude variations in their LC are high enough to be rejected by the N final classifiers and even predicted as L. In this case, the revised classes of these stars are usually a mixed class of N + L, implying that for the final algorithm it can be difficult to predict LCs with noisy signal along another nonstochastic behavior. Still, only 18% of N stars have mixed morphologies. Finally, DB stars occasionally have the presence of other morphologies (36%), but in only half of these cases, the final classifier predicts the DB morphology. In two out of five DB stars with mixed classes, the other classifiers identify all of the other present morphologies in the LC. For example, DB+P stars are accepted by the P final classifier whether they were accepted by the DB classifier or not.

4.2. Infrared Excess in P and DB classes

As presented in Section 2.1, the origin of the stellar variability in the DB class is associated with the presence of a protoplanetary disk that produces extinction fluctuations and/or flux contribution from the accretion phenomena. Cody & Hillenbrand (2018) presented a method to recognize these classes using periodicity and asymmetry measurements in the brightness distribution obtained after a flattening process. It is worth mentioning that in this work, we do not apply previous flattening methods to classify the different morphological classes, including the DB class, since it could delete long-term brightness variations that can have a physical origin (L, Pi, Pd). In Figure 7, we show a histogram and a color–magnitude diagram used to select stars with IR excess at $12\ \mu\text{m}$ (Luhman & Mamajek 2012). More than 61% of the DB class exhibit IR excess with $(K - W_3) > 1$. As a reference, we include the P class, in which only less than 13% have IR excess. This analysis is made using 706 stars, derived from cross-matching

our P and DB candidates with the AllWISE data (Cutri et al. 2021), and selecting those with good quality in their photometry, having a profile-fit flux signal-to-noise ratio over 2, and without possible contamination or bias due to proximity to an image artifact. The P class is generally expected in TTS when stellar spots produce periodic decreases in the LC. Since diskless stars do not have the contribution of variable phenomena associated with protoplanetary disks, it is not surprising that their proportion of stars with IR excess is smaller than the one found in DB stars. Even though the morphological classes used by Cody & Hillenbrand (2018) are not exactly the same as we define here, the rate proportions of our DB class with high disk presence are similar. Likewise, periodic associated classes also tend to have smaller proportions of disk presence.

4.3. Period Analysis of Increasing and Decreasing Morphologies

After the classification and visual revision of the morphological classes of the testing sample, we focus on the systematic increasing or decreasing behaviors of the Pi, Pd, and some L stars with periodic characteristics. These behaviors can affect the derivation of reliable periodic estimations. To study this effect, we perform a trend correction by subtracting either a linear or quadratic fit of the LCs. The best fit was chosen using the Bayesian information criterion (Schwarz 1978). From a sample of 528 stars, 190 were corrected by a linear fit, while 338 were by a quadratic one. Periods for these stars are recomputed using the Lomb-Scargle algorithm over the re-scaled LCs. An example of this procedure is shown in Figure 8. A careful inspection of the obtained phased LCs shows that 20 stars are not periodic, while 292 stars have period differences of less than 0.5 day. The remaining 26 stars show significant differences between the periods computed by Lomb-Scargle and Fourier algorithms. The latter was used to perform a more detailed search period of the re-scaled LC. The visual inspection shows that period differences greater than 0.5 day change dramatically the phased LCs. Four main causes explain these observed period differences. (1) The influence of the data transmission window generates a gap in the LC (15 stars). Sometimes, this is interpreted by the LSP algorithm as a false-periodic pattern that repeats about every 10–13 days. (2) The period that gives the best-phased LC does have a peak in the periodogram, but not as strong as the highest Lomb-Scargle peak (four stars). (3) The presence of a more complex trend in the data than the one subtracted by the re-scaling procedure (four stars). (4) The Lomb-Scargle period is an alias of the real period (three stars).

Inquiring more in the first reason, the false periodicity is accentuated when the local variation of magnitude in the LC is small. Additionally to the few Pi and Pd stars affected by the transmission window, our N and L morphologies are affected too. This implies a biased estimation of the period depending features that affect the prediction of the classifiers for these morphologies.

5. Summary and Conclusions

We present a machine-learning application to classify the TESS LCs morphology of TTS candidates in the stellar formation regions OSFC, γ Velorum, IC 348, Upper Scorpius, Corona Australis, and Perseus OB2. Eleven morphological

