arXiv:2203.08125v1 [astro-ph.SR] 15 Mar 2022

A machine-learning photometric classifier for massive stars in nearby galaxies

I. The method

Grigoris Maravelias[1,2], Alceste Z. Bonanos[1], Frank Tramper[1,3], Stephan de Wit[1], Ming Yang[1], and Paolo Bonfini[2,4,5]

[1] IAASARS, National Observatory of Athens, GR-15236, Penteli, Greece
e-mail: maravelias@noa.gr
[2] Institute of Astrophysics, FORTH, GR-71110, Heraklion, Greece
[3] Institute of Astronomy, KU Leuven, Celestijnenlaan 200D, 3001, Leuven, Belgium
[4] Computer Science Department, University of Crete, GR-71003, Heraklion, Greece
[5] Carrera Group, Jacksonville Beach, Florida, USA

Received XX XX, XXXX; accepted XX XX, XXXX

ABSTRACT

Context. Mass loss is a key parameter in the evolution of massive stars. Despite the recent progress in the theoretical understanding of how stars lose mass, discrepancies between theory and observations still hold. Moreover, episodic mass loss in evolved massive stars is not included in the models while the importance of its role in the evolution of massive stars is currently undetermined.
Aims. A major hindrance to determining the role of episodic mass loss is the lack of large samples of classified stars. Given the recent availability of extensive photometric catalogs from various surveys spanning a range of metallicity environments, we aim to remedy the situation by applying machine learning techniques to these catalogs.
Methods. We compiled a large catalog of known massive stars in M31 and M33 using IR (*Spitzer*) and optical (Pan-STARRS) photometry, as well as *Gaia* astrometric information which helps with foreground source detection. We grouped them in 7 classes (Blue, Red, Yellow, B[e] supergiants, Luminous Blue Variables, Wolf-Rayet, and outliers, e.g. QSOs and background galaxies). As this training set is highly imbalanced, we implemented synthetic data generation to populate the underrepresented classes and improve separation by undersampling the majority class. We built an ensemble classifier utilizing color indices as features. The probabilities from three machine-learning algorithms (Support Vector Classification, Random Forests, Multi-layer Perceptron) were combined to obtain the final classification.
Results. The overall weighted balanced accuracy of the classifier is $\sim 83\%$. Red supergiants are always recovered at $\sim 94\%$. Blue and Yellow supergiants, B[e] supergiants, and background galaxies achieve $\sim 50-80\%$. Wolf-Rayet sources are detected at $\sim 45\%$ while Luminous Blue Variables are recovered at $\sim 30\%$ from one method mainly. This is primarily due to the small sample sizes of these classes. In addition, the mixing of spectral types, as there are no strict boundaries in the features space (color indices) between those classes, complicates the classification. In an independent application of the classifier to other galaxies (IC 1613, WLM, Sextans A) we obtained an overall accuracy of $\sim 70\%$. This discrepancy is attributed to the different metallicity and extinction effects of their host galaxies. Motivated by the presence of missing values we investigated the impact of missing data imputation using simple replacement with mean values and an iterative imputor, which proved to be more capable. We also investigated the feature importance to find that $r-i$ and $y-[3.6]$ were the most important, although different classes are sensitive to different features (with potential improvement with additional features).
Conclusions. The prediction capability of the classifier is limited by the available number of sources per class (which corresponds to the sampling of their feature space), reflecting the rarity of these objects and the possible physical links between these massive star phases. Our methodology is also efficient in correctly classifying sources with missing data, as well as at lower metallicities (with some accuracy loss), making it an excellent tool for accentuating interesting objects and prioritizing targets for observations.

Key words. Stars: massive – Stars: mass-loss – Stars: evolution – Galaxies: individual: WLM, M31, IC 1613, M33, Sextans A – Methods: statistical

1. Introduction

Although rare, massive stars ($M_{*} > 8-10M_{\odot}$) play a crucial role in multiple astrophysical domains in the Universe. Throughout their life they continuously lose mass via strong stellar winds that transfer energy and momentum to the interstellar medium. As the main engines of nucleosynthesis, they produce a series of elements and shed chemically processed material as they evolve through various phases of intense mass loss. And they do not simply die, they explode as spectacular supernovae enhancing significantly the galactic environment of their host galaxies. Their end products (neutron stars, black holes) offer the opportunity to study extreme physics (gravity, temperature), as well as gamma-ray bursts and gravitational wave sources. As they are very luminous they can be observed in more distant galaxies which makes them the ideal tool to understand stellar evolution across cosmological time, especially for interpreting observations from the first galaxies (such as those to be obtained from *James Webb Space Telescope*).

While the role of different stellar populations on galaxy evolution has been thoroughly investigated in the literature (Bruzual

& Charlot 2003; Maraston 2005), a key ingredient of their models, the evolution of massive stars beyond the main-sequence, is still uncertain (Martins & Palacios 2013; Peters & Hirschi 2013). Apart from the initial mass, the main factors determining the evolution and final stages of a single massive star are metallicity, stellar rotation, and mass-loss (Ekström et al. 2012; Georgy et al. 2013; Smith 2014). Additionally, the presence of a companion (common among massive stars, with binary fractions of $\sim 50-70\%$; Sana et al. 2012, 2013; Dunstall et al. 2015) can significantly alter the evolution of a star through strong interactions (de Mink et al. 2014; Eldridge et al. 2017). Although, all these factors critically determine the future evolution and the final outcome of the star, they are, in many cases, not well-constrained.

In particular, mass loss is of paramount importance as it determines not only the stellar evolution but the enrichment and the formation of the immediate circumstellar environment (for a review see Smith 2014 and references therein). Especially in the case of single stars their strong radiation-driven winds during the main-sequence phase remove material continuously but not necessarily in a homogeneous way due to clumping (Owocki & Puls 1999). On top of that, there are various transition phases in the stellar evolution of massive stars where they experience episodic activity and outbursts, such as Wolf-Rayet stars (WRs), Luminous Blue Variables (LBVs), Blue Supergiants (BSGs), B[e] Supergiants (B[e]SGs), Red Supergiants (RSGs), Yellow Supergiants (YSGs). This contributes to the formation of complex structures (such as shells and bipolar nebulae in WR and LBVs, Gvaramadze et al. 2010; Wachter et al. 2010; disks in B[e]SGs, Maravelias et al. 2018). But how important the episodic mass loss is, how it depends on the metallicity (in different galaxies), and what links between the different evolutionary phases exists, are still open questions.

To address these questions the ERC-funded project ASSESS[1] (*"Episodic Mass Loss in Evolved Massive stars: Key to Understanding the Explosive Early Universe"*) aims to determine the role of episodic mass by: a. assembling a large sample of evolved massive stars in a selected number of nearby galaxies at a range of metallicities through multi-wavelength photometry, b. performing follow-up spectroscopic observations on candidates to validate their nature and extract stellar parameters, c. testing the observations against the assumptions and predictions of the stellar evolution models. In this paper we present our approach for the first step, which is to develop an automated classifier based on multi-wavelength photometry.

One major complication for this work is the lack of a sufficiently large number of massive stars with known spectral types. Some of these are rare, which makes the identification of new sources even more difficult to find in nearby galaxies. Moreover, spectroscopic observations at these distances are challenging due to the time and large telescopes required. On the other hand, photometric observations can provide information for thousands of stars, but at the cost of a much lower (spectral) resolution, leading to coarser spectral-type classification (e.g. Massey et al. 2006; Bonanos et al. 2009, 2010; Yang et al. 2019). Using the Hertzsprung–Russell Diagram (HRD) and color-color diagrams one needs a detailed and careful approach to properly determine the boundaries between the different populations and identify new objects (a process which is not free from contaminants, e.g. Yang et al. 2019).

To circumvent this problem we can use a data-driven approach. In this case, data can be fed to more sophisticated algorithms that are capable to "learn" from the data and find the (mathematical) relations that best separate the different classes. These machine-learning methods have been extremely successful in various problems in Astronomy (see Section 3.1). Still though, applications of these techniques tailored for classification of massive stars with photometry are, to the best of our knowledge, scarce if not almost non-existent. Morello et al. (2018) studied the *k*-nearest neighbors on IR colors to select Wolf-Rayet stars from other classes, while Dorn-Wallenstein et al. (2021) explored other techniques to obtain a wider classification based on *Gaia* and IR colors for a large number of Galactic objects. This work provides an additional tool focusing on massive stars in nearby galaxies. It presents the development of a photometric classifier, which will be used in a future work to provide the classification for thousands of, previously unclassified, sources.

In Section 2 we present the construction of our training sample (spectral types, foreground removal, and photometric data). In Section 3 we provide a quick summary of the methods used, we describe the class and feature selection, as well as the implementation and the optimization of the algorithms. In Section 4 we show the performance of our classifier for the M31 and M33 galaxies (on which it was trained on), and the application to an independent set of galaxies (IC 1613, WLM, Sextans A). In Section 5 we discuss the necessity of a good training sample and labels, as well as the feature sensitivity. Finally, in Section 6 we summarize and conclude our work.

2. Building the training sample

In the following section we describe the steps we followed to crete our training sample, starting for the available IR photometric catalogs, removing foreground sources using *Gaia* astrometric information, and collecting spectral types from the literature.

2.1. Surveys used

IR bands are ideal probes for distinguishing massive stars and particular those with dusty environments (Bonanos et al. 2009, 2010). The use of IR colors is a successful method for target selection, as demonstrated by Britavskiy et al. 2014, 2015. We based our catalog composition on mid-IR photometry (3.6 μm, 4.5 μm, 5.8 μm, 8.0 μm, 24 μm), using pre-compiled point-source catalogs from the *Spitzer* Space Telescope (Khan et al. 2015; Khan 2017; Williams & Bonanos 2016), which have only recently become publicly available. This allows us to use positions derived from a single instrument, a necessity for cross-matching since spectral typing comes from various works and instruments. The cross-match radius applied in all cases was 1", since this corresponds to a significant physical separation that grows gradually as a function of distance. Additionally, we only kept sources with single matches, as it is impossible to choose the correct match to the *Spitzer* source when two or more candidates exist within the search radius (accounting for about 2-3% of all sources in M31 and M33).

Although the inclusion of near-IR data would help better sampling of the spectral energy distribution of our sources, this is currently impossible given the shallowness of 2MASS (for our target galaxies), and the - unfortunate - lack of any other public all sky near-IR survey. Some data (for a particular band only; $J_{\rm UK}$) have been collected from the UKIRT Hemisphere Survey[2] (Dye et al. 2018).

[1] `https://assess.astro.noa.gr/`

[2] `http://wsa.roe.ac.uk/uhsDR1.html`

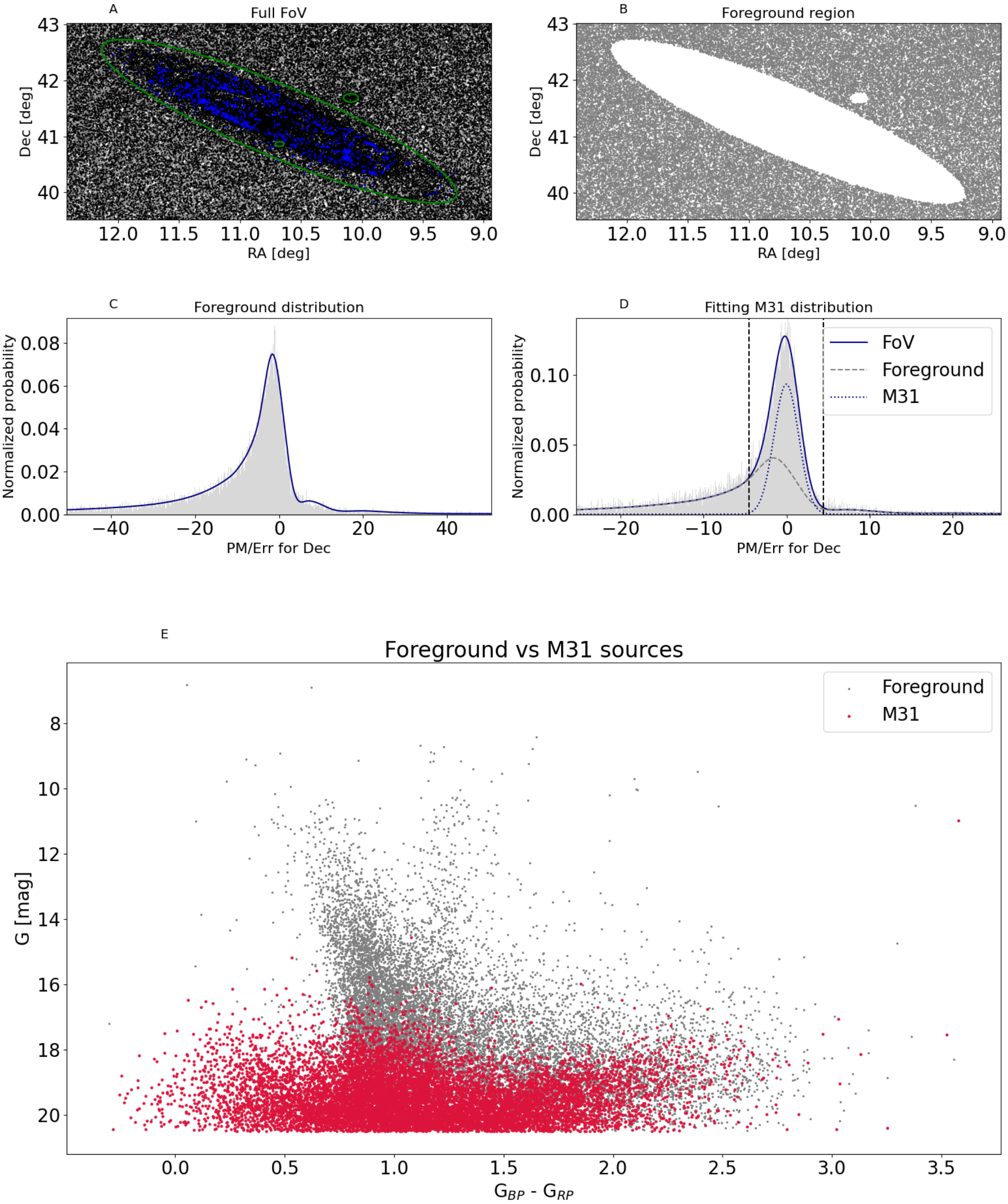


Fig. 1. Using *Gaia* to identify and remove foreground sources. (A) The field-of-view of *Gaia* sources (black dots) for M31. The big green ellipse marks the boundary we defined for the M31 galaxy, while the smaller green ellipses define M110 and M32 (inside M31's ellipse) and are excluded. The blue dots highlight the sources with known spectral classification in M31. (B) The foreground region, excluding the sources inside M110. (C) The distribution of the proper motion over its error for Dec, for all *Gaia* sources in the foreground region, fitted with a spline function. (D) The distribution of the proper motion over its error for Dec (solid line), for all sources along the line-of-sight of M31, which includes both foreground and galactic (M31) sources. We fitted this with a scaled spline, to account for the number of foreground sources expected inside M31 (dashed line), and a Gaussian function (dotted line). The vertical dashed lines correspond to the 3σ threshold of the Gaussian. Any source with values outside this region is flagged as a potential foreground source. (E) The *Gaia* CMD of all sources identified as galactic (red points) and foreground (gray). The majority of the foreground sources lie on the yellow branch of the CMD which is exactly the position at which we expect the largest fraction of the contamination.

The dataset was supplemented with optical photometry (g, r, i, z, y) obtained from the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS; Chambers et al. 2016), using sources with `nDetections` ≥ 2 to exclude spurious detections[3]. We also collected photometry from the *Gaia* DR2 (G, G_{BP}, G_{RP}; Gaia Collaboration et al. 2016, 2018).

We investigated all other available surveys in the optical and IR but the aforementioned catalogs provided the most numerous and consistent sample with good astrometry for our target galaxies (M31, M33, IC 1613, WLM, Sextans A). Significant populations of massive stars are well-known for the Magellanic Clouds and the Milky Way but there are issues that prohibited us from using them. The Clouds are not covered by the Pan-STARRS survey, which means that photometry from other surveys should be used that would make the whole sample inhomogeneous (with all possible systematics introduced by the different instrumentation, data reductions, etc). Although Milky Way is covered by both Pan-STARRS and *Spitzer* surveys, there are hardly any data available for the most interesting sources, such as B[e] Supergiants, WRs, and LBVs, through the *Spitzer* Enhanced Imaging Products (which focus on the quality of the products and not completeness). Therefore, we are limited to M31 and M33 galaxies to build our training sample.

2.2. Removing foreground stars

The source lists compiled from the photometric surveys described in the previous section, contain mostly genuine members of the corresponding galaxies. It is possible though that foreground sources may still contaminate these lists. To optimize our selection we queried the *Gaia* DR2 catalog (Gaia Collaboration et al. 2016, 2018). With the statistical handling of the astrometric data we were able to identify and remove most probable foreground sources in the line-of-sight of our galaxies.