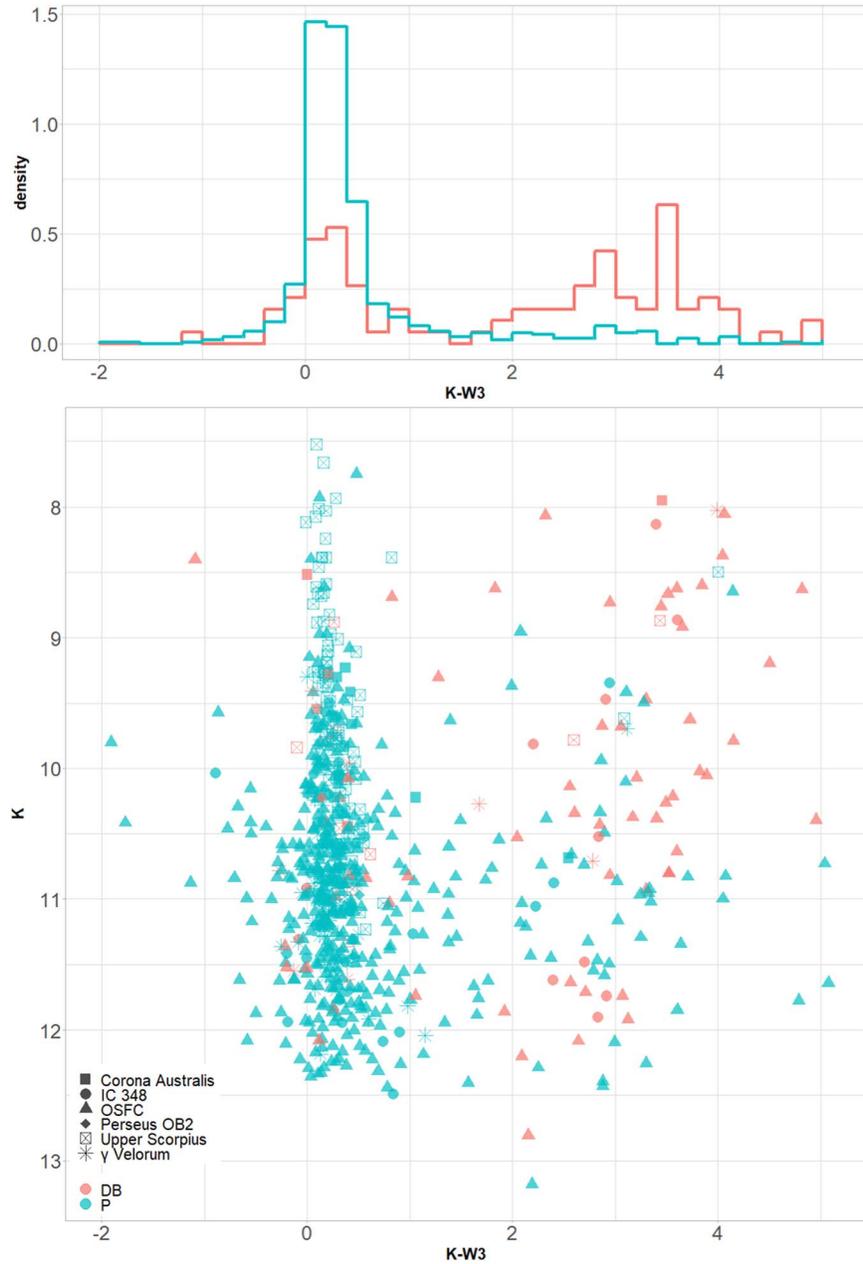


Figure 7. Distribution of 706 candidates to P or DB classes proposed by the algorithm. In the upper panel we have a density histogram of the color of both classes giving an estimation of their probability densities. In the bottom panel we show a color–magnitude diagram using K and W3 bands obtained from the AllWISE catalog (Cutri et al. 2021), and selected to have a profile-fit flux signal-to-noise ratio over 2, and discarding those with possible contamination or bias due to proximity to an image artifact. The number of members in each star-forming region is distributed as follows: 457 stars in OSFC, 112 in Upper Scorpius, 60 in γ Velorum, 49 in IC 348, 19 in Corona Australis, and nine in Perseus OB2. The color–magnitude diagram shows that the behavior of the infrared excess of both classes occurs among all regions we studied.

classes are proposed to represent the LC diversity observed in our data. Supervised KNN, CART, RF, and SVM models are trained in both binary and multiclass approaches, using kappa and recall performance metrics to optimize the hyperparameter grids. The recall is also used to compare the performance of different models, feature spaces, and training samples. The SVM models were usually the best performed, followed closely by RF (Table 4). In the case of our algorithm, this result is associated with the flexibility in the range of the possible values of the hyperparameters in each model. For RF, we tune the size of the subsampled set of features selected in each split of the trees, which gives us 18 discrete possible values to optimize among. In contrast, SVM allows us to propose a wider range of

orders of magnitude for the hyperparameters, which implies a thinner search along the tuning grid.

A final classification algorithm is constructed using the best-performed models per each class, allowing for mixed classes in the final classification (Table 5). Then, we visually revised those classifications. For our complete sample (training and testing), periodic morphologies (P, Pn, Pi, Pd, Pia, Pda, and MP) are quite common ($\sim 64\%$ of the cases), being the P class the most common (Section 2 and Table 6), and its final classifier being one of the best performed (Table 5). In parallel, the DB morphology represents about 8% of the stars in both samples. The infrared excess analysis in Section 4.2 suggests the presence of protoplanetary disks in about 61% of the

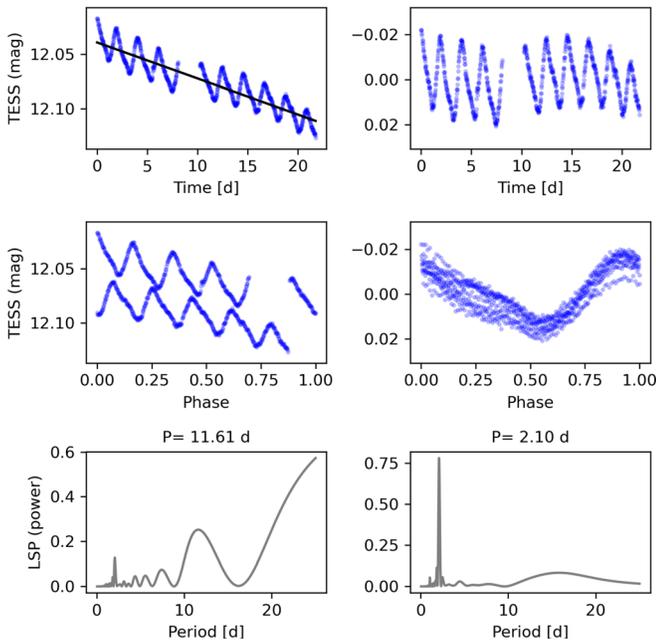


Figure 8. Correction process done to star TIC 279581307. The top-left panel shows the original LC and its linear fit. The top-right panel shows the corrected LC, where the fit is subtracted. The middle panels show their phased curves using the proposed periods found by the LSP, which are shown in the bottom panels.

classified DB stars. Next, the N and L classes, which are usually nonperiodical, are present in 27% of all cases. Finally, the EB morphology has a very low representation in the training and testing samples (below 1%), meaning that the cross-validation and direct estimation of the EB class final classifier’s recalls have high uncertainty. The low representation also affects the number of features that the final classifier uses (Table 5), which is higher compared to the final classifiers for the remaining classes, implying the use of all information possible to characterize the morphology.

The performance of the algorithms is estimated using two approaches: cross-validation process when the optimized hyperparameters are selected; and direct estimation over the clean testing sample, comparing predictions with the classes obtained from our visual inspection. The a posteriori estimations are always lower but usually close to the cross-validation estimations (Figure 5). This result can be understood considering several reasons. First, outlying behaviors in the LCs, not contemplated in the original classes or training sample, are associated with image artifacts or events as flares or sole transits registered in the observations. Second, the sample used to train the models comes from only the OSFC since it is one of the regions with the most confirmed young stellar members (Section 2.1), and the testing sample uses other stellar formation regions, which could change the accuracy of the estimators used to calculate the features in the LCs presented in those regions, especially those related to magnitude. Third, as seen in Figure 5, the a posteriori recall estimations increase closer to the expected value when a sample of only revised single-classed stars is used. So, even though the final classifier allows for mixed classes, the fact that the training sample and the cross-validation estimations used only single-class stars affects the predictions of stars that have a certain combination of morphologies and the estimation of the performance of the whole classifier with new data. Results reported at the end of

Section 4 let us conclude that even though the final algorithm can predict the mixed classes, its performance excels when only one of the defined morphologies is mostly present in the LC of the analyzed data to predict.

Another pattern that must be accounted for, when interpreting the performance of our final algorithm, is the periodic behavior that the LSP could falsely detect. This false periodicity mainly occurs when the LSP mistakes the lack of points in the data transmission window. In most of these cases, the correction described in Section 4.3 permits a better period estimation, which is applied after the prediction by the classification algorithm. Also, we emphasize this trend correction is not meant to obtain new classifications, and is only used to study the effects of global variations in the LCs. On the other hand, the quasiperiodic behavior present among some stars of specific classes (even the P class) must not be confused with the previously mentioned false-periodic behavior. The quasiperiodic magnitude variations of a star mean real physical phenomena that creates a recurrent behavior in time, but the exact value of brightness is not completely repeated in each cycle. The dips in stars within DB class are a good example of this quasiperiodic behavior. The mentioned difference among periodic, false-periodic, and quasiperiodic behaviors must be considered when the periods in the final catalogs presented in this work are interpreted by the reader, since the LSP and therefore our final classification algorithm see no significant difference.

The usual class combinations in both the predictions and the visual revision are a product of *the visual classification limit* between morphologies with common properties. This means that human classification does not give fixed limits to the boundaries between some characteristics that give the difference between two classes. For example, some L stars have a periodic behavior that does not dominate their large-scale variation due to the amplitude of their periodic behavior. This gives a common zone where both L and Pd (or Pi) morphologies can be identified in a certain LC and are difficult to distinguish. A similar situation happens with Pd (or Pi) and P classes, with mild decreasing (or increasing) behavior. On the other hand, Pia, Pda, and even P morphologies can be part of a longer MP behavior (>25 days) and are difficult to discern when the modulation is low, having amplitude variations in the classification limit. Since the training sample is classified by visual revision, the learning algorithm inherits this lack of explicit quantitative values near the common zone among some classes. This is another reason to use our allow-mixed-classes approach and to understand the existence of mixed classes in our samples.