We first defined a sufficiently large box around each galaxy, such as a 3.5 deg × 3.5 deg for M31 and 1.5 deg × 1.5 deg for M33, which yielded 145837 and 34662 sources, respectively. From these we first excluded all sources with either non-existent or poorly defined proper motions (pmra_error ≥ 3.0 mas, pmdec_error ≥ 3.0 mas), or parallax (parallax_error ≥ 1.5mas), with large astrometric excess noise (astrometric_excess_noise ≥ 1.0; following the cleaning suggestions by Lindegren et al. 2018), and fainter than our limit set in the optical (phot_g_mean_mag ≥ 20.5). These quality cuts left us with 78375 and 26553 sources in M31 and M33, respectively.

The boundary for each galaxy was determined as the ellipse at which the star density dropped significantly at approximately the density of the background. This boundary was visually inspected also so that it masks the main body (disk) of each galaxy where our targets are expected to be located (and to exclude contaminating regions inside and outside the galaxy, namely M32 and M110 galaxies for M31, see Fig. 1, Panel A; for M33 see Fig. B.1). Therefore, we could securely assign the remainder of stars as foreground objects (see Fig. 1, Panel B). From these we obtained the distributions on the proper motions in RA and Dec (over their corresponding errors) and the parallax (over its error). We fitted these distributions with a spline to allow more flexibility (see Dec for example in Fig. 1, Panel C).

Similarly, we plotted the distributions for all sources within the ellipse, which contained both galactic and foreground sources. To fit these we used a combination of a Gaussian and a spline function (see Fig. 1, Panel D). The spline was derived from the sources outside the galaxy (Fig. 1, Panel C), but when used for the sources within the ellipse it was scaled down according to the ratio of the area outside and inside the galaxy (assuming that the foreground distribution does not change). From the estimated widths of the Gaussian distributions (M31: pmRA/error = 0.04 ± 1.28, pmDEC/error = −0.03 ± 1.48, parallax/error = 0.21 ± 1.39; M33: pmRA/error = 0.12 ± 1.18, pmDEC/error = 0.05 ± 1.31, parallax/error = −0.03 ± 1.16) we defined as foreground sources those with values larger than 3σ in any of the above quantities. For the parallax we took into account the systematic 0.03 mas offset induced by the global zero point found by Lindegren et al. (2018). This particular cut was applied only to sources with actual positive values, as zero or negative values are not decisive for exclusion. In the *Gaia* CMD of Fig. 1, Panel E, we show all sources identified as members of the host galaxy (red points) and foreground (gray). The majority of the foreground sources lie on the yellow branch of the CMD which is exactly the position at which we expect the largest fraction of the contamination.

This process was successful in the cases of M31 and M33 due to the numerous sources that allow their statistical handling. In the other galaxies, where the field-of-view is substantially smaller, the low numbers of sources led to poorer (if any) estimation of these criteria. Consequently, for those galaxies we considered as foreground sources those with any of their *Gaia* properties (pmRA/error, pmDEC/error, or parallax/error) larger than 3σ of the largest measured errors, following the most conservative approach. In practice, this means that we used the same criteria to characterize foreground sources as with M31.

2.3. Collecting spectral types

The use of any supervised machine-learning application requires a training sample. It is of paramount importance that this is well defined, i.e. to cover the parameter space spanned by the objects under consideration. For this reason, we performed a meticulous search of the literature to obtain a sample as complete as possible to our knowledge with known spectral types (that were used as labels). The vast majority of collected data are found in M31 and M33. The source catalogs were retrieved primarily from Massey et al. (2016) as part of their Local Group Galaxy Survey (LGGS) survey, complemented by other works (see Table 1 for the full list of numbers and references used).

In all cases we carefully checked for and removed duplicates, while in a few cases we updated the classification of some sources based on newer works (e.g. candidate LBVs to B[e]SGs based on Kraus 2019). The initial catalogs for M31 and M33 contain 1142 and 1388 sources with spectral classification see Fig. 1, Panel A, blue dots), respectively. Within these sources we purposely included some outliers (such as background galaxies, QSO, e.g. Massey et al. 2019).

A significant fraction of these sources (∼ 64%) have *Gaia* astrometric information. Applying the criteria of the previous section we got 58 (M31) and 76 (M33) sources marked as foreground[4]. After removing those we are left with 1084 M31 and 1312 M33 sources, which are cross-matched with the photometric catalogs, considering single matches only at 1" (see Section

[3] Although DR2 became available after the compilation of our catalogs it contains information from the individual epochs. DR1 provides the object detection and their corresponding photometry from the stacked images, which we opted to use.

[4] There are 696 (M31) and 926 (M33) sources with *Gaia* information. The identification of 58 (M31) and 76 (M33) sources as foreground corresponds to a ∼ 8% contamination. Given that there are 446 (M31) and 462 (M33) additional sources but without *Gaia* values we expect another ∼ 72 sources to be foreground (according to our criteria) that remained in our catalog.

Table 1. List of references with their corresponding number of sources that contribute to our collected sample.

Galaxy (total)	Reference	# sources
WLM (36)	Bresolin et al. (2006)	20
	Britavskiy et al. (2015)	9
	Levesque & Massey (2012)	7
M31 (1142)	Massey et al. (2016)	966
	Gordon et al. (2016)	82
	Neugent et al. (2019)	37
	Drout et al. (2009)	18
	Massey et al. (2019)	17
	Kraus (2019)	11
	Humphreys et al. (2017)	6
	Neugent et al. (2012)	3
	Massey et al. (2009)	2
IC 1613 (20)	Garcia & Herrero (2013)	9
	Bresolin et al. (2007)	9
	Herrero et al. (2010)	1
	Britavskiy et al. (2014)	1
M33 (1388)	Massey et al. (2016)	1193
	Massey & Johnson (1998)	49
	Neugent et al. (2019)	46
	Humphreys et al. (2017)	24
	Massey et al. (2007)	13
	Gordon et al. (2016)	12
	Drout et al. (2012)	11
	Massey et al. (2019)	10
	Kraus (2019)	7
	Massey (1998)	6
	Kourniotis et al. (2018)	4
	Humphreys et al. (2014)	4
	Massey et al. (1996)	3
	Martin & Humphreys (2017)	2
	Neugent & Massey (2011)	2
	Bruhweiler et al. (2003)	2
Sextans A (16)	Camacho et al. (2016)	9
	Britavskiy et al. (2015)	5
	Britavskiy et al. (2014)	1
	Kaufer et al. (2004)	1

2.1). After this screening process our final sample consists of 527 (M31) and 562 (M33) sources.

We compiled spectral types for three more galaxies, i.e. WLM, IC 1613, and Sextans A (see Table 1), to use as test cases. Among a larger collection of galaxies, these three offered the most numerous (albeit small) populations of classified massive stars, i.e. 36 sources in WLM, 20 in IC 1613, and 16 in Sextans A. Although a handful more sources could potentially be retrieved for other galaxies the effort to collect the data (individually from different works) would not match the very small increase in the sample.

We present the first few lines of the compiled list of objects for guidance regarding its form and content in Table A.1.

3. Application of machine learning

In this section we provide a short description of the algorithms chosen for this work (for more details see, e.g., Baron 2019; Ball & Brunner 2010). The development of a classifier for massive stars requires the inclusion of "difficult" cases, like those that are short-lived (such as YSGs with a duration of a few thousand years; Neugent et al. 2010; Drout et al. 2009) or very rare (e.g. LBVs, Weis & Bomans 2020; B[e]SGs, Kraus 2019). To secure the training of the algorithms with specific targets, we preferred the use of supervised algorithms. However, any algorithm needs the proper input which is determined by the class and feature selection. Finally, we show the implementation and the optimization of the methods.

3.1. Selected Algorithms

Support Vector Machines (Cortes & Vapnik 1995) is one of the most well-established methods used in a wide range of topics. Some indicative examples include classification problems for variable stars (Pashchenko et al. 2018), black hole spin (González & Guzmán 2019), molecular outflows (Zhang et al. 2020), and supernova remnants (Kopsacheili et al. 2020). The method searches for the line / hyperplane (in two / multiple dimensions, respectively) that separates the input data (features) into distinct classes. The optimal line (hyperplane) is defined as the one that maximizes the support vectors, i.e. the distance of each point with the boundary, which leads to the optimal distinction between the classes. One manifestation of the method, designed better for classification purposes, such as our problem, is the Support Vector Classification (SVC; Ben-Hur et al. 2002) which uses a kernel to better map the decision boundaries between the different classes.

Astronomers are great machines when it comes to classification processes. A well-trained individual can easily identify the most important features for a particular problem (e.g. spectroscopic lines) and, according to specific (tree-like) criteria, can make fast and accurate decisions to classify sources. However, their strongest drawback is low efficiency as they can only process one object at a time. Although automated decision trees can be much more efficient than humans, they tend to overfit, i.e. they learn too well the data they are trained on and can fail when applied to unseen data. A solution to overfitting is Random Forest (RF; Breiman 2001), an ensemble of decision trees, each one trained on a random subset of the initial features and sample of sources. Some example works include Jayasinghe et al. (2018) and Pashchenko et al. (2018) for variable stars, Arnason et al. (2020) to identify new X-ray sources in M31, Möller et al. (2016) on supernovae Type Ia classification, Plewa (2018) and Kyritsis et al. (2022) for stellar classification. When RF is called to action, the input features of an unlabeled object propagate through each decision tree and provide a predicted label. The final classification is the result of a majority vote among all labels predicted by independent trees. Therefore, RF overcomes the problems of single decision trees as they generalize very well and can handle large numbers of features and data efficiently.

Neural networks originate from the idea of simulating the biological neural networks in animal brains (McCulloch & Pitts 1943). The nodes (that are located in layers) are connected and process an input signal according to their weight which was assigned to them during the training process. Initial applications in Astronomy were first performed in the 1990s (e.g. Odewahn et al. 1992 on star and galaxy discrimination, Storrie-Lombardi et al. 1992 on galactic morphology classification) but recent advance in computational power as well as in software development, allowing easy implementation, have revolutionized the field. Deeper and more complex neural network architectures have been developed, such as using deep convolutional networks to classify stellar spectra (Sharma et al. 2020), and supernovae along with their host galaxies (Muthukrishna et al. 2019), generative adversarial networks to separate stars from quasars (Makhija et al. 2019), recurrent neural networks for variable star

Table 2. Groups of spectral types of our initial sample (column 1) and their corresponding number of sources (column 2). Combining into classes (columns 3) leads to the total number of sources combined per class (column 4, see Section 3.2), and the final numbers (column 5) after removing 44 objects without full photometry in all bands.

Group [1]	initial # [2]	Class [3]	class # [4]	final # w/phot [5]
O	17	BSG	261	250
Oc	1	-		
Oe	2	BSG		
On	6	BSG		
B	156	BSG		
Bc	11	-		
Be	7	BSG		
Bn	18	BSG		
A	51	BSG		
Ac	3	-		
Ae	2	BSG		
An	2	BSG		
WR	50	WR	53	42
WRc	3	-		
WRn	3	WR		
LBV	6	LBV	6	6
LBVc	18	-		
BeBR	6	BeBR	17	16
BeBRc	11	BeBR		
F	21	YSG	103	99
Fc	4	-		
G	15	YSG		
YSG	67	YSG		
YSGc	16	-		
K	67	RSG	512	496
Kc	3	-		
M	142	RSG		
Mc	5	-		
RSG	250	RSG		
RSGb	53	RSG		
RSGc	36	-		
AGN	2	GAL	24	23
QSO	17	GAL		
QSOc	1	-		
GAL	5	GAL		
Total	1077		976	932

classification (Naul et al. 2018). For the current project a relatively simple shallow network with a few fully-connected layers (Multilayer Perceptron, MLP) proved sufficient.

In summary, the aforementioned techniques are based on different concepts, e.g. SVC tries to find the best hyperplane that separates the classes, RF decides the classification result based on the thresholds set at each node (for multiple trees), while neural networks attempt to highlight the differences in the features that best separate the classes. We implemented an ensemble meta-algorithm which combines the results from all three, different, approaches. Initially each method provides a classification result with a probability distribution across all selected classes (see Section 3.2). Then these are further combined to obtain the final classification (described in detail in Section 4.2).

3.2. Class selection

When using supervised machine-learning algorithms it is necessary to properly select the output classes. In our case we are particularly interested in evolved massive stars, because the magnitude-limited observations of our target galaxies mainly probe the upper part of the HRD. In our compiled catalog we had a large range of spectral types, from detailed ones (such as M2.5I, F5Ia, B1.5I) up to more generic terms (such as RSG, YSG). Given the small numbers per individual spectral type, as well as the continuous nature of spectral classification which makes the separation of neighboring types difficult, we lack the ability to build a classifier sensitive to each individual spectral type. To address that we combined spectral types in broader classes, without taking into account luminosity classes (i.e. main sequence stars and supergiants for the same spectral type were assigned to the same group). This is a two-step process as we first assigned all types to certain groups, and then, during the application of the classifier, we experimented with which classes are best detectable with our approach (given the lack of strict boundaries between these massive stars, which is a physical limitation and not a selection bias). For the first step we grouped the 1089 sources (both in M31 and M33) as following:

- Sources of detailed sub-types were grouped by their parent type (e.g. B2 I, B1.5 Ia to B group; A5 I, A7 I to A group; M2.5 I, M2-2.5 I to M group, etc). Some individual cases with uncertain spectral type were assigned as follows: three K5-M0 I sources to the K group, one mid-late O to the O group, one F8-G0 I to the F group, one A9I/F0I to the A group.
- All sources with emission or nebular lines were assigned to the parent type group with an "e" or "n" indicator (e.g. B8 Ie to Be group, G4 Ie to the Ge, B1 I+Neb to Bn, O3-6.5 V+Neb to On).
- Sources with an initial classification as RSG or YSG were assigned directly to their corresponding group.
- RSG binaries with a B-companion (Neugent et al. 2019) to the RSGb group.
- Secure LBVs and B[e]s were kept as separate groups (as LBV and BeBR, respectively). A source classified as HotLBV was assigned to the LBV group.
- All sources classified as Wolf-Rayet stars (of all subtypes) including some individual cases (WC6+B0 I, WN4.5+O6-9, WN3+abs, WNE+B3 I, WN4.5+O, and five Ofpe/WN9) were grouped under one group (WR), except three sources that are characterized by nebular lines and were assigned to the WRn group.
- Galaxies, AGN and QSOs were grouped under their corresponding groups (GAL, AGN, QSO, respectively).
- All sources with an uncertainty flag (":" or "c") were assigned their broader group followed by a "c" flag, to indicate that these are candidates (i.e. not secure) classifications, such as Ac, Bc, YSGc, WRc, QSOc. One source classified as B8Ipec/cLBV was assigned to the LBVc group.
- Complex or very vague cases were disregarded (eight "Hot-Supergiant" sources, and one source from each of the following types: "WarmSG", "LBV/Ofpe/WN9", "Non-WR(AI)", "FeIIEm.Line(sgB[e])").

Thus, after removing the 12 sources from the last step we are left with 1077, split into 35 groups (see Table 2, column 1 and their corresponding numbers in column 2). However, these groups may contain similar objects, or in many cases a limited number of sources which may not be securely classified. To optimize our approach we experimented extensively by combining (similar) groups to broader classes to obtain the best results.

All hot stars (i.e. O,B,A groups, including sources with emission "e" and nebular "n" lines) were combined under the BSG class, after removing the uncertain sources (indicated as candidates). For the YSG class we considered all sources from the F, G and YSG groups, excluding only the candidates again (i.e. Fc and YSGc, since many of the latter ones are highly uncertain; Massey et al. 2016). For the RSG class we combined the K, M, RSG and RSGb groups,excluding the candidates (i.e Kc, Mc, RSGc). The BeBR class includes both the secure and the candidate sources, because they show the same behavior (see Section 3.4) and there are more constraints to characterize a source as B[e] (see Kraus 2019). More specifically the BeBRc sources were actually the result of constraining further the classification of candidate LBVs (Kraus 2019). Therefore, we kept only the secure LBVs (LBV group) to form their corresponding class. For the WR class we used all available sources, although they are of different types, as a further division would not be efficient. The last class, GAL, includes all non-stellar background objects (galaxies, AGNs, QSOs, except for the one candidate QSO) that were used as potential outliers. We do not expect any other type of outlier (but for an $\sim 8\%$ foreground contamination) since at the distances of our target galaxies we are actually probing the brighter parts of the HRD where the supergiant stars are located. The number of sources finally selected for each class is shown in Table 2 (column 4), where we used the class name to indicate which groups contribute to the class (column 3) while a "-" shows that a particular group is ignored. The total number of selected sources is 976.

3.3. Imbalance treatment

What is evident from Table 2 is that we have an imbalanced sample of classes, which is very typical in astronomical applications (see also in Dorn-Wallenstein et al. 2021 for similar problems). In particular, RSG are the most populated class, along with BSG, with a few hundred sources. YSG include about a hundred sources, but WR, GAL, BeBR, and most importantly LBV, include a few tens at most. To tackle this we can either use penalizing metrics of performance, i.e. evaluations in which the algorithm provides different weights to specific classes, or to train the model using adjusted sample numbers (by over- / under-sampling the least / most populated classes). We experimented with both approaches and we found a small gain when using the resampling approach also.