The aforementioned considerations and the distribution of stars with mixed classes or rejected by all classifiers (Table 6) support the idea that several physical phenomena can operate simultaneously on the star implying mixed morphological classes or even preventing a classification of the LCs. This translates into LCs that behave with similar properties that can be classified simultaneously in more than one class and allow flexibility when new data is introduced to be predicted. Still, a main phenomenon can be associated with single-classed LCs. It is also necessary to consider that a mutually exclusive approach using our mixed classes as new and separate classes to develop a different machine-learning algorithm from scratch could overfit our training sample and end up performing poorly with

new data since the unbalance and low representation problems would be dramatic.

Our last considerations are to emphasize the intention of the final algorithm as a tool to hasten and facilitate the investigator's task when identifying candidates for further analysis of the physical processes connected to the photometric variability of these stars. The supervised machine-learning algorithms used here are destined to optimize time in scientific investigation but inherited human bias in the training sample can be present and must be considered when results are interpreted. Also, we must reiterate that our final algorithm systematizes the classification process among the declared classes and is not designed to methodically detect flares, image artifacts, or systematic errors present in the LCs. Finally, the selection of our samples in stellar formation regions, along with the high presence of the periodic morphologies, allows for presenting a catalog of TTS candidates with stellar spots on their surface and with rotational period estimations.

Acknowledgments

B.R.-F. acknowledges financial support from the Fondo de Investigaciones de la Facultad de Ciencias de la Universidad de los Andes, Colombia, through financiación de proyectos de investigación y presentación de resultados en eventos académicos with project código INV-2019-86-1852 and doctoral fellowship support from the Departamento de Física de la Universidad de los Andes, Colombia, in form of asistencia graduada doctoral docencia. A.G.-V. acknowledges financial support given by Fondo de Investigaciones de la Facultad de Ciencias de la Universidad de los Andes, Colombia, through programa de investigación código INV-2023-162-2853. The authors would like to thank the Vice Presidency of Research & Creation's Publication Fund at Universidad de los Andes for its financial support. J.H. and J.S. acknowledge support from the National Research Council of México (CONACyT) project No. 86372, PAPIIT-UNAM projects IA102921 and CG101723, and fellowship support in the Posgrado en Astrofísica graduate program at Instituto de Astronomía, UNAM. An anonymous referee provided many insightful comments that improved the content and presentation of this work. We acknowledge the collaboration group ARYSO (Activity and Rotation in Young Stellar Objects) for the helpful suggestions for this work. This work made use of Python and R open source software: Python software foundation <https://www.python.org/>; The R Project for Statistical Computing <https://www.r-project.org/>. This paper includes data collected with the TESS mission, obtained from the MAST data archive at the Space Telescope Science Institute (STScI). Funding for the TESS mission is provided by the NASA Explorer Program. This publication makes use of data products from the Wide-field Infrared Survey Explorer, which is a joint project of the University of California, Los Angeles, and the Jet Propulsion Laboratory/California Institute of Technology, and NEOWISE, which is a project of the Jet Propulsion Laboratory/California Institute of Technology. WISE and NEOWISE are funded by the National Aeronautics and Space Administration. This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France.

Facilities: TESS (Ricker et al. 2014), AllWISE (Cutri et al. 2021).

Software: TESSextractor (Serna et al. 2021), Astrocut (Brasseur et al. 2019), Astropy (Astropy Collaboration et al. 2013), Caret (Kuhn et al. 2022), FastDTW (Salvador &

Chan 2007), GammaPy (Nigro et al. 2019), kernlab (Karatzoglou et al. 2022), Matplotlib (Hunter 2007), Numpy (Harris et al. 2020), Photutils (Bradley et al. 2019), Sklearn (Pedregosa et al. 2011), rpart (Therneau et al. 2022), randomForest (Liaw et al. 2022), TopCat (Taylor 2005).

Appendix A Machine-learning Models

A.1. KNN

K-nearest neighbors were proposed by Fix & Hodges (1951) in a technical report about discriminatory analysis for the United States Air Force, but were not actually published. The importance and relevance of their work was recognized and commented almost four decades later by Silverman & Jones (1989). The idea to construct their decision rule is quite simple: all new data in the feature space will be classified depending on the predominant class of the closest training data. The number of near neighbors considered (k) is used as a hyperparameter. Low values of k tend to overfit the training sample, and high values usually develop an overly generalized model. For the R implementation, *caret* (Kuhn et al. 2022) does not need additional libraries to develop KNN classifiers, while for the Python implementation we use *sklearn.neighbors.KNeighborsClassifier* (Pedregosa et al. 2011). The grid used for selecting optimal values of k passes through all values of $k = 1, 2, \dots, 300$ neighbors for binary classifiers, and $k = 1, 2, \dots, 200$ for multiclass classifiers. KNN is a memory-based method and is not required to fit a model per se, so given the size of our training sample, they take the least computational time to be trained, among all models considered in this work. Further and extensive analysis of KNN model, is done by Devroye et al. (1996).

A.2. CART

Classification and regression trees were developed by Brieman et al. (1984). A tree is generated by successive partitions of the training data. Each partition, called a node (t), is created optimizing the *purity* of the subset that contains. This means that among all possible divisions (called splits) of the form $\{\bar{x}: x_j \leq a, a \in \mathbb{R}, j \leq C\}$ in the feature space, the one with higher ratios of the presence of a single class is selected, where C is the total of classes. This is implemented using a *impurity function* ($i(t)$) that given the proportions of each class in the node (p_j), interpreted as conditional probabilities, will be minimized to select the subsequent partition in the node. The two typical impurity functions are the one using the Gini index $i(t) = \sum_{j=1}^C p_j(1 - p_j)$ and that using the information index $i(t) = \sum_{j=1}^C -p_j \log(1 - p_j)$. The *decrease of impurity* given a split in a node can be calculated by subtracting the impurities of the original node and the impurities of the subsequential nodes. The split that optimizes the decrease is selected.

Given a specific tree, the cost of continuing the splitting is calculated using a complexity cost function that depends on the complexity of the tree (number of terminal nodes), the risk of the tree (the misclassification rate), and a complexity parameter cp that weights the value of the complexity. Given a certain value for cp , there is a certain tree that will optimize the complexity cost function. So, the complexity parameter is used as the hyperparameter to be tuned using cross-validation. High values of cp will *prune* too much of the tree, meaning less and fewer nodes, i.e., an overgeneralized model. Low values of cp

will permit more terminal nodes at the cost of having less data in each one, having the model overfit the training sample.

Also, the impurity function is useful when calculating the usefulness of a specific feature. Given the optimal splits that a certain feature can have when splitting each of the nodes of a tree, the sum of the decreases of impurity in those splits is used as an *importance* measure that allows us to compare all of the features, and rank their relevance to the classification problem we are trying to solve. For the R implementation, *caret* will need the *rpart* package (Therneau et al. 2022). For the Python implementation, we use *sklearn.tree.DecisionTreeClassifier* (Pedregosa et al. 2011). We used a succession of 510 values of different orders of magnitude (10^{-4} , 10^{-3} , 10^{-2} , and 10^{-1}) as a grid to optimize the value of *cp*.

A.3. RF

The algorithm of Random Forests was mainly developed by Breiman (1999, 2001). It uses a modification of the bootstrap aggregation (bagging) technique that obtains a set of small classifiers that use a random subset of the training sample. The decision rule is obtained by the voting of the small classifiers. In the specific case of RF, the small classifiers are classification trees, and their splitting algorithm is modified to select partitions among only a subset of the original features. An advantage of RF is that adding more trees does not overfit the model since the method has a limit value of the generalizing error (Breiman 2001). The number of trees and the size of the features subset used in each split (*mtry*) are the common hyperparameters to be tuned. On the other hand, mean values of the importance of a feature for all trees are used as an importance measure for the whole RF. For the R implementation, the *caret* package needs the *randomForest* package (Liaw et al. 2022), and for the Python implementation we use *sklearn.ensemble.RandomForestClassifier* (Pedregosa et al. 2011).

Both implementations use a fixed value of 500 trees, and all possible values for the features subset, i.e., *mtry* = 1, 2, ..., 18.

A.4. SVM

Support Vector Machines were developed by Vapnik (1999). The basic idea behind SVM is that a transformation of the data in the feature space can result in a transformed space where the data can be separated using a hyperplane. The derived minimization problem can be tackled using the generalization of the Lagrange optimization proposed by Kuhn & Tucker (1951). Interestingly, the solution optimization problem does not depend explicitly on the transformation, but on the inner product in the transformed space of the points near the division frontier (the so-called *support vectors*). This inner product can be interpreted as a *kernel function*, whose parameters modify the transformation, without knowing it explicitly. For the function to be minimized, a sum of the margins (distances to the decision frontier) of the misclassified points is added. The factor multiplying this sum is called the cost (*C*), and it is another hyperparameter to be tuned. For the R implementation, the *caret* package needs the *kernelab* package (Karatzoglou et al. 2022). As for the Python implementation, we use *sklearn.svm.SVC* (Pedregosa et al. 2011). We used the radial basis kernel function $K(\vec{x}, \vec{y}) = \exp(-\gamma\|\vec{x} - \vec{y}\|^2)$, which have the inverse kernel width (γ) as hyperparameter. We used a grid of hyperparameters of each combination of γ and *C*, where γ varies among a logarithmic succession of values between 10^{-8} and 10^{-1} , and *C* among a logarithmic succession between 10 and 10^{12} .