A typical approach to oversampling is duplicating objects. Although this may be a solution in many cases, it doesn't help with sampling better the feature space, i.e. it does not provide more information. An alternative approach is to create synthetic data. To this purpose, in this work we used a commonly adopted algorithm, the Synthetic Minority Oversampling TEchnique (SMOTE; Chawla et al. 2002) which generates more data objects by following these steps: i. it selects randomly a point (A) that corresponds to a minority class, ii. it finds k-nearest neighbors (of the same class), iii. it randomly chooses one of them (B), iv. it creates a synthetic point randomly along the line that connects A and B in the feature space. The benefits from this approach is that the feature space is sampled better and all features are taken into account to synthesize the new data points. On the other hand, the limitation is how well the initial sample per class is really representative of each class' feature space. In any case, the number of points to be added is arbitrary and can very well match the majority class.

Table 3. Number and fraction of sources per class before and after resampling to treat for imbalance (using the SMOTEENN approach). The fractions correspond to the total number of sources used in the original and resampled sets, respectively.

Class	Original sources		Resampled sources	
	(#)	(%)	(#)	(%)
BSG	250	26.8	496	14.9
YSG	99	10.6	488	14.6
RSG	496	53.2	493	14.8
BeBR	16	1.7	495	14.9
LBV	6	0.6	444	13.3
WR	42	4.5	453	13.6
GAL	23	2.4	452	13.6

At the same time this procedure can create noise, especially when trying to oversample classes with very few sources (e.g. LBV with only 6 sources available in total). Better results are obtained when the oversampling of the minority classes is combined with undersampling the majority class. For the latter we experimented with two similar approaches: the Tomek Links Tomek (1976) and the Edited Nearest Neighbors (ENN; Wilson 1972). In the first one, the method identifies the pairs of points that are closest to each other (in the feature space) and belong to different classes (the Tomek links). These are noisy instances or located on the boundary between the two classes (in a multiclass problem it is the one-versus-rest scheme that is used, i.e. the minority compared to all other classes collectively referred to a majority). By removing the point corresponding to the majority class the class separation increases, and the number of majority class points are reduced. In the ENN approach the three-nearest neighbors to a minority point are found and removed when belonging to the majority class. Thus, the ENN approach is a bit more aggressive than Tomek links, as it removes more points.

In conclusion, the combination of SMOTE, which creates synthetic points from the minority class to balance the majority class, and the undersampling technique (either Tomek Links or ENN), which cleans irrelevant points in the boundary of the classes, help to increase the separation. For the implementation we used the `imbalanced-learn` package[5] (Lemaître et al. 2017) and more specifically the ENN approach `imblearn.combine.SMOTEENN()`, which provided slightly better results from Tomek Links. We used `k_neighbors=3` for SMOTE (due to the small number of LBV). We opted to use the default values for `sampling_strategy`, which corresponds to 'not majority' for SMOTE (which means that all classes are resampled except for RSGs) and 'all' for ENN function, which cleans the majority points (considering one-versus-rest classes). In Table 3 we provide an example of the numbers and fractions of sources per class available before and after resampling (the whole sample).

3.4. Feature selection

Feature selection is a key step in any machine-learning problem. To properly select the optimal features in our case we first examined data availability. In Table 4 we list the different classes (column 1) and the number of available sources per class (column 2).

[5] `https://github.com/scikit-learn-contrib/imbalanced-learn`

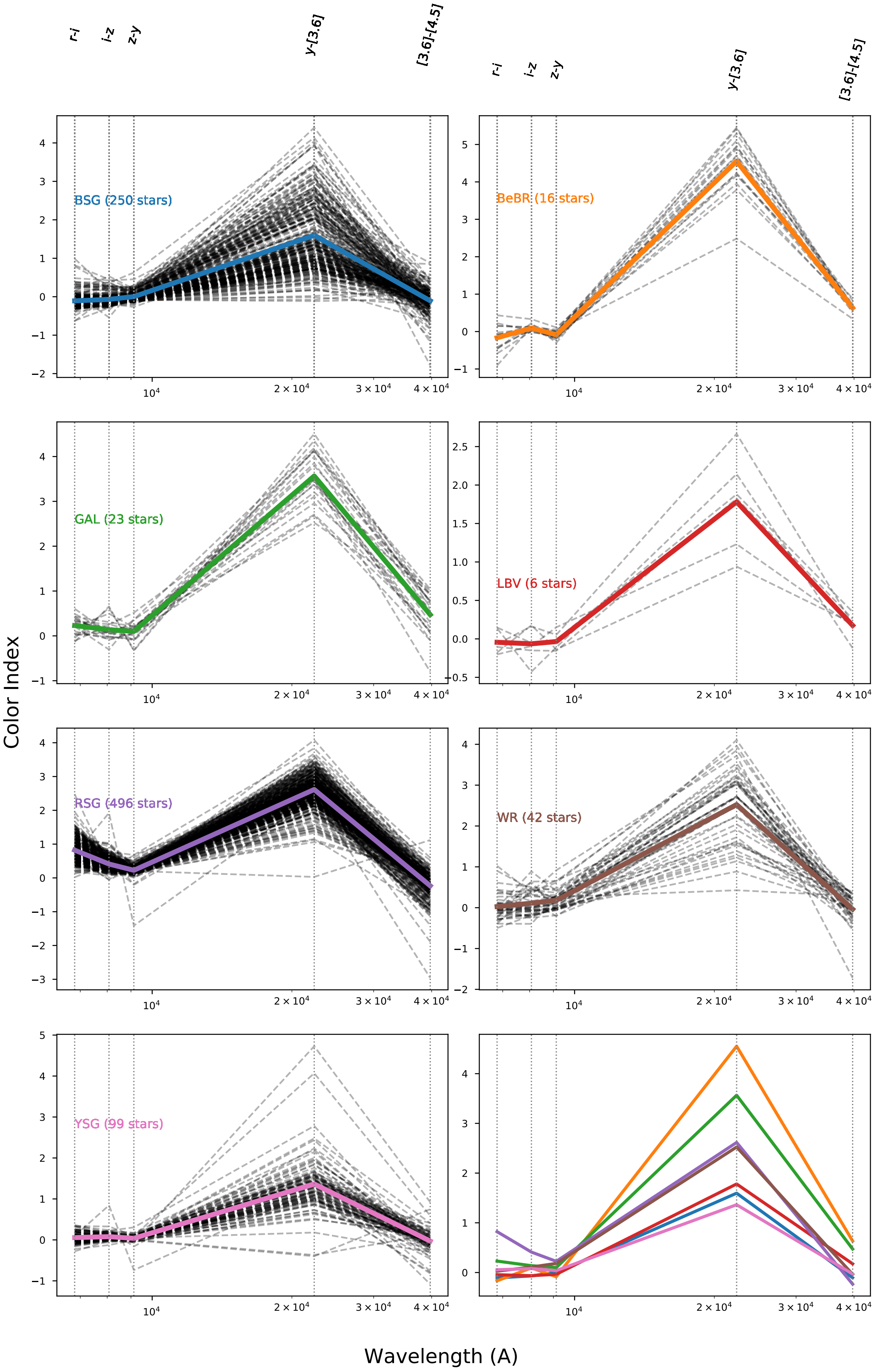


Fig. 2. The color indices (features) vs. wavelength per class. The black dashed lines correspond to the individual sources while the colored solid lines corresponds to their average. The last panel contains only the averaged lines to highlight the differences between the classes, with the most pronounced differences in the $y-[3.6]$ index (as BeBR are the brightest IR sources, on average, followed by the GAL, RSG, and WR classes; see text for more). The number of sources in each panel corresponds to the total number of selected sources (see Table 2, column 5). The vertical dashed lines correspond to the average wavelength per color index, as shown at the top of the figure.

Table 4. Data availability per class and photometric band. The first column lists the classes used and the second one the corresponding number of sources in the sample. For each class, the following columns provide the fractions of sources with secure measurements in the corresponding photometric bands and their errors (which do not include objects with problematic measurements and upper limits).

Class	Sources	[3.6]	$\sigma_{[3.6]}$	[4.5]	$\sigma_{[4.5]}$	[5.8]	$\sigma_{[5.8]}$	[8.0]	$\sigma_{[8.0]}$	[24]	$\sigma_{[24]}$
	(#)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
BSG	261	100	100	100	100	100	80	100	70	100	41
YSG	103	100	100	100	100	100	92	100	78	99	30
RSG	512	100	100	100	100	99	99	100	93	99	41
BeBR	17	100	100	100	100	100	100	100	100	100	94
LBV	6	100	100	100	100	100	100	100	100	100	66
WR	53	100	100	100	100	100	94	100	86	100	43
GAL	24	100	100	100	100	100	100	100	100	100	100

Class	Sources	g	σ_g	r	σ_r	i	σ_i	z	σ_z	y	σ_y
	(#)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)	(%)
BSG	261	96	96	96	96	96	96	96	96	96	96
YSG	103	96	96	96	96	96	96	96	96	96	96
RSG	512	96	93	97	97	97	97	97	97	97	97
BeBR	17	100	100	100	100	100	100	100	100	94	94
LBV	6	100	100	100	100	100	100	100	100	100	100
WR	53	83	83	84	83	88	88	90	88	86	84
GAL	24	100	100	100	100	100	100	100	95	100	100

Class	Sources	$J_{\rm UK}$	G	G$_{\rm BP}$	G$_{\rm RP}$
	(#)	(%)	(%)	(%)	(%)
BSG	261	81	90	87	87
YSG	103	82	96	95	95
RSG	512	84	96	94	94
BeBR	17	70	100	100	100
LBV	6	66	83	66	66
WR	53	75	71	50	50
GAL	24	83	95	83	83

In the following columns we provide the fractions of objects with photometry in the corresponding bands and with proper errors (i.e. excluding problematic sources and upper limits), per survey queried (*Spitzer*, Pan-STARRS, UKIRT Hemisphere Survey and *Gaia*). To build our training sample we required the sources to have (well) measured values across all bands. To avoid a significant decrease of the training sample (up to almost half in the case of LBV and BeBR classes) we chose not to include the $J_{\rm UK}$. Although we used *Gaia* data to derive the criteria to identify foreground stars, the number of stars with *Gaia* photometry in the majority of other nearby galaxies is limited. Thus, to ensure the applicability of our approach we discarded these bands also (which are partly covered by the Pan-STARRS bands).

The IR catalogs were built upon source detection in both the [3.6] and [4.5] images (e.g. Khan et al. 2015), while the measurements in the longer bands were obtained by just performing photometry in those coordinates (regardless of the presence of a source or not). However, in most cases there is a growing (with wavelength) number of sources with only upper limits in the photometry. As these do not provide secure measurements for the training of the algorithm we could not use them. If we were to take into account sources with valid measurements up to the [24] we would end up with less than 50% of sources for some classes (see in Table 4, e.g. the corresponding fractions of WR and YSG with secure error measurements). As this is not really an option we decided to remove all bands that contained a significant fraction of upper limits, i.e. above [5.8]. This, rather radical, selection is also justified by the fact that the majority of the unclassified sources (in the catalogs to which were are going to apply our method) do not have measurements in those bands. It is also interesting to point out that the majority of the disregarded sources belong to the RSG class (the most populated), which means that we do not lose any important information (for the training process).

From the optical set of bands, we excluded g for two reasons. About 130 sources fainter than 20.5 mag tend to have systematic issues with their photometry, especially red stars for which $g - r$ turns to bluer values. Also, due to the lack of known extinction laws for most galaxies, and the lack of data for the many sources, we opted not to correct for it. As g is the band most affected by extinction we opted to use only the redder bands to minimize its impact (Schlafly & Finkbeiner 2011; Davenport et al. 2014). Therefore, we kept r, i, z, and y bands and we performed the same strict screening to remove sources with upper limits. In total, we excluded 44 sources, reflecting a small fraction of the sample (∼ 4.5% - treating both M31 and M33 sources as a single catalog). We show the final number of sources per class in column 5 in Table 2 summing to 932 objects in total.

To remove any distance dependence in the photometry we opted to work with color terms, i.e. we obtained the consecutive magnitude differences: $r - i$, $i - z$, $z - y$, $y -$ [3.6], [3.6] $-$ [4.5]. We examined different combinations of these color indices, but the difference in the accuracy with respect to the best ones found is negligible. Those combinations contained color indices with wider wavelength range that are affected more from extinction that the consecutive colors. Moreover, they tend to be systematically more correlated, resulting in poorer generalization (i.e. when applied to the test galaxies, section 4.3). Some (less pronounced) correlation still exists in the consecutive color set also, because of the use of each band into two color combinations (except for r and [4.5]), and due to the stellar continuum, since the flux at each band is not totally independent from the flux measured in other bands. We also noticed that more optical colors help to better sample the optical part of the spectral energy distribution and separate more efficiently some classes (BSG and YSG in particular). The consecutive color set seems as the most intuitive selection, including well studied colors. Moreover, it represents how the slopes of the spectral energy distribution changes with wavelength.

We also experimented with other transformations of these data, such as fluxes, normalized fluxes, standardizing data (scaling magnitudes around their mean over their standard deviation) but we have not seen any significant improvement in the final classification results. Therefore, we opted for the simplest representation of the data which is the aforementioned color set.

In Fig. 2 we plot the color indices with respect to the their corresponding wavelengths (indicated by the vertical dashed lines). The dashed black lines correspond to the individual sources for each class while the colored solid lines to their average. In the last panel, we overplot all averaged lines to display the difference among the various classes. As this representation is equivalent to the consecutive slopes of the spectral energy distributions for each class we notice that the redder sources tend to have a more pronounced $y-[3.6]$ feature, a color index that characterizes the transition form the optical to the mid-IR photometry. The BeBR class presents the higher values due to the significant amount of dust (and therefore brighter IR magnitudes), followed by the GAL due to their PAH emission, the (intrinsically redder sources) RSG, and the WR class (due to their complex environments).

3.5. Implementation and optimization

An important step of every classification algorithm is to tune its hyperparameters, i.e. the parameters that control the training process. After having defined these the algorithm determines the values of the parameters used for each model (e.g. weights) based on the training sample. The implementation of all three methods (SVC, RF, MLP) was done through the `scikit-learn` v.0.23.1[6] (Pedregosa et al. 2011)[7].

For the optimal selection of the hyperparameters (and their corresponding errors) we performed a stratified K-fold cross-validation (`sklearn.model_selection.StratifiedKFold()`). With this, the whole sample is split into K sub-samples or folds (5 in our case), preserving the fraction representation of all classes of the initial sample into each of the folds. At each iteration one fold is used as the validation sample and the rest as training. By permuting the validation fold the classifier is trained over the whole sample. Since we are performing a resampling approach to correct for the imbalance in our initial sample (see Section 3.3), we note that this process is performed only in the training folds, while the evaluation of the model's accuracy is done on the (unmodified) validation fold. We stress that the validation fold in "unmodified", i.e. not resampled, in order to avoid data leakage and hence overfitting. The final accuracy score is the average value and its uncertainty corresponds to the standard deviation across all folds.

For the SVC process we used the `sklearn.svm.SVC()` function. We opted to train this model with the following selection of hyperparameters: `probability=True` to get probabilities (instead of a single classification result), `decision_function_shape = 'ovo'` which is the default option for multi-class problems, `kernel = 'linear'` which is faster than the alternative non-linear kernels and proved to be more efficient[8], `class_weight='balanced'` which weights rarer classes more (even after the resampling approach, as described in Section 3.3). We also optimize the regularization C parameter, which represents a penalty for misclassifications, i.e. the objects falling on the "wrong" side of the separating hyperplane. For larger values a smaller margin for the hyperplane is selected so that the misclassified sources decrease and the classifier performs optimally for the training objects. This may result in poorer performance when applied to unseen data. On the opposite, smaller values of C leads to a larger margin (i.e. a loose separation of the classes) at the cost of more misclassified objects. To optimize C we tested the result in the accuracy by changing the value of C from 0.01 to 200 (with a step of 0.1 in log space). We present these results in Fig. C.1, where the red line corresponds to the averaged values and the gray area to the 1σ error. As the parameter reaches fast to a plateau, the choice of this particular value does not affect significantly the accuracy above ~ 25, which is the adopted value.

For the RF classifier we used `sklearn.ensemble.RandomForestClassifier()`. To optimize it we searched the following hyperparameters over a range of values: `n_estimators` which is the number of trees in the forest (10-1000, step 50), `max_leaf_nodes` which limits the number of nodes in each tree, i.e. how large it can grow (2-100, step 2) , and `max_depth` which is the maximum depth of the tree (1-100, step 2), while the rest of the hyperparameters were left to their default values. We present their corresponding validation curves as obtained from 5-fold CV (with mean values as red lines and their 1σ uncertainty as gray areas) in Fig. C.2. Again, we see that above certain values the accuracy reaches to a plateau. Given the relative large uncertainties and the statistical nature of this test, the selection of the best values is not absolutely strict (they provide almost identical results). We opted to use the following values: `n_estimators=400`, `max_leaf_nodes=50`, `max_depth=30`. We also set `class_weight=`"balanced", similar to SVC, in addition to the resampling approach.

For the neural networks we used `sklearn.neural_network.MLPClassifier()`. In this case we performed a grid search approach (`sklearn.model_selection.GridSearchCV()`). This method allows for an exhaustive and simultaneous search over the requested parameters (with a cost in computation time). We started first by investigating the architecture of the network (e.g. number of hidden layers, number of nodes per layer) along with the three available types of methods for weight optimization (`'lbfgs'`, `'sgd'`, `'adam'`). We tried up to 5 hidden layers with up to 128 nodes per layer, using `'relu'` as the activation function (a standard selection). We present the results of this grid search in Fig. C.3 from which we obtained systematically better results for the `'adam'` solver (Kingma & Ba 2014), with

[6] `https://scikit-learn.org/`

[7] For the MLP/neural networks we have experimented extensively with TensorFlow v1.12.0 (Abadi et al. 2015) and Keras v2.2.4 API (Chollet et al. 2015). This allowed us to easily build and test various architectures for our networks. We used both dense (fully connected) and convolutional (CNN) layers, in which case the input data are 1D vectors of the features we are using. Given our tests we opted to use a simple dense network, which can be easily implemented also within the `scikit-learn` that helps with the overall simplification of the pipeline.

[8] The 'linear' kernel was more efficient in recovering the LBV class systematically in contrast to the default 'rbf' option.

the (relatively) best configuration being a shallow network with two hidden layers with 128 nodes each. Given this combination we further optimized the regularization parameter (`alpha`), the number of samples used to estimate the gradient at each epoch (`batch_size`) and the maximum number of epochs for training (`max_iter`), with the rest of the parameters left to their default values (with `learning_rate_init`=0.001). Similarly to the previous hyperparameters selections, from their validation curves (Fig. C.4) we selected as best values: `alpha`=0.13, `batch_size`=128, and `max_iter`=560. The classifier uses the Cross-Entropy loss, which allows probability estimates.

4. Results

We first present our results from the individual applications of the different machine-learning algorithms to the M31 and M33 galaxies. Then, we describe how we combine the three algorithms to obtain a combined result. Finally, we apply the combined classifier to the test galaxies.

4.1. Individual application to M31 and M33

4.1.1. Overall performance

Having selected the optimal hyperparameters for our three algorithms we investigated the individual results as obtained by directly applying them to our dataset. For this we need to split the sample into a training set (70%) and evaluate the results on a validation set (30%), which is a standard option in the literature. The split has been performed individually per class to ensure the same fractional representation of all classes in the validation sample. The resampling approach to balance our sample (as described in Section 3.3) is applied only on the training set. The model is then trained on this balanced set and the predictions are made on the original validation set.

Given a specific class, as True Positives (TP) we refer to the objects that are predicted correctly to belong to this class, while True Negatives (TN) are those that are predicted correctly to not belong to the class. False Positives (FP) are the ones which are incorrectly predicted to belong while False Negatives (FN) are the ones that are incorrectly predicted to not belong to the class.

In Fig. 3 we show example runs for the SVC, RF, and MLP methods. The first row corresponds to the confusion matrix, a table that displays the correct and incorrect classification results per class. Ideally this should be a diagonal table. The presence of sources in other elements provides information about the contamination of classes (or how much the method is miss-classifying the particular class). Another representation is given in the second row where we plot the scores of the (typically used) metrics for each class. Precision (defined as $\mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$) refers to the number of objects that are predicted correctly to belong to a particular class over the total number of identified objects for this class (easily derived if we look at the numbers across the columns of the confusion matrix). Recall (defined as $\mathrm{TP}/(\mathrm{TP} + \mathrm{FN})$) is the number of class objects over the total real population for this class (derived from the rows of the confusion matrix). Therefore, the precision indicates the ability of the method to detect an object of the particular class while recall its ability to recover the real population. F1-score is the harmonic mean of the two previous metrics (defined as $\mathrm{F1-score} = 2 \times (\mathrm{precision} \times \mathrm{recall})/(\mathrm{precision} + \mathrm{recall})$). In our case, we are mainly using the recall metric as we are interested to minimize the contamination and therefore to recover as many correct objects as possible. This is required especially for the classes with the smallest numbers that reflect the rarity of their objects, such as BeBR and LBV. We report our results using the weighted balance accuracy (henceforth accuracy) which corresponds to the average of recall values across all classes, weighted by the number of objects per class. This is a reliable metric of the overall performance when training over a wide number of classes (Grandini et al. 2020).

From Fig. 3 we see that the accuracy achieved for SVC, RF, and MLP is $\sim$ 78%, $\sim$ 82%, and $\sim$ 83%, respectively. These values are based on a single application of the algorithms, i.e the evaluation of the models on the validation set (the 30% of the whole sample). However, we left out an important fraction of information, which due to our small sample, is important. Even though we upsampled to account for the scarcely-populated classes, this happened (at each iteration) solely for those sources of the training sample, which implies that - again - only a part of the whole dataset's feature space was actually explored. To compensate for that, the final model was actually obtained by training over the whole sample (after resampling). In this case there was no validation set to perform directly the evaluation. To address that we used a repeated K-fold cross-validation to obtain the mean accuracy and the recall per class, which in turn provided the overall expected accuracy. Using 5 iterations (and 5 folds per iteration) we obtained 78 ± 3%, 82 ± 2%, and 82 ± 2% for SVC, RF, and MLP, respectively (the error is the standard deviation of the average values over all K-folds performed). In Table 5 we show the accuracy ('overall'), and the recall as obtained per class.

Dorn-Wallenstein et al. (2021), using the SVC method and a larger set of features (12) including variability indices, achieved slightly better accuracy ($\sim$ 90.5%) but for a coarser classification of their sources (i.e. for only four classes: 'hot', 'emisison', 'cool', and 'contamination' stars). When they used their finer class grid with 12 classes their result is $\sim$ 54%[9].

4.1.2. Class recovery rates

The results per class are similar for all three methods. They can recover the majority of the classes efficiently, with the most prominent class being the RSG with $\sim$ 95% success (similar to Dorn-Wallenstein et al. 2021). Decent results are returned for the BSG, YSG and GAL, within a range of $\sim 60 - 80$%.

The class for which we obtained the poorest results is the LBV. SVC is the most effective in recovering a fraction of the LBV ($\sim$ 30%, albeit with a large error of 43%) while the other two methods failed. LBV is an evolutionary phase of main-sequence massive O-type stars before they lose their upper atmosphere (due to strong winds and intense mass-loss episodes) and end up as WR stars. They tend to be variable both photometrically as well as spectroscopically, displaying spectral types from B- to G-type. Hence, physical confusion between WR, LBV, and BSG is expected, as it is indicated by the lower recall values and the confusion matrices (see Fig. 3). Moreover, the rarity of these objects leads to a small-populated class for which their features are not well-determined and, consequently, the classifier has significant issues to distinguish them from other classes. On the other hand, SVC examines the entire feature space which is the reason for the (slightly) improved recall for LBV in this case

[9] The balanced accuracy reported by Dorn-Wallenstein et al. (2021) is the average recall across all classes, i.e. without weighting with the frequency for each class. This metric is insensitive to class distribution (Grandini et al. 2020). We converted the reported values to the weighted balanced accuracy to directly compare our results.

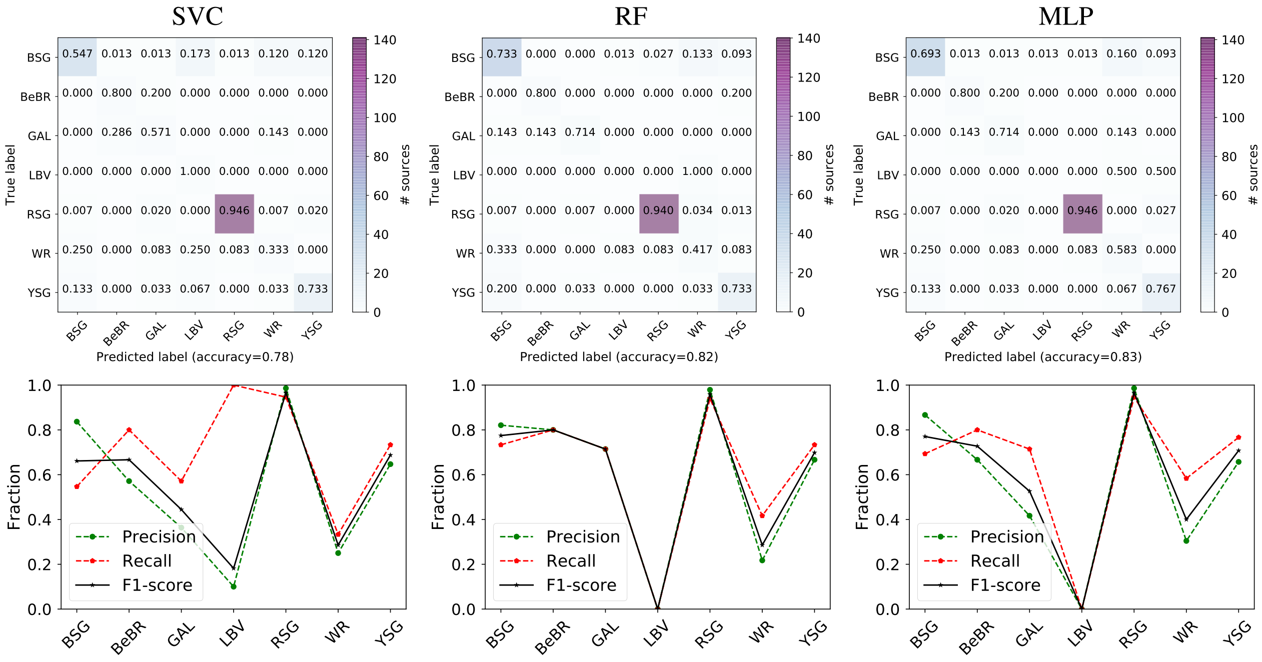


Fig. 3. The confusion matrices (upper panels) for the SVC, RF, and MLP methods, respectively, along with the characteristic metrics (precision, recall, and F1-score; lower panels). These results originate from single runs, i.e. by splitting the initial sample into 70% for the training sample, which is then resampled to produce a balanced sample before training each model, and finally applying the model to the rest of the sample (30%, the validation). In general, the algorithms perform well except for the cases of LBV and WR (see Section 4.1 for more details).

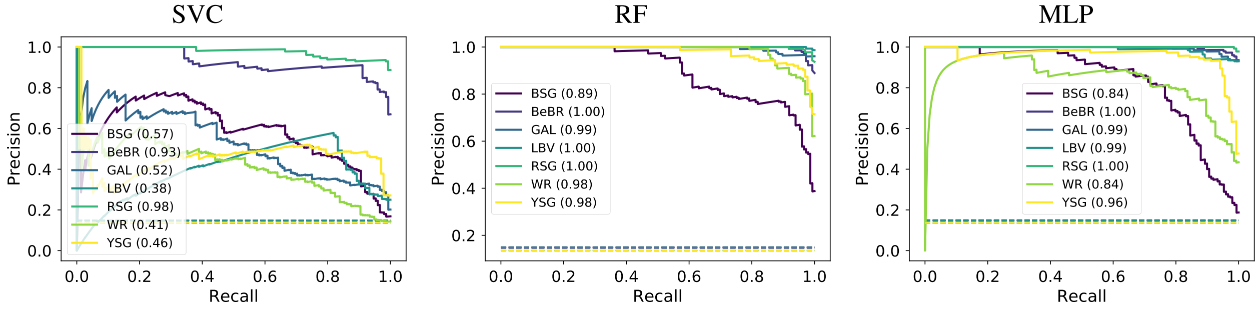


Fig. 4. The Precision Recall curves for the three methods, along with the values of the Area Under Curve for each class (in parenthesis). In all cases the result of the comparison of each class against all others provide very good and consistent results, well above the random classifier (indicated for each class by the horizontal dashed lines; see Section 4.1).

Table 5. The performance for each method and per class, after a repeated K-fold cross validation (see Section 4.1 for details).

Class	SVC	RF	MLP	combined
overall	0.78 ± 0.03	0.82 ± 0.02	0.82 ± 0.02	0.83 ± 0.02
BSG	0.58 ± 0.08	0.71 ± 0.06	0.71 ± 0.07	0.71 ± 0.06
BeBR	0.80 ± 0.23	0.79 ± 0.24	0.73 ± 0.25	0.81 ± 0.17
GAL	0.58 ± 0.22	0.63 ± 0.21	0.73 ± 0.24	0.71 ± 0.17
LBV	0.28 ± 0.43	0 ± 0	0 ± 0	0 ± 0
RSG	0.93 ± 0.03	0.95 ± 0.02	0.94 ± 0.02	0.94 ± 0.02
WR	0.43 ± 0.15	0.40 ± 0.16	0.46 ± 0.19	0.48 ± 0.24
YSG	0.78 ± 0.08	0.75 ± 0.10	0.77 ± 0.12	0.80 ± 0.08

(Dorn-Wallenstein et al. 2021 report full recovery but probably because of overfitting). Due to the small number and the rather inhomogeneous sample of WR, all the classifiers have difficulties to correctly recover the majority of these sources. The best result is provided by MLP at ∼ 46%, less than the ∼ 75% reported by Dorn-Wallenstein et al. 2021. Despite the small sample size of the BeBR class, it is actually recovered successfully (> 79%). As BeBR (including candidate sources) form a more homogeneous sample than LBV and WR, their features are well-characterized that helps the algorithms to separate them.

To better visualize the performance of these methods we constructed the Precision Recall curves, which are better suited in the case of imbalanced data (Davis & Goadrich 2006; Saito & Rehmsmeier 2015). During this process the classifier works in a one-versus-rest mode, i.e. it only checks whether the objects belong or not to the examined class. In Fig. 4 we show the curves for each algorithm. The dashed (horizontal) lines correspond to the ratio of positive objects (per class) over the total number of objects in the training data. Any model found at this line or below has no ability to predict anything better than random (or worse). Therefore, the optimal curve directs towards the upper right corner of the plot (with precision=recall=1). In all cases the classifiers are better than random. RF displays systematically the best curves. In SVC, RSG and BeBR are almost excellent and the rest of the classes display similar behavior. For MLP, all classes except for BSG and WR are very close to the optimal position.

Another metric is obtained if we measure the fraction of the Area Under the Curve. This returns a single value (within the 0-1 range) depicting the ability of the classifier to distinguish the corresponding class over all the rest. In Fig. 4 we show these values within the legends. In general, we achieve high values which means that our classifiers can efficiently distinguish the members of a class against all others. These consistent results add further support that the careful selection of our sample has worked and that the methods work efficiently (given the class limitations).

4.2. Ensemble models

4.2.1. Approaches

A common issue with machine learning applications is choosing the best algorithm for the specific problem. This is actually impossible to achieve a priori. Even in the case of different algorithms that provide similar results it can be challenging to select one of them. However, there is no reason to exclude any. Ensemble methods refers to approaches that combine predictions from multiple algorithms, similar to combining the opinions of various "experts" in order to reach to a decision (e.g. Re & Valentini 2012). The motivation of ensemble methods is to reduce the variance and the bias of the models (Mehta et al. 2019).

A general grouping consists of bagging, stacking, and boosting. Bagging (bootstrap aggregation) is based on training on different random sub-samples whose predictions are combined either by majority vote (e.g. which is the most common class) or by averaging the probabilities (RF is the most characteristic example). Stacking (stacked generalization) refers to a model which trains on the predictions of other models. These base models are trained on the training data and their predictions (sometimes along with the original features) are provided to train the meta-model. In this case, it is better to use different methods that use different assumptions, so that to minimize the bias inherent by each method. Boosting refers to methods that focus on the improvement of the missclassifications from previous applications. After each iteration, the method will actually bias training towards the points that are harder to predict.

Given the similar results among the algorithms that we used, as well as the fact that they are trained differently and plausibly sensitive to different characteristic features per class, we were motivated to combine their results to maximize the predictive power and to avoid potential biases. We chose to use a simple approach, with a classifier that averages the output probabilities from all three classifiers.

There are two ways to combine the outputs of the models, either through "hard" or "soft" voting. In the former case the prediction is based on the largest sum of votes across all models, while in the latter the class corresponding to the largest summed probability is returned.

To set an example with hard voting, if the results (from the three models we used) were BSG, BSG, and YSG then the final class would be BSG. However, this voting scheme does not grasp all the information. Soft voting can be more efficient. Given the final summed probability distribution across all classes it is possible that the final classification may be different than the one that the hard voting would return. Additionally, it can solve cases when a single class cannot be determined, i.e. when each of the three classifiers predicts a different class. With the final probability distribution we can also provide error estimates on the class predictions (and define confidence thresholds).

4.2.2. Combined classifier

The simplest approach is to combine the individual probabilities per class using equal weights per algorithm (since each one's accuracy is similar):

$$P_{\rm final} = (P_{\rm SVC} \times 1/3) + (P_{\rm RF} \times 1/3) + (P_{\rm NN} \times 1/3). \quad (1)$$

In Fig. 5 we show some example distributions for a few sources with correct and incorrect final classifications.

We performed a repeated (5 iterations) 5-fold CV test to estimate the overall accuracy of the combined approach at 0.83 ± 0.02 (see Table 5). The recall values are consistent with the results from the individual classifiers, with the highest success obtained for RSG ($\sim$ 94%), and BeBR and YSG ($\sim$ 80%), while LBV are not recovered. Despite this result, it is possible to get LBV classification at probably lower significance (i.e. probability). However, even a small number of candidates for this class are important for follow-up observations, due to their rarity and their critical role in outburst activity (e.g. Smith 2014).