Appendix B Star Catalog

Table 7 shows a portion of the complete catalog of our training and testing samples (Section 2.1).

Table 7

Catalog of Both the Training and Testing Samples Including TIC ID (First Column), TESS Magnitude (Second Column), Equatorial Coordinates (J2000, Third and Fourth Columns), and Estimated Period (Fifth Column)

TIC ID	Tmag	R.A. (deg)	Decl. (deg)	Per (day)	PRED	INSPEC	Flag	Sample	CLE
4352899	10.996	81.412553	-5.934374	1.573		Pda		training	0.043
11296154	15.014	84.860307	-2.437643	4.92		Pn		training	0.302
11453668	14.404	85.460613	-2.049187			DB		training	0
24697436	10.751	81.185515	-9.237376	13.561	DB+N+EB	Pd+MP	^e	testing	0.005
34470764	10.871	82.547898	-6.569161		DB+EB	NC		testing	0.299
50622261	10.998	82.285844	-5.974223	1.598		Pia		training	0.019
50657928	12.635	82.457949	-1.459174	6.294		P		training	0.026
50748110	12.798	83.159774	-4.514691	0.749		MP		training	0.028
50792805	14.251	83.284454	-5.322373	1.673		Pi		training	0
72829000	13.619	85.845115	-2.269161			EB		training	0.034
101133244	12.149	54.946623	34.491997		DB+EB	NC	^a	testing	0.285
101134050	14.692	54.914699	33.268537	2.897	NC	Pn	^d	testing	0.147
127860350	13.766	218.168808	-46.168504	3.047	Pia+Pn	Pn	^c	testing	0.671
164304284	14.108	191.629685	-50.907928	1.199	MP	Pi	^{a, b}	testing	1.344
176449381	12.907	86.695176	-1.68259	7.493	Pn	Pn		testing	0.028
249040852	15.309	79.32379	-0.401759			N		training	0.09
249070479	14.805	79.48498	-4.506944	1.698		Pd		training	0.212
279581307	12.077	85.619125	-9.607878	2.101858	Pd	Pd+L	^f	testing	0.003
388918109	14.748	83.948536	-6.360037	2.872	DB+Pd	DB+Pd		testing	0
436013242	10.179	83.702369	9.515849			L		training	0.132

Notes. For the testing sample, predictions given by the final classification algorithm (sixth column), are compared to the final classification revised by the visual revision (seventh column). On the other hand, for the training sample, their assigned class by visual inspection is in the seventh column. Note that even in this human revision, we agree some light curves do have patterns associated with more than one of the initially established classes. Also, some extra tags (eighth column) were occasionally added describing problems in the initial photometry that affect the criteria of the algorithm. Consider that these flags are not products of the algorithm, and appeared only in the visual inspection. The ninth column specifies the sample in which the star was used. The CLE described in Section 2 is included in the tenth column. Table 7 is published in its entirety in the machine-readable format. A portion is shown here for guidance regarding its form and content.

^a Persisting systematic errors dominating the LC.

^b Atypical extreme magnitude values not consistent with the behavior of the rest of data in the LC.

^c Single transit eclipse in the observational window.

^d Isolated flare in the LC.

^e Very short period star not detected by the LSP.

^f Revised period obtained from the process described in Section 4.3.

(This table is available in its entirety in machine-readable form.)

ORCID iDs

Astaroth Elizabethson  <https://orcid.org/0000-0002-2249-8474>

Javier Serna  <https://orcid.org/0000-0001-7351-6540>

Alejandro García-Varela  <https://orcid.org/0000-0001-8351-0628>

Jesús Hernández  <https://orcid.org/0000-0001-9797-5661>

Juan F. Cabrera-García  <https://orcid.org/0000-0002-0020-843X>

References

Ansdell, M., Gaidos, E., Hedges, C., et al. 2019, *MNRAS*, **492**, 572
 Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., et al. 2013, *A&A*, **558**, A33
 Auvergne, M., Bodin, P., Boisnard, L., et al. 2009, *A&A*, **506**, 411
 Battley, M. P., Armstrong, D. J., & Pollacco, D. 2022, *MNRAS*, **511**, 4285
 Borucki, W. J., Koch, D., Basri, G., et al. 2010, *Sci*, **327**, 977
 Bouvier, J., Alencar, S. H. P., Boutelier, T., et al. 2007, *A&A*, **463**, 1017
 Bradley, L., Sipőcz, B., Robitaille, T., et al. 2019, *astropy/photutils*: v0.6, Zenodo, doi:10.5281/zenodo.2533376
 Brasseur, C. E., Phillip, C., Fleming, S. W., Mullally, S. E., & White, R. L. 2019, *Astrocot*: Tools for Creating Cutouts of TESS Images, Astrophysics Source Code Library, ascl:1905.007
 Breiman, L. 2001, *Mach. Learn.*, **45**, 5