In Fig. 6 we show the distributions of probabilities of the sources in a validation sample identified correctly (blue) and incorrectly (orange). The blue and orange dashed lines correspond to the mean probability values for the correct (at 0.86 ± 0.01) and incorrect (at 0.60 ± 0.03) classifications. Although the distributions are based on a single evaluation of the classifier on the validation set, the values corresponding to the these lines originate from a 5-iterations repeated 5-fold CV application.

4.3. Testing in other galaxies

As an independent test we used the collection of sources with known spectral types in the IC 1613, WLM, and Sextans A galaxies (see Section 2.3). In order to take into account all available information we resampled the whole M31 and M33 sample and we trained all three models.

The application follows the exact same protocol of the training, except from the resampling approach (which is used only for the training): i. load the photometric data for the new sources, ii. perform the necessary data processing to derive the features (color indices), iii. load the fully trained models for the three classifiers[10], iv. apply each of them to obtain the individual (per classifier) results, v. calculate the total probability distribution from which we get the final classification result, vi. compare the predictions with the original classes. For the last step we converted the original spectral types to the classes we have formed while training. Out of the 72 sources we excluded 9 with uncertain classification: 4 carbon stars, 2 identified simply as "emission" stars, one with "composite" spectrum, one classified as a GK star, and one M foreground star.

[10] Using Python's built-in persistence model `pickle` for saving and loading.

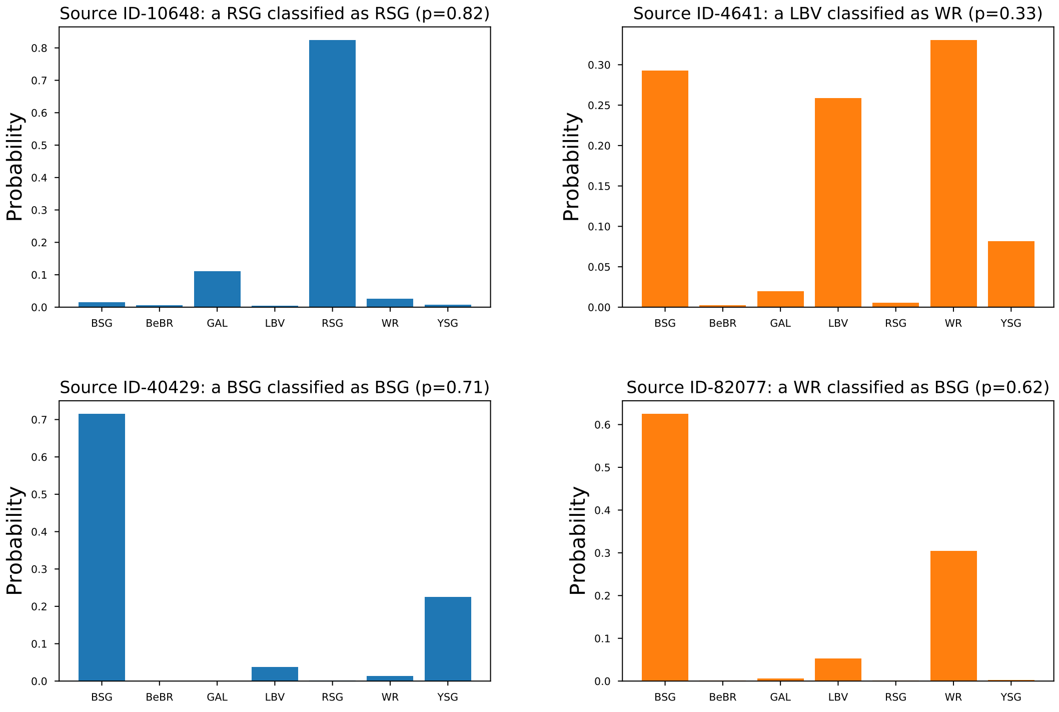


Fig. 5. Examples of probability distributions for a number of objects with correct (left) and incorrect (right) final classifications

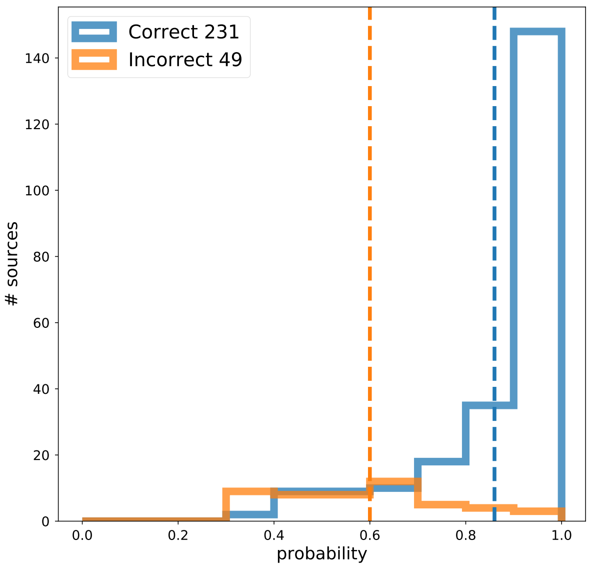
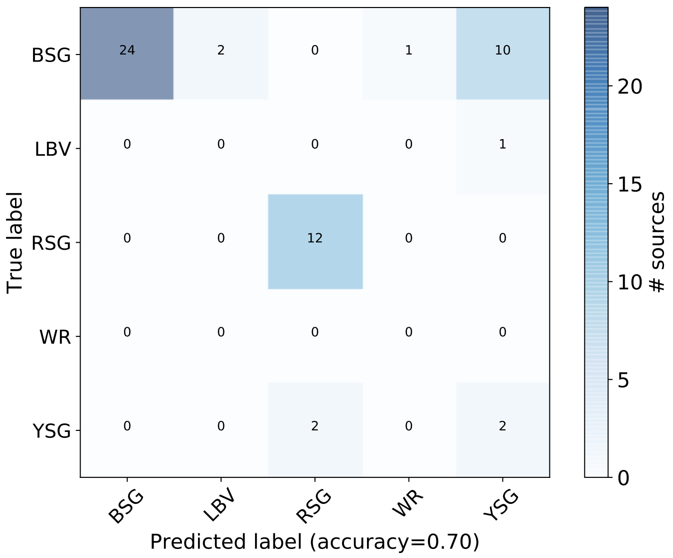


Fig. 6. Probability distributions for sources classified correctly (blue) and incorrectly (orange), for the validation sample. We are successfully recovering the majority of the objects in the validation sample (∼ 83%). The blue and orange dashed lines correspond to the mean probability values for the correct (at 0.86) and incorrect (at 0.60) classifications, based on repeated 5-fold CV (5 iterations).



	BSG	LBV	RSG	WR	YSG
BSG	24	2	0	1	10
LBV	0	0	0	0	1
RSG	0	0	12	0	0
WR	0	0	0	0	0
YSG	0	0	2	0	2

Fig. 7. The confusion matrix for 54 sources without missing values in the three galaxies (IC 1613, WLM, and Sextans A). We achieve an overall accuracy of ∼ 70%, and we
notice that the largest confusion occurs between BSG and YSG. The overall difference in the accuracy compared to that obtained with the M31 and M33 sample is attributed to the photometric errors, and the effect of metallicity and extinction in these galaxies.

In Fig. 7 we show the confusion matrix for the sample of the test galaxies, where we have additionally (for this plot) excluded another 9 sources with missing values (see next section). By doing this we can directly compare the results with what we have obtained from the training galaxies M31 and M33. We successfully recovered ∼ 70% which is less than what we achieved for the training (M31 and M33) galaxies (∼ 83%). We note that due to the very small sample size (54 sources) even a couple of missclasifications can change the overall accuracy by a few percents. Nevertheless, a difference is still present.

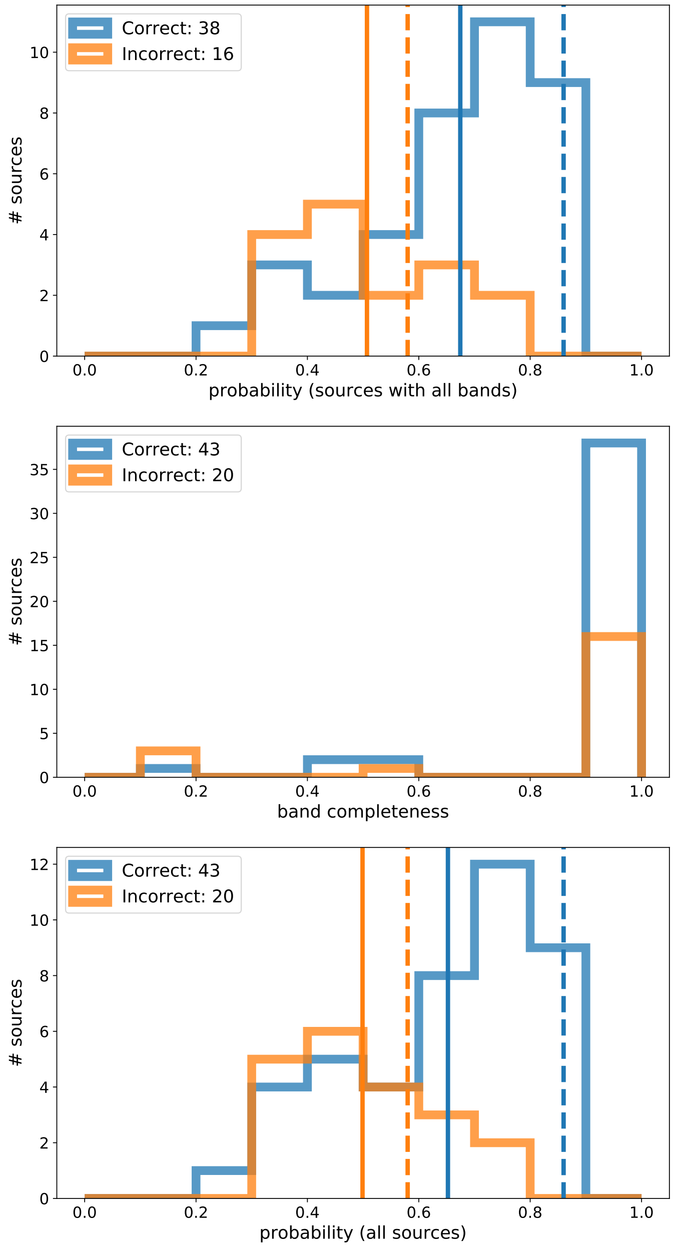


Fig. 8. Probability and band completeness distributions for the sources of the three galaxies (IC 1613, WLM, Sextans A) with and without missing data. (Top) The probability distributions of the correct (blue) and incorrect (orange) final classifications for the total sample of stars with known spectral types and with measurements in all bands. We achieved a recovery rate of ∼ 70%. The vertical dashed lines are the same with those in Fig. 6), while the solid ones correspond to the peak of the probability distributions for the current sample. (Middle) The distribution of the band completeness, i.e. the fraction of features without missing values. (Bottom) The probability distributions for all sources, including those without measurements in multiple bands (vertical lines have the same meaning with top panel). The success rate of ∼ 68%, is the same with the top panel, indicating the effectiveness of iterative imputor for missing data imputation.

Evidently, the largest disagreement arises from the prediction of most BSG as YSG. These two classes do not have a strict boundary in the HRD, making their classification, at larger distances, even more challenging. Moreover, the sources in these galaxies are at the faint end of the magnitude distribution for the *Spitzer* bands, which may influence the accuracy of their photometry.

While M31 has a metallicity above solar and M33 a gradient from solar to subsolar (Peña & Flores-Durán 2019) the three test galaxies are of lower metallicity (Boyer et al. 2015). However, it is not certain how this influences the classification performance. Lower metallicity affects both extinction and evolution that could lead to shifts in the intrinsic color distributions. Currently, given the lack of photometric data and source numbers for lower metallicity galaxies, it is impossible to examine the effect of metallicity thoroughly.

In the upper panel of Fig. 8 we show the distribution of the probabilities of correct (blue) and incorrect (orange) classifications. The dashed lines represent the same limits as defined in Section 4.2 for the training sample (at 0.86 and 0.60, respectively), while the solid ones correspond to the mean values defined by the current sample, at 0.67 and 0.51 for correct and incorrect, respectively. These shifts of the peak probabilities, especially for the correct classifications, shows the increased difficulty of the classifier to achieve a confident prediction.

4.4. Missing data imputation

In the previous section we excluded 9 sources which contain missing values, i.e. they do not have measurements in one or more bands. This is important for two reasons. In order for the methods to work they need to be fed with a value for each feature. Simultaneously, the majority of the sources in the catalogs with unclassified sources (to which this classifier will be applied) do not possess measurements in all bands.

To solve this, we performed a data imputation process in two ways. One, typical approach, is to replace missing values with a median/mean value. For this we first derived the median value (to remove extremes) of each feature distribution per class and from all available sources in the training sample of M31 and M33. Then we took the mean of the feature's values over all classes. Another approach is to use iterative imputation, in which each feature is modeled as a function of others originating from the Multivariate Imputation by Chained Equations (MICE; van Buuren & Groothuis-Oudshoorn 2011). This is a plausible assumption in our case, since we are dealing with spectral features which are indeed co-variant to some degree (spectra do not fluctuate suddenly across neighboring bands unless a peculiar emission/absorption feature is present). It is hence plausible to impute a missing bandwidth value given the others. The imputation of each feature is done sequentially, which allows the use of previous values to be considered as part of the model in predicting following features. The process is repeated (typically 10 times) which further allows to improve even more the missing values. We implemented this by using `impute.IterativeImputer()` (with default parameters).

To further investigate the influence of data with missing values we run a test by simulating sets with missing values form the original M31 and M33 sample. As usual, we split the sample into training (70%) and validation (30%) samples. After resampling the training set it is used to train the three classifiers, and an initial estimate of the accuracy is obtained with the validation sample. Then, according to how many features we can afford to "miss" we randomly select the features of each object in the validation sample. We either replace these features with the corresponding mean values or we apply the iterative impustor. Then the accuracy is obtained on this modified validation set (all objects contain missing values). In the upper panel of Fig. 9 we show an example of the difference between the initial (unmodi-

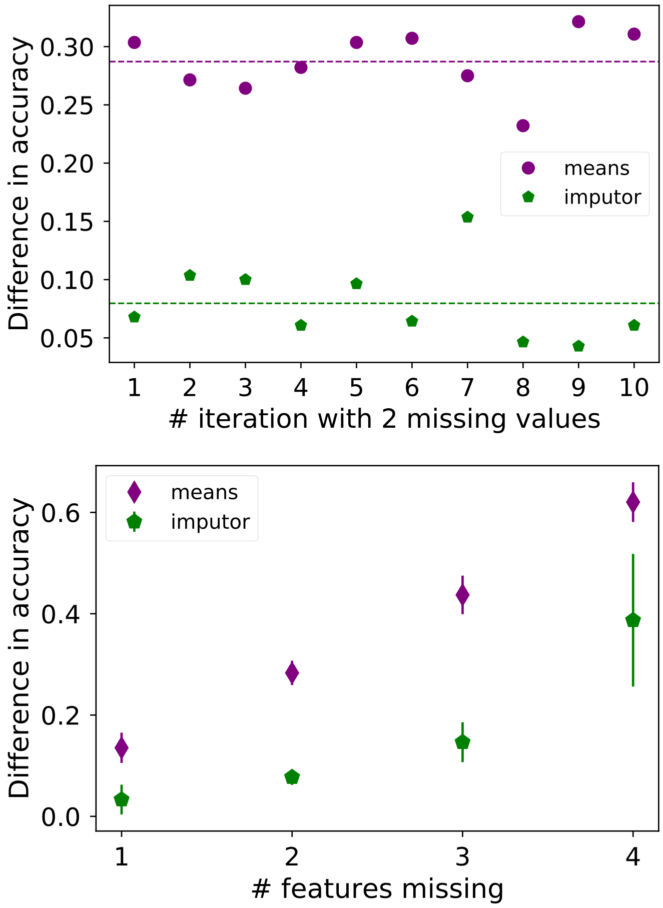


Fig. 9. Accuracy changes with missing features. (Top) Comparing the drop in accuracy from a typical (30% split) validation set without missing data to one where missing data have been generated by randomly selecting two features (per object) and replacing it with the corresponding mean values (purple circles) or the values imputed by the iterative imputor (green pentagons). The mean value obtained for the imputor is less than 0.1 and almost three times better than the mean drop for mean values. (Bottom) The iterative imputor is more capable to handle increased number of missing features, with a limit at three (out of 5 available in total). The loss in accuracy is less than 20%.

fied) validation set and the ones with missing values, by replacing randomly two features (per object) and imputing data with the iterative imputor (green pentagons) and mean values (purple circles). The mean drop in accuracy (over 10 iterations) is less than 0.1 for the imputor (green dashed line) but almost 0.3 for the means. In the bottom panel of Fig. 9 we show the drop in accuracy with increasing number of missing features. Obviously the imputor is performing more efficiently than simply replacement with mean values, and it can work up to three missing features (out of five available in total).

We also quantified the fraction of missing values by defining a "band completeness" term, simply as $1 - N_{\text{bands\_without\_measurement}}/N_{\text{total\_bands}}$. In the middle panel of Fig. 8 we show the distribution of this completeness for correct and incorrect sources. Given that about half of the 9 sources with missing values have band completeness 0.2 (meaning only one feature present) and the others are missing two to three, the success rate of 5 out of 9 of these sources classified correctly (∼ 55%) matches what we would expect approximately from the bottom panel of Fig. 9.

In the bottom panel of Fig. 8 we show now the probability distribution for all sources. The score is 68% which is the same as the accuracy obtained for the sample without any missing values (at 70%). The dashed and solid lines have the same meaning as previously, and there is no significant change (at 0.65 ad 0.59 for correct and incorrect classifications, respectively). In this particular dataset the presence of a small number of sources (9 out of 63; ∼ 14%) with missing values does not affect the performance of the classifier.

5. Discussion

In the following sections we discuss our results with respect to the sample sizes, label availability and feature sensitivity per class of our classifier.

5.1. *Exploring sample volumes and class mixing*

One of the major concerns when building machine-learning applications is the representativeness of the samples used. To explore this we performed iterative runs for each algorithm by adjusting the size of the training sample used.

At each iteration after the initial split into train/validation (70/30) sets, we kept a fraction of the training. After randomly selecting the sources per class, we performed the resampling in order to create the synthetic data. However, we needed at least two sources per class for the SMOTE to run (for this process we adjusted `n_neighbor=1`). Therefore, we started from 10% up to the complete training sample. Especially for LBV we added by hand an additional source for the first two fractions (after 0.3 enough sources were selected automatically).

In Fig. 10 we plot the recall per class for each method (for completeness in Fig. D.1 we also present precision and f1-score). We see an expected overall improvement with increasing sample size. This means that the larger the initial sample, the more representative it is of the parent population. The resampling method can interpolate, but does not extrapolate, which means that even though we are creating new sources they originate from the available information of the feature space. For example, Kyritsis et al. (2022) experimented with three different variants of RF to find that the results were dominated by the internal scatter of the features. Therefore, any limitations are actually transferred to the synthetic data. More information results in a better representation of their features by the classifier (leading to more accurate predictions).

5.1.1. BSG and RSG

BSG and RSG are the most populous classes and they achieve a high accuracy much faster (except for BSG in the SVC). RSG perform well also in the work of Dorn-Wallenstein et al. (2021), at ∼ 96%. In their refined label scheme they split (the equivalent to our) BSG sources into more classes which results in a poorer performance.

5.1.2. BeBR

The careful reader will notice that the sample size of BeBR is similar to that of the LBV and smaller than the WR one. Despite that, we are able to get really good results due to the specific nature of these objects. The B[e] phenomenon (presence of forbidden lines in spectra) actually includes a wide range of evolutionary stages and masses, from pre-main sequence stars to

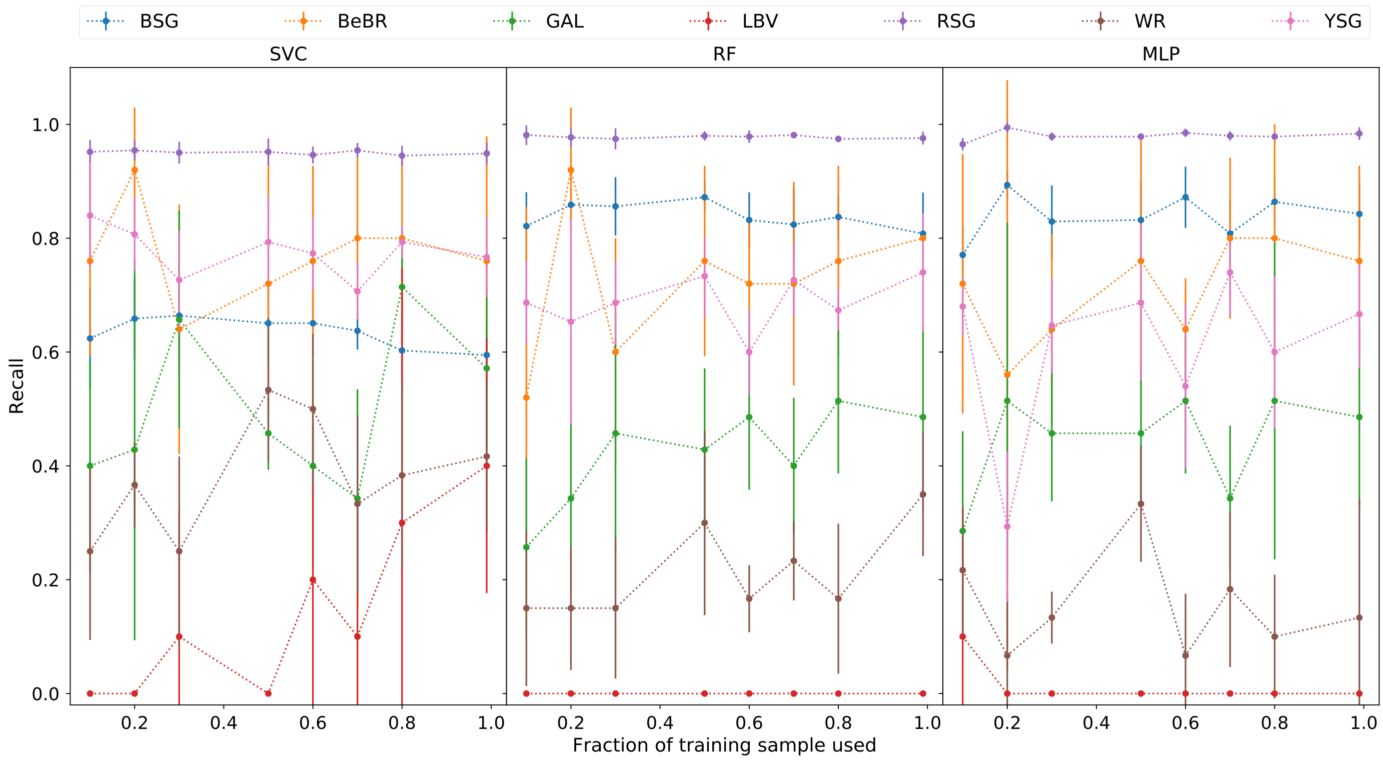


Fig. 10. Recall vs. the fraction of training sample used per class. We notice a significant improvement for BeBR, and YSG with increased training samples. When the samples sizes are already adequate the maximum possible value is achieved faster (e.g. RSG and BSG). GAL and WR show an increase while the LBV sample is too small to produce meaningful results.

evolved ones, symbiotics and planetary nebulae (Lamers et al. 1998). The sub-group of evolved stars is perhaps the most homogeneous group, as they are very luminous ($\log(L/L_\odot) > 6.0$) characterized by strong Balmer lines in emission (usually with P-Cygni profiles), narrow low-excitation lines (such as FeII, [FeII], and [OI]), and they display chemically processed material (such as TiO bands, ^{13}CO enrichment) indicative of their evolved nature. Moreover, these spectral characteristics (arising from dense, dusty disks of unknown origin) are generally long-lived (Maravelias et al. 2018) and these sources tend to be photometrically stable (Lamers et al. 1998). Those characteristics, along with strong IR excess due to their circumstellar dust (see Kraus 2019, but also Bonanos et al. 2009, 2010) make them a small but relatively robust class. Interestingly, Dorn-Wallenstein et al. (2021) recover BeBR at the same accuracy with our approach.

5.1.3. LBV

LBV show a clear gain with increased training sample, but only for SVC which is the most efficient method to recover this class. When in quiescence, LBV share similar observables (e.g. colors, spectral features) with BSG, WR, and BeBR (e.g. Weis & Bomans 2020; Smith 2014). Therefore, it is quite challenging to separate them and the only way to certify the nature of these candidate LBV is when they actually enter an active outburst phase. During this, the released material obstructs the central part of the star, changing its spectral appearance from a O/B-type to an A/F-type (which in turn would mix them with the YSG sources), while they can significantly brighten in the optical (> 2 mag, but at constant bolometric luminosity; Clark et al. 2005). In order to form the most secure LBV sample we excluded all candidate LBV (some of which resemble more BeBR; Kraus 2019), and we were left with a very small sample of 6 stars. LBV display additional photometric variability of at least a few tenths of a magnitude (Clark et al. 2005). This information could be included as a supplementary feature through a variability index (such as χ^2, median absolute deviation, etc; Sokolovsky et al. 2017). However, this is not currently possible as the datasets we are using are very limited in the epoch coverage (for example, at the very best only a few points are available per band in the Pan-STARRS survey). Furthermore, the optical (Pan-STARRS) and IR (*Spitzer*) data for the same source have been obtained at different epochs which may result in sampling the source's flux from different modes. This effect, along with their small sample size (far from complete; Weis & Bomans 2020), may well explain the limited prediction capability of our method. On the other hand, Dorn-Wallenstein et al. (2021) took into account variability (using *WISE* lightcurves) and they report a full recovery of LBV, which might be due to overfitting. Because of the small size of their sample (2 sources) they did not discuss it any further.

5.1.4. WR

In the single case scenario, LBV are a phase in the transition of O-type stars before their outer layers are stripped, because of the intense mass loss and/or massive eruptions (Smith 2014). Binaries are another channel where efficient stripping can lead to WR stars (Shenar et al. 2020). Depending on the metallicity and their rotation, WR may also form directly from main-sequence stars (Meynet & Maeder 2005). As their evolution is highly uncertain, they can originate from both LBV or BSG stars. Stellar evolution is a continuous process which does not display strict boundaries between those groups in the HRD. Therefore, their features (color indices) can be mixed. They are bright sources and this has enabled the detection of almost their complete population (see Neugent & Massey 2019 for a review) but the actual numbers are limited due to their rarity. Their small sample size, which actually includes a number of different subtypes of WR (such as WN and WC, as well as some known binaries with O-type companions) has an impact on our prediction capability, but it is better than LBV. We also note that their recall benefits from the increase in the training sample for SVC and RF, but not much for MLP. Rosslowe & Crowther (2018) have shown that WR and LBV can be better distinguished using near-IR (JHK bands), a region which is unfortunately excluded from our feature list, because of the lack of extensive and consistent surveys for our galaxies (although 2MASS exists it is not deep enough for our more distant galaxies). On the contrary, Dorn-Wallenstein et al. (2021) include these bands which may explain their improved accuracy for WR and (possibly) for LBV.

5.1.5. YSG

The YSG class contains all sources that are found in between the BSG and the RSG. In general, this is a relatively short-lived phase as the star evolves off the main-sequence or evolves back to hotter phases, after the RSG phase (e.g. Kourniotis et al. 2018; Gordon & Humphreys 2019; excluding the contamination by foreground sources which we have minimized by pre-processing with the *Gaia* properties, but definitely not eliminated). However, it is hard (if not impossible) to get strict boundaries in the color-magnitude diagrams (CMDs) between the BSG/YSG and YSG/RSG populations. Yang et al. (2019) presents a ranking scheme which is based on the presence of each source in a number of multiple CMDs (c.f. fig. 16). With our current work we are able to remove this complexity as we take into account the information from multiple CMDs (through the color indices) at once. We are able to correctly predict the majority of this sample at ~ 73%, in contrast to the ~ 27% from Dorn-Wallenstein et al. (2021). The major factor in this case is the use of more optical colors that help distinguishing YSG with BSG more effectively, while Dorn-Wallenstein et al. (2021) work mainly with IR colors.

5.2. Label uncertainties

Uncertainty in the labels (the spectral types / classes) can come in two flavors, either because of classification errors (e.g. human bias, instrument limitations) or due to the natural mixing of these sources. After all, there are uncertainties in the evolution of massive stars after the main-sequence, as we still lack robust knowledge with respect to the transition of these sources through the various phases. However, it is a typical prerequisite in supervised machine-learning applications that the labels/classes are the absolute truth. This can lead to inaccurate predictions. Dorn-Wallenstein et al. (2021) comment specifically on this, as with their refined classes (containing 12 classes) they achieve an accuracy of ~ 53% for the SVC, because their labels for Galactic sources are "derived inhomogeneously, and many are from spectroscopy that is now more than 50 years old". In our case, we have obtained a more homogeneous sample, since we are working with specific galaxies (distance uncertainties are minimized) and the results originate from consistent surveys and modern instruments/facilities. In other words, our labels are more secure and help us achieve a better result. A way to tackle this is by properly handling label uncertainties *during the training process* which, nonetheless, is not trivial to implement.

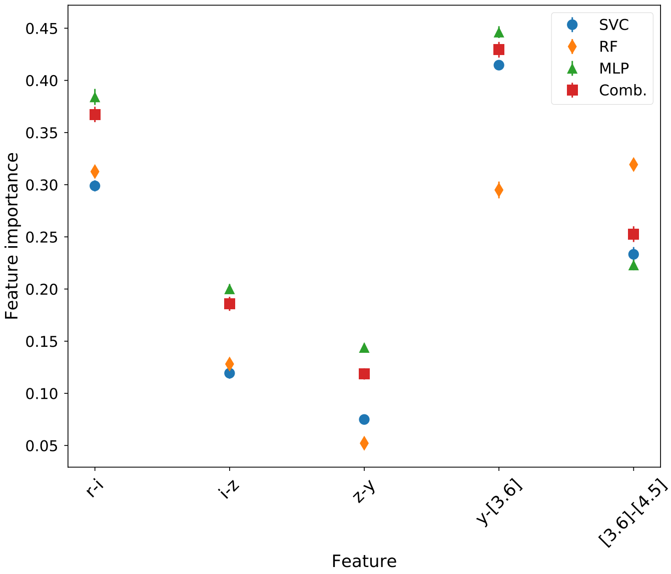


Fig. 11. Permutation feature importance (i.e. the difference in accuracy between the original dataset and the shuffled one) per feature for each classifier independently and the combined one. The features $r - i$, $y -$ [3.6], and [3.6] − 4.5 appear to be the most important consistently.

5.3. Feature sensitivity

During the feature selection (Section 3.4) we disregarded bands that would significantly decrease our sample and/or they would introduce noise (J_{UK}, *Gaia*, *Spitzer* [5.8], [8.0], and [24], Pan-STARRS g). The *Spitzer* [3.6] and [4.5] bands are present for all of our sources (by construction of our catalogs) while the availability of optical ones (Pan-STARRS r, i, z, y) vary depending on the source. In order not to lose any more information we included all optical bands (except for g), and perform missing data imputation whenever necessary. How sensitive is the classifier to these features, and which are more important per class?

We first investigate how the overall performance of the classifier depends on the features. For this we performed a permutation feature importance test (`sklearn.inspection.permutation_importance()`). By shuffling the values of a specific feature we see how much it influences the final result. In this case the metric used is the difference between the accuracy of the original dataset and of the shuffled one[11]. In the case of a non-significant feature this change will be small and the opposite holds for an important feature. In Fig. 11 we show the results per classifier as well as their combined model ('all'). We notice that the most significant features are $r - i$, $y -$ [3.6], and [3.6] − [4.5] while the least important are $i - z$ and $z - y$ and [4.5] − [5.8]. This is not a surprise actually since these features are the ones for which we have the largest separation among the averaged lines of classes (see Fig. 2). There are small differences between the individual algorithms but they are relatively consistent. They show similar sensitivity to the optical colors. The only exception is RF for $y -$ [3.6] which seems less sensitive than the others.

[11] The process is performed on the training sample, so we include all M31 and M33 sources into this, and resampled accordingly.

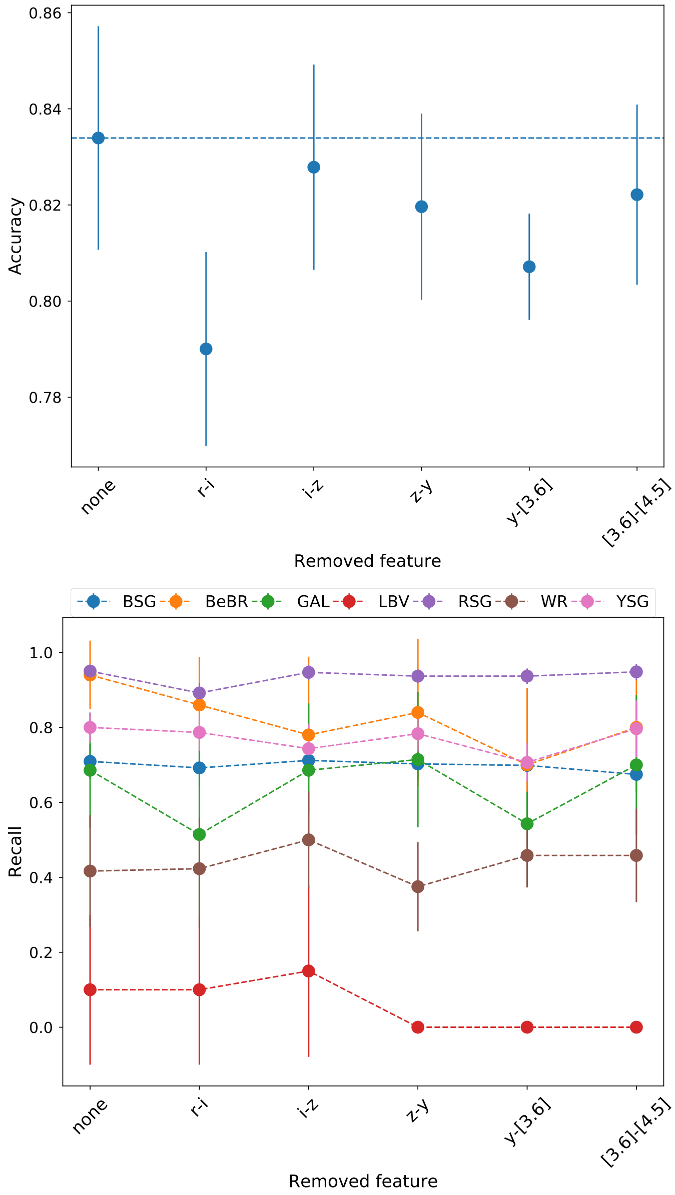


Fig. 12. Feature importance per removed feature. The feature is removed and the whole combined model is retrained. The first point corresponds to the full feature set. (Top) Considering the overall accuracy for the combined classifier the most significant features are $r - i$ and $y - [3.6]$, consistent with the permutation importance test. (Bottom) The recall per class, where we see that different features become more significant for BeBR, GAl, WR, YSG, and LBV, while RSG and BSG do not show important changes (see text for more).