Breiman, L. 1999, *Random Forests Random Features*, Technical Report 567, Statistics Department, University of California, <https://www.stat.berkeley.edu/~breiman/random-forests.pdf>
 Briceño, C., Calvet, N., Hernández, J., et al. 2005, *AJ*, **129**, 907
 Briceño, C., Calvet, N., Hernández, J., et al. 2019, *AJ*, **157**, 85
 Brieman, L., Friedman, J., Olshen, R., & Stones, C. 1984, *Classification and Regression Trees* (1st edn.; New York: Routledge)
 Brys, G., Hubert, M., & Struyf, A. 2004, *JCGS*, **13**, 996
 Brys, G., Hubert, M., & Struyf, A. 2006, *CSDA*, **50**, 733
 Cantat-Gaudin, T., Jordi, C., Vallenari, A., et al. 2018, *A&A*, **618**, A93
 Cassisi, C., Montalto, P., Aliotta, M., Cannata, A., & Pulvirenti, A. 2012, *Advances in Data Mining Knowledge Discovery and Applications* (London: Intech Open)
 Claytor, Z. R., van Saders, J. L., Llama, J., et al. 2022, *ApJ*, **927**, 219
 Cody, A. M., & Hillenbrand, L. A. 2018, *AJ*, **156**, 71
 Cody, A. M., Hillenbrand, L. A., David, T. J., et al. 2017, *ApJ*, **836**, 41
 Cody, A. M., Stauffer, J., Baglin, A., et al. 2014, *AJ*, **147**, 82
 Cohen, J. 1960, *Educ. Psychol. Meas.*, **20**, 37
 Cutri, R. M., Wright, E. L., Conrow, T., et al. 2021, *yCat*, **II/328**
 Davenport, J. R. A. 2016, *ApJ*, **829**, 23
 Devroye, L., Györfi, L., & Lugosi, G. 1996, *A Probabilistic Theory of Pattern Recognition* (Berlin: Springer)
 Espallat, C. C., Robinson, C. E., Romanova, M. M., et al. 2021, *Natur*, **597**, 41
 Feinstein, A. D., Montet, B. T., Ansdell, M., et al. 2020, *AJ*, **160**, 219
 Fix, E., & Hodges, J. L. 1951, *Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties*, Technical Report, 4 21-49-004, USAF School of Aviation Medicine
 Fritzewski, D. J., Kitzé, M., Mugrauer, M., et al. 2016, *MNRAS*, **462**, 2396
 Gabruseva, T., Zlobin, S., & Wang, P. 2020, *JAI*, **9**, 2050005
 Giorgino, T. 2009, *J. Stat. Softw.*, **31**, 1