One key issue with this approach is that it is more accurate in the case of uncorrelated data. In our case there is some correlation both due to the fact that the consecutive color indices contain their neighboring bands and because the fluxes at each band are not totally independent from the others. An alternative and more robust way is by testing the accuracy of our model by dropping a feature each time. The general drawback in this approach is the computational time as it is needed to retrain (including resampling at each iteration) the model from the beginning, contrary to the previous test where only the values of a feature change and the model is applied. Fortunately, our training sample and modeling is neither that large or complicated to prohibit us. Thus, using the combined classifier we iteratively removed one feature at the time. Then we calculated the metric of this iteration with respect to the initial feature set. In Fig. 12 (upper panel) we plot this accuracy, where $r - i$ and $y - [3.6]$ show (relatively) larger deviation and seem to be the most important features. This is in agreement with what we found with the feature permutation approach. Interestingly, $r - i$ is the "blue-est" feature and seems to be important for the overall classification (the optical part is excluded from the work of Dorn-Wallenstein et al. 2021).

When examining the results for the recall per class (Fig. 12; lower panel) we see that for different classes are sensitive to different features. For BeBR $i - z$ and $y - [3.6]$ seem to be the most important, although smaller offsets are visible for the rest of the features also (the mean curve of BeBR peaks at this feature; see Fig. 2 This can be attributed to the overall redder colors because of the dusty environment around these objects. GAL are sensitive to both $r - i$, the feature closer to the optical part, and $y - [3.6]$, partly due to the PAH component (GAL display the second strongest peak in Fig. 2). Although not so significant $i - z$ seems to favor WR classification. WR is a collection of different flavors or classical (evolved) WR stars, including binary systems. YSG are more sensitive to $y - [3.6]$ and a bit less in $i - z$, similar to BeBR, as they also tend to have dusty environments. BSG and RSG are the most populated classes and they do not show any significant dependence. This might be because although distinct to the other classes they contain a wider range of objects that possibly mask significant differences between the bands (see Bonanos et al. 2009, 2010). For example, we have included in the BSG sources with emission lines, such as Be stars that display redder colors. For LBV $i - z$ seems important but due to their small population the error is quite significant. Also the redder features lie at zero which may be due to the incapability of our model to predict these sources with higher confidence. If we were to exclude any of these features, we would get poorer results for some of the classes. The inclusion of more colors would benefit the performance of our classifier as it would help with the sampling of the spectral energy distributions of the sources (going to the optical blue part will not help the redder sources but it would be valuable for the hotter classes).

6. Summary and Conclusions

In this work we present the application of machine-learning algorithms to build an ensemble photometric classifier for the classification of massive stars in nearby galaxies. We compiled a *Gaia* cleaned selected sample of 932 M31 and M33 sources and we grouped their spectral types into 7 classes (representing Blue, Yellow, Red, and B[e] supergiants, Luminous Blue Variables, Wolf-Rayet, and background sources as outliers). To address the imbalance of the sample, we employed a synthetic data approach with which we managed to increase the underrepresented classes, although this is always limited by the feature space that the initial sources sample. We used as features the consecutive color indices from [3.6] and [4.5] *Spitzer*, and r, i, z, y Pan-STARRS bands (not corrected for extinction). We implemented three well-known supervised machine-learning algorithms: Support Vector Classification, Random Forest, and Multi-layer Perceptron, to develop our classifier. The application of each of the algorithms results in fairly good overall results (recovery rate): BSG, GAL, and YSG from ~ 60% to ~ 80%), BeBR at ~ 73 – 80%, WR at ~ 45%, with best results obtained for the RSG (~ 94%) and the worst for LBV (~ 28% for SVC only). These results are in par or improved compared to the results by Dorn-Wallenstein et al. (2021), who are working with a much less homogeneous with respect to the labels (but more populated) Galactic sample. Given the similar performance of the three methods and to maximize our prediction capability we combined all outputs into a single probability distribution. This final meta-classifier achieved a similar overall (weighted balanced) accuracy (∼ 83%), and similar good results per class.

By searching the impact of the training volume size we noticed that the sample size plays a critical role (as expected) in the accurate prediction of a class. When many sources of a class are available (e.g. RSG, BSG) then the classifier works efficiently. In less populated classes (such as BeBR, WR) the inclusion of more objects increases the information provided to the classifier, and improves the prediction ability. However, we are hampered by low-number statistics as these classes correspond to rare and/or short-lived phases.

Additional information can be retrieved by using more features. We investigated the feature importance to find that for the current dataset $r-i$ and $y-[3.6]$ are the most important, although different classes are sensitive to different features. Thus, the inclusion of more color indices (i.e. observations at different bands) could improve the separation of the classes.

To test our classifier with an independent sample we used data collected for IC 1613, WLM, and Sextans A sources, some of which (∼ 14%) had missing values. We performed data imputation by replacing the features' values using means and an iterative imputor. Although the missing values do not affect significantly the results for this particular dataset, further tests showed that the iterative imputor can efficiently handle datasets with up to 3 missing features (out of 5 in total available). The final obtained accuracy is ∼ 70%, lower than what we achieved for M31 and M33. The discrepancy can be attributed partly to photometric issues and to the total effect of metallicity. The latter can modify the intrinsic colors of the sources and extinction due to the different galactic environments. Despite that the result from this application is promising. In a follow-up paper we will present in detail its application to previously unclassified sources for a large number of nearby galaxies.

Currently, the metallicity dependence is impossible to address. For this we need larger samples of well-characterized sources and in different metallicity environments. Although challenging, because of the observing time required in large facilities, the ASSESS team is actively working towards this direction. A number of observing spectroscopic campaigns are completed and ongoing, which they will provide the ultimate test-bed of our classifier's actual performance along with opportunities for improvement.

Acknowledgements We thank the anonymous referee for their constructive comments and suggestions that helped us to improve this work. GM, AZB, FT, SdW, MY acknowledge funding support from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (Grant agreement No. 772086). GM would like to thank Alkis Simitsis, Thodoris Bitsakis, Elias Kyritsis, Andreas Zezas, Jeff Andrews, Konstantinos Paliouras for many fruitful discussions on machine learning and beyond.

Facilities: This work has made use of data from the European Space Agency (ESA) mission *Gaia* (`https://www.cosmos.esa.int/gaia`), processed by the *Gaia* Data Processing and Analysis Consortium (DPAC, `https://www.cosmos.esa.int/web/gaia/dpac/consortium`). Funding for the DPAC has been provided by national institutions, in particular the institutions participating in the *Gaia* Multilateral Agreement. The Pan-STARRS1 Surveys (PS1) and the PS1 public science archive have been made possible through contributions by the Institute for Astronomy, the University of Hawaii, the Pan-STARRS Project Office, the Max-Planck Society and its participating institutes, the Max Planck Institute for Astronomy, Heidelberg and the Max Planck Institute for Extraterrestrial Physics, Garching, The Johns Hopkins University, Durham University, the University of Edinburgh, the Queen's University Belfast, the Harvard-Smithsonian Center for Astrophysics, the Las Cumbres Observatory Global Telescope Network Incorporated, the National Central University of Taiwan, the Space Telescope Science Institute, the National Aeronautics and Space Administration under Grant No. NNX08AR22G issued through the Planetary Science Division of the NASA Science Mission Directorate, the National Science Foundation Grant No. AST-1238877, the University of Maryland, Eotvos Lorand University (ELTE), the Los Alamos National Laboratory, and the Gordon and Betty Moore Foundation. The UHS is a partnership between the UK STFC, The University of Hawaii, The University of Arizona, Lockheed Martin and NASA.

Software: This research made use of Numpy (Harris et al. 2020), matplotlib (Hunter 2007), sklearn (Pedregosa et al. 2011), Jupyter Notebooks (Kluyver et al. 2016), Mlxtend (Raschka 2018). This research made use of TOPCAT, an interactive graphical viewer and editor for tabular data (Taylor 2005).

We wish to thank the "2019 Summer School for Astrostatistics in Crete"[12] for providing training on the statistical methods adopted in this work. We also thank Jeff Andrews for organizing the SMAC (Statistical methods for Astrophysics in Crete) seminar[13]. We also acknowledge useful information provided by Jason Brownlee from his site Machine Learning Mastery[14]

This research has made use of NASA's Astrophysics Data System. This research has made use of the SIMBAD database, operated at CDS, Strasbourg, France. This research has made use of the SVO Filter Profile Service (http://svo2.cab.inta-csic.es/theory/fps/) supported from the Spanish MINECO through grant AYA2017-84089.

References

Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, software available from tensorflow.org

Arnason, R. M., Barmby, P., & Vulic, N. 2020, MNRAS, 492, 5075

Ball, N. M. & Brunner, R. J. 2010, International Journal of Modern Physics D, 19, 1049

Baron, D. 2019, arXiv e-prints [`1904.07248v1`]

Ben-Hur, A., Horn, D., Siegelmann, H. T., & Vapnik, V. 2002, J. Mach. Learn. Res., 2, 125–137

Bonanos, A. Z., Lennon, D. J., Köhlinger, F., et al. 2010, AJ, 140, 416

Bonanos, A. Z., Massa, D. L., Sewilo, M., et al. 2009, AJ, 138, 1003

Boyer, M. L., McQuinn, K. B. W., Barmby, P., et al. 2015, ApJS, 216, 10

Breiman, L. 2001, Machine Learning, 45, 5

Bresolin, F., Pietrzyński, G., Urbaneja, M. A., et al. 2006, ApJ, 648, 1007

Bresolin, F., Urbaneja, M. A., Gieren, W., Pietrzyński, G., & Kudritzki, R.-P. 2007, ApJ, 671, 2028

Britavskiy, N. E., Bonanos, A. Z., Mehner, A., Boyer, M. L., & McQuinn, K. B. W. 2015, A&A, 584, A33

Britavskiy, N. E., Bonanos, A. Z., Mehner, A., et al. 2014, A&A, 562, A75

Bruhweiler, F. C., Miskey, C. L., & Smith Neubig, M. 2003, AJ, 125, 3082

Bruzual, G. & Charlot, S. 2003, MNRAS, 344, 1000

Camacho, I., Garcia, M., Herrero, A., & Simón-Díaz, S. 2016, A&A, 585, A82

Chambers, K. C., Magnier, E. A., Metcalfe, N., et al. 2016, arXiv e-prints, arXiv:1612.05560

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. 2002, J. Artif. Int. Res., 16, 321–357

Chollet, F. et al. 2015, Keras, `https://keras.io`

Clark, J. S., Larionov, V. M., & Arkharov, A. 2005, A&A, 435, 239

Cortes, C. & Vapnik, V. 1995, Machine Learning, 20, 273

Davenport, J. R. A., Ivezić, Ž., Becker, A. C., et al. 2014, MNRAS, 440, 3430

Davis, J. & Goadrich, M. 2006, in Proceedings of the 23rd International Conference on Machine Learning, ICML '06 (New York, NY, USA: Association for Computing Machinery), 233–240

de Mink, S. E., Sana, H., Langer, N., Izzard, R. G., & Schneider, F. R. N. 2014, ApJ, 782, 7

Dorn-Wallenstein, T. Z., Davenport, J. R. A., Huppenkothen, D., & Levesque, E. M. 2021, ApJ, 913, 32

Drout, M. R., Massey, P., & Meynet, G. 2012, ApJ, 750, 97

[12] `http://astro.physics.uoc.gr/Conferences/Astrostatistics_School_Crete_2019/`

[13] `https://githubhelp.com/astroJeff/SMAC`

[14] `https://machinelearningmastery.com`

Drout, M. R., Massey, P., Meynet, G., Tokarz, S., & Caldwell, N. 2009, ApJ, 703, 441

Dunstall, P. R., Dufton, P. L., Sana, H., et al. 2015, A&A, 580, A93

Dye, S., Lawrence, A., Read, M. A., et al. 2018, MNRAS, 473, 5113

Ekström, S., Georgy, C., Eggenberger, P., et al. 2012, A&A, 537, A146

Eldridge, J. J., Stanway, E. R., Xiao, L., et al. 2017, PASA, 34, e058

Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, A&A, 616, A1

Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., et al. 2016, A&A, 595, A1

Garcia, M. & Herrero, A. 2013, A&A, 551, A74

Georgy, C., Ekström, S., Granada, A., et al. 2013, A&A, 553, A24

González, J. A. & Guzmán, F. S. 2019, Phys. Rev. D, 99, 103002

Gordon, M. S. & Humphreys, R. M. 2019, Galaxies, 7, 92

Gordon, M. S., Humphreys, R. M., & Jones, T. J. 2016, ApJ, 825, 50

Grandini, M., Bagli, E., & Visani, G. 2020, arXiv e-prints, arXiv:2008.05756

Gvaramadze, V. V., Kniazev, A. Y., & Fabrika, S. 2010, MNRAS, 405, 1047

Harris, C. R., Millman, K. J., van der Walt, S. J., et al. 2020, Nature, 585, 357

Herrero, A., Garcia, M., Uytterhoeven, K., et al. 2010, A&A, 513, A70

Humphreys, R. M., Gordon, M. S., Martin, J. C., Weis, K., & Hahn, D. 2017, ApJ, 836, 64

Humphreys, R. M., Weis, K., Davidson, K., Bomans, D. J., & Burggraf, B. 2014, ApJ, 790, 48

Hunter, J. D. 2007, Computing In Science & Engineering, 9, 90

Jayasinghe, T., Kochanek, C. S., Stanek, K. Z., et al. 2018, MNRAS, 477, 3145

Kaufer, A., Venn, K. A., Tolstoy, E., Pinte, C., & Kudritzki, R.-P. 2004, AJ, 127, 2723

Khan, R. 2017, ApJS, 228, 5

Khan, R., Stanek, K. Z., Kochanek, C. S., & Sonneborn, G. 2015, ApJS, 219, 42

Kingma, D. P. & Ba, J. 2014, arXiv e-prints, arXiv:1412.6980

Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, in Positioning and Power in Academic Publishing: Players, Agents and Agendas, ed. F. Loizides & B. Scmidt (IOS Press), 87–90

Kopsacheili, M., Zezas, A., & Leonidaki, I. 2020, MNRAS, 491, 889

Kourniotis, M., Kraus, M., Arias, M. L., Cidale, L., & Torres, A. F. 2018, MNRAS, 480, 3706

Kraus, M. 2019, Galaxies, 7, 83

Kyritsis, E., Maravelias, G., Zezas, A., et al. 2022, A&A, 657, A62

Lamers, H. J. G. L. M., Zickgraf, F.-J., de Winter, D., Houziaux, L., & Zorec, J. 1998, A&A, 340, 117

Lemaître, G., Nogueira, F., & Aridas, C. K. 2017, Journal of Machine Learning Research, 18, 1

Levesque, E. M. & Massey, P. 2012, AJ, 144, 2

Lindegren, L., Hernández, J., Bombrun, A., et al. 2018, A&A, 616, A2

Makhija, S., Saha, S., Basak, S., & Das, M. 2019, Astronomy and Computing, 29, 100313

Maraston, C. 2005, MNRAS, 362, 799

Maravelias, G., Kraus, M., Cidale, L. S., et al. 2018, MNRAS, 480, 320

Martin, J. C. & Humphreys, R. M. 2017, AJ, 154, 81

Martins, F. & Palacios, A. 2013, A&A, 560, A16

Massey, P. 1998, ApJ, 501, 153

Massey, P., Bianchi, L., Hutchings, J. B., & Stecher, T. P. 1996, ApJ, 469, 629

Massey, P. & Johnson, O. 1998, ApJ, 505, 793

Massey, P., McNeill, R. T., Olsen, K. A. G., et al. 2007, AJ, 134, 2474

Massey, P., Neugent, K. F., & Levesque, E. M. 2019, AJ, 157, 227

Massey, P., Neugent, K. F., & Smart, B. M. 2016, AJ, 152, 62

Massey, P., Olsen, K. A. G., Hodge, P. W., et al. 2006, AJ, 131, 2478

Massey, P., Silva, D. R., Levesque, E. M., et al. 2009, ApJ, 703, 420

McCulloch, W. S. & Pitts, W. 1943, The bulletin of mathematical biophysics, 5, 115

Mehta, P., Bukov, M., Wang, C.-H., et al. 2019, Phys. Rep., 810, 1

Meynet, G. & Maeder, A. 2005, A&A, 429, 581

Möller, A., Ruhlmann-Kleider, V., Leloup, C., et al. 2016, J. Cosmology Astropart. Phys., 2016, 008