- Günther, M. N., Zhan, Z., Seager, S., et al. 2020, *AJ*, **159**, 60
- Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, *Natur*, **585**, 357
- Hartmann, L., Herczeg, G., & Calvet, N. 2016, *ARA&A*, **54**, 135
- Hastie, T., Tibshirani, R., & Friedman, J. 2009, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (Berlin: Springer), <https://books.google.com.mx/books?id=eBSgoAECAAJ>
- Herbig, G. H., & Bell, K. R. 1988, *Third Catalog of Emission-line Stars of the Orion Population* (Santa Cruz, CA: Lick Observatory)
- Herbst, W., Herbst, D. K., Grossman, E. J., & Weinstein, D. 1994, *AJ*, **108**, 1906
- Hernández, J., Calvet, N., Perez, A., et al. 2014, *ApJ*, **794**, 36
- Hinton, P. C., France, K., Batista, M. G., et al. 2022, *ApJ*, **939**, 82
- Hon, M., Huber, D., Kuzlewicz, J. S., et al. 2021, *ApJ*, **919**, 131
- Huber, P. J. 1964, *Ann. Math. Stat.*, **35**, 73
- Huber, P. J., & Ronchetti, E. M. 2009, *Robust Statistics* (2nd edn.; Hoboken, NJ: Wiley)
- Hunter, J. D. 2007, *CSE*, **9**, 90
- Ismailov, N. 2005, *AREP*, **49**, 309
- Karatzoglou, A., Smola, A., Hornik, K., et al., 2022 *Kernel-Based Machine Learning Lab*, <https://CRAN.R-project.org/package=kernlab>
- Kounkel, M., & Covey, K. 2019, *AJ*, **158**, 122
- Kounkel, M., Covey, K., Suárez, G., et al. 2018, *AJ*, **156**, 84
- Kounkel, M., Covey, K. R., Stassun, K. G., et al. 2021, *AJ*, **162**, 184
- Kuhn, H. W., & Tucker, A. W. 1951, *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability* (Berkeley, CA: Univ. California Press), 481
- Kuhn, M., Wing, J., Weston, S., et al., 2022 *caret: Classification and Regression Training*, <https://CRAN.R-project.org/package=caret>
- Kuhn, M. A., Hillenbrand, L. A., Sills, A., Feigelson, E. D., & Getman, K. V. 2019, *ApJ*, **870**, 32
- León-Figueroa, B. 2020, Bachelor's thesis, Universidad de los Andes, <http://hdl.handle.net/1992/51268>
- Liaw, A., Wiener, M., Breiman, L., & Cutler, A., 2022 *Breiman and Cutler's Random Forests for Classification and Regression*, <https://CRAN.R-project.org/package=randomForest>
- Lomb, N. R. 1976, *Ap&SS*, **39**, 447
- Luhman, K. L., & Mamajek, E. E. 2012, *ApJ*, **758**, 31
- Morales-Calderón, M., Stauffer, J. R., Hillenbrand, L. A., et al. 2011, *ApJ*, **733**, 50
- Murphy, S. J., Joyce, M., Bedding, T. R., White, T. R., & Kama, M. 2021, *MNRAS*, **502**, 1633
- Naul, B., Bloom, J. S., Pérez, F., & van der Walt, S. 2018, *NatAs*, **2**, 151
- Nigro, C., Deil, C., Zanin, R., et al. 2019, *A&A*, **625**, A10
- Ofman, L., Averbuch, A., Shlisselberg, A., et al. 2022, *NewA*, **91**, 101693
- Olmschenk, G., Ishitani Silva, S., Rau, G., et al. 2021, *AJ*, **161**, 273
- Pashchenko, I. N., Sokolovsky, K. V., & Gavras, P. 2017, *MNRAS*, **475**, 2326
- Pearson, K. A. 2019, *AJ*, **158**, 243
- Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, *JMLR*, **12**, 2825
- Pérez-Ortiz, M. F., García-Varela, A., Quiroz, A. J., Sabogal, B. E., & Hernández, J. 2017, *A&A*, **605**, A123
- Rao, S., Mahabal, A., Rao, N., & Raghavendra, C. 2021, *MNRAS*, **502**, 2845
- Rebull, L. M., Cody, A. M., Covey, K. R., et al. 2014, *AJ*, **148**, 92
- Rebull, L. M., Stauffer, J. R., Bouvier, J., et al. 2016, *AJ*, **152**, 114
- Ricker, G. R., Winn, J. N., Vanderspek, R., et al. 2014, *Proc. SPIE*, **9143**, 914320
- Salvador, S., & Chan, P. 2007, *Intell. Data Anal.*, **11**, 561
- Scargle, J. D. 1982, *ApJ*, **263**, 835
- Schwarz, G. 1978, *AnSta*, **6**, 461
- Silverman, B. W., & Jones, M. C. 1989, *ISR*, **57**, 233
- Serna, J., Hernandez, J., Kounkel, M., et al. 2021, *ApJ*, **923**, 177
- Sokolovsky, K., & Lebedev, A. 2018, *A&C*, **22**, 28
- Sousa, A. P., Alencar, S. H. P., Bouvier, J., et al. 2016, *A&A*, **586**, A47
- Stassun, K. G. 2019, *yCat*, **IV/38**
- Stauffer, J., Cody, A. M., Baglin, A., et al. 2014, *AJ*, **147**, 83
- Still, M., & Barclay, T. 2012, *PyKE: Reduction and analysis of Kepler Simple Aperture Photometry data*, *Astrophysics Source Code Library*, [ascl:1208.004](https://arxiv.org/abs/1208.004)
- Sun, Y., Wong, A. K., & Kamel, M. S. 2009, *Int. J. Pattern Recognit. Artif.*, **23**, 687
- Tajiri, T., Kawahara, H., Aizawa, M., et al. 2020, *ApJS*, **251**, 18
- Taylor, M. B. 2005, in *ASP Conf. Ser. 347, Astronomical Data Analysis Software and Systems XIV*, ed. P. Shopbell, M. Britton, & R. Ebert (San Francisco, CA: ASP), 29
- Therneau, T., Atkinson, B., & Ripley, B., 2022 *rpart: Recursive Partitioning and Regression Trees*, <https://CRAN.R-project.org/package=rpart>
- Vapnik, V. 1999, *The Nature of Statistical Learning Theory* (2nd edn.; New York: Springer)
- Venuti, L., Bouvier, J., Cody, A. M., et al. 2017, *A&A*, **599**, A23
- Venuti, L., Cody, A. M., Rebull, L. M., et al. 2021, *AJ*, **162**, 101
- Vida, K., Bódi, A., Szklenár, T., & Seli, B. 2021, *A&A*, **652**, A107
- Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, *NatMe*, **17**, 261
- von Neumann, J. 1941, *Ann. Math. Stat.*, **12**, 367
- Yu, L., Vanderburg, A., Huang, C., et al. 2019, *AJ*, **158**, 25
- Zari, E., Brown, A. G. A., & de Zeeuw, P. T. 2019, *A&A*, **628**, A123
- Zhang, K., & Bloom, J. S. 2021, *MNRAS*, **505**, 515