Morello, G., Morris, P. W., Van Dyk, S. D., Marston, A. P., & Mauerhan, J. C. 2018, MNRAS, 473, 2565

Muthukrishna, D., Parkinson, D., & Tucker, B. E. 2019, ApJ, 885, 85

Naul, B., Bloom, J. S., Pérez, F., & van der Walt, S. 2018, Nature Astronomy, 2, 151

Neugent, K. & Massey, P. 2019, Galaxies, 7, 74

Neugent, K. F., Levesque, E. M., Massey, P., & Morrell, N. I. 2019, ApJ, 875, 124

Neugent, K. F. & Massey, P. 2011, ApJ, 733, 123

Neugent, K. F., Massey, P., & Georgy, C. 2012, ApJ, 759, 11

Neugent, K. F., Massey, P., Skiff, B., et al. 2010, ApJ, 719, 1784

Odewahn, S. C., Stockwell, E. B., Pennington, R. L., Humphreys, R. M., & Zumach, W. A. 1992, AJ, 103, 318

Owocki, S. P. & Puls, J. 1999, ApJ, 510, 355

Pashchenko, I. N., Sokolovsky, K. V., & Gavras, P. 2018, MNRAS, 475, 2326

Peña, M. & Flores-Durán, S. N. 2019, Rev. Mexicana Astron. Astrofis., 55, 255

Pedregosa, F., Varoquaux, G., Gramfort, A., et al. 2011, Journal of Machine Learning Research, 12, 2825

Peters, G. J. & Hirschi, R. 2013, The Evolution of High-Mass Stars, ed. T. D. Oswalt & M. A. Barstow, Vol. 4 (Springer Netherlands), 447

Plewa, P. M. 2018, MNRAS, 476, 3974

Raschka, S. 2018, The Journal of Open Source Software, 3

Re, M. & Valentini, G. 2012, Ensemble Methods, ed. M. J. Way, J. D. Scargle, K. M. Ali, & A. N. Srivastava, 563–593

Rosslowe, C. K. & Crowther, P. A. 2018, MNRAS, 473, 2853

Saito, T. & Rehmsmeier, M. 2015, PLOS ONE, 10, 1

Sana, H., de Koter, A., de Mink, S. E., et al. 2013, A&A, 550, A107

Sana, H., de Mink, S. E., de Koter, A., et al. 2012, Science, 337, 444

Schlafly, E. F. & Finkbeiner, D. P. 2011, ApJ, 737, 103

Sharma, K., Kembhavi, A., Kembhavi, A., et al. 2020, MNRAS, 491, 2280

Shenar, T., Gilkis, A., Vink, J. S., Sana, H., & Sander, A. A. C. 2020, A&A, 634, A79

Smith, N. 2014, ARA&A, 52, 487

Sokolovsky, K. V., Gavras, P., Karampelas, A., et al. 2017, MNRAS, 464, 274

Storrie-Lombardi, M. C., Lahav, O., Sodre, L., J., & Storrie-Lombardi, L. J. 1992, MNRAS, 259, 8P

Taylor, M. B. 2005, in Astronomical Society of the Pacific Conference Series, Vol. 347, Astronomical Data Analysis Software and Systems XIV, ed. P. Shopbell, M. Britton, & R. Ebert, 29

Tomek, I. 1976, IEEE Transactions on Systems, Man, and Cybernetics, SMC-6, 769

van Buuren, S. & Groothuis-Oudshoorn, K. 2011, Journal of Statistical Software, 45, 1–67

Wachter, S., Mauerhan, J. C., Van Dyk, S. D., et al. 2010, AJ, 139, 2330

Weis, K. & Bomans, D. J. 2020, Galaxies, 8, 20

Williams, S. J. & Bonanos, A. Z. 2016, A&A, 587, A121

Wilson, D. L. 1972, IEEE Transactions on Systems, Man, and Cybernetics, SMC-2, 408

Yang, M., Bonanos, A. Z., Jiang, B.-W., et al. 2019, A&A, 629, A91

Zhang, S., Yang, J., Xu, Y., et al. 2020, ApJS, 248, 15

Table A.1. Sources with known spectral types.

ID	RA (J2000) (deg)	Dec (J2000) (deg)	SpType	Ref
M31-1	9.26500	40.33747	RSG:	(1)
M31-2	9.27583	40.02225	AI	(2)
M31-3	9.30000	39.91256	YSG:	(2)
M31-4	9.35208	40.30647	RSG:	(1)
M31-5	9.35667	40.12550	YSG:	(2)
M31-6	9.37083	40.33547	B1I	(2)
M31-7	9.38875	40.01014	B2Ib-B5I	(2)
M31-8	9.39250	40.01250	RSG	(2)
M31-9	9.39333	40.02131	B0I	(2)
M31-10	9.41625	39.97739	M1Ia	(2)
M31-11	9.43875	39.97319	F5Ia	(2)
M31-12	9.46625	39.98383	A2Ib	(2)
M31-13	9.59500	40.54425	B9.5I+Dbl:	(2)
M31-14	9.62167	40.51672	RSG:	(1)
M31-15	9.63042	40.54169	RSG:	(1)
M31-16	9.63458	40.51069	B7I	(2)
M31-17	9.73610	40.57960	QSO	(3)
M31-18	9.73833	40.52558	WN7:+Neb	(2)
M31-19	9.73875	40.68153	M2I	(2)
M31-20	9.75917	40.65200	M1I	(2)

References: (1) Gordon et al. (2016), (2) Massey et al. (2016), (3) Massey et al. (2019).
Note: The table is available in its entirety in CDS.

Appendix A: List of classified sources

In this Section we provide in detail the list of sources with spectral classifications presented in Table 1. For each source we provide a simple identification id with the galaxy and an increasing number (column 1), RA and Dec (columns 2 and 3) as obtained from their corresponding source, the spectral type (column 4) and the reference (column 5).

Only the first rows of the catalog is provided here for guidance, as it is published in its entirety in CDS.

Appendix B: *Gaia* processing plots for M33

Similar to Fig. 1 we present the corresponding plots for M33 in Fig. B.1.

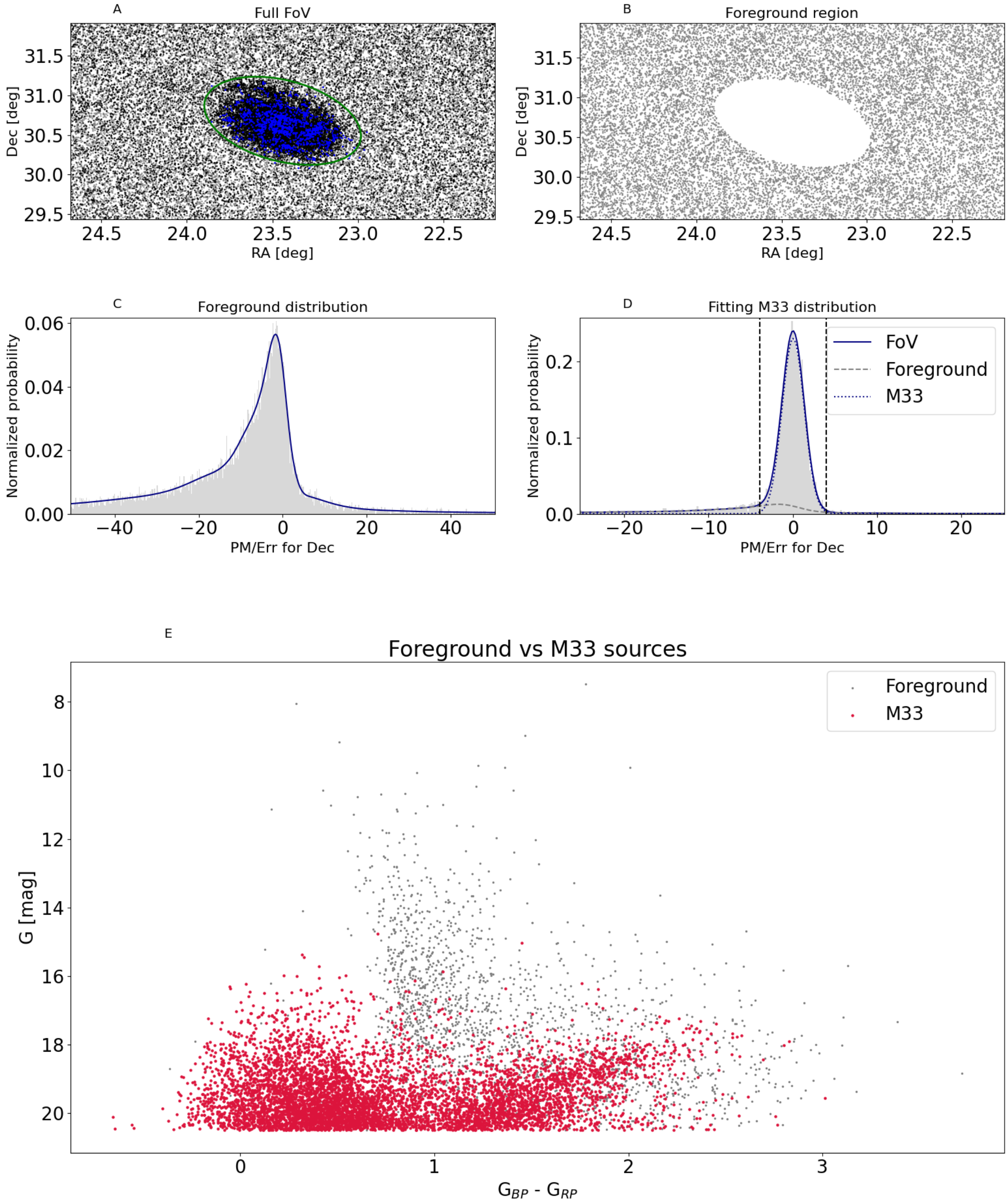


Fig. B.1. Using *Gaia* to identify and remove foreground sources for M33. (A) The field-of-view of *Gaia* sources (black dots) for M33, with the green ellipse marking the galaxy's boundary. The blue dots highlight the sources with known spectral classification in M33. (B) The foreground region for M33. (C) The distribution of the proper motion over its error for Dec, for all *Gaia* sources in the foreground region (fitted with a spline function). (D) The distribution of the proper motion over its error for Dec (solid line), for all sources along the line-of-sight of M33, including both foreground and galactic (M33) sources. The dashed line refers to the scaled spline (accounting for the number of foreground sources expected inside M33) and the dotted line to a Gaussian function. The vertical dashed lines correspond to the 3σ threshold of the Gaussian. Any source with values outside this region is flagged as a potential foreground source. (E) The *Gaia* CMD of all sources identified as galactic (red points) and foreground (gray). The majority of the foreground sources lie on the yellow branch of the CMD which is exactly the position at which the largest fraction of the contamination is expected the largest fraction of the contamination.

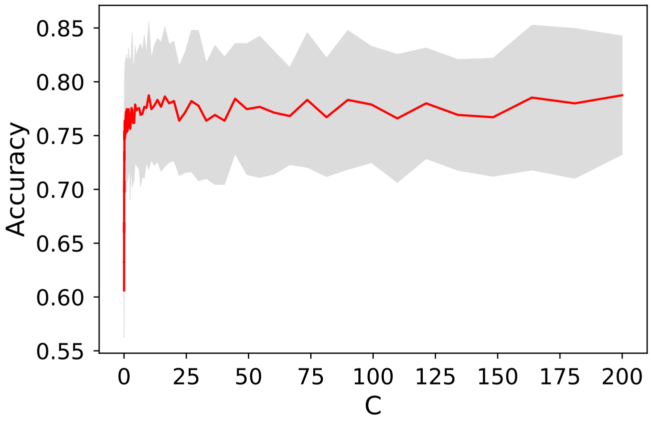


Fig. C.1. Achieved accuracy with respect to the regularization parameter C for SVC. The gray area corresponds to 1σ.

Appendix C: Validation curves

In this section we provide the plots for the hyperparameter optimization, for the three algorithms we used. ig. C.1 shows the validation curve for the `C` for SVC. Fig. C.2 displays the basic parameters for the RF: `n_estimators`, `max_lead_nodes`, and `max_depth`. In Fig. C.3 we present the results from the optimal structure for a NN, while in Fig. C.4 we plot the validation curves for `alpha`, `batch_size`, and `max_iter`.

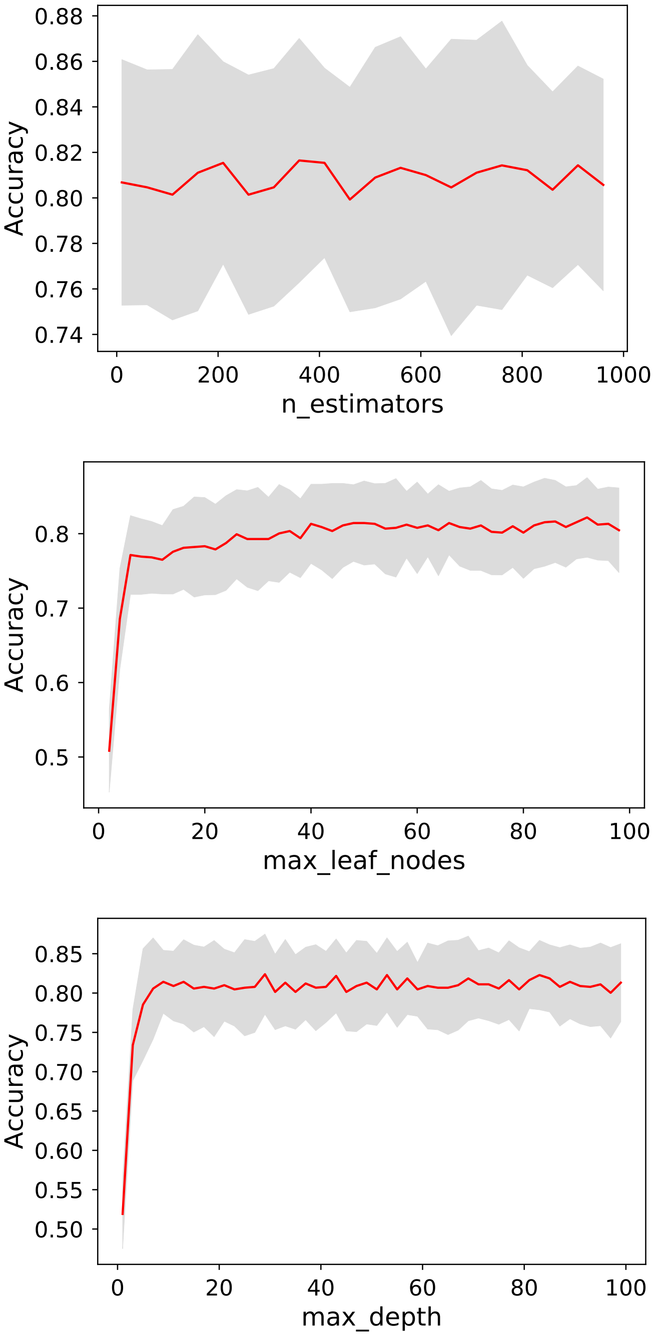


Fig. C.2. Validation curves of selected hyperparameters for the RF model: `n_estimators` is the number of trees in the forest (top), `max_leaf_nodes` is the number of nodes in each tree (middle), `max_depth` the maximum depth of the tree (bottom). The other hyperparameters are left to their default values. The gray area corresponds to 1σ.

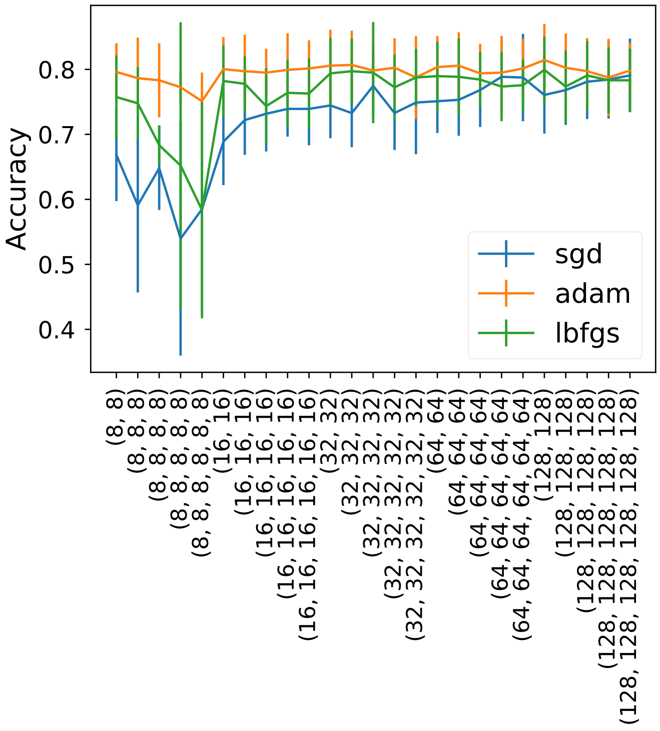


Fig. C.3. Obtained accuracy for the different MLP architectures (number of hidden layers with the number of nodes) per solver. `'adam'` seems to work systematically better among the solvers, with the best accuracy achieved for a network with two layers with 128 nodes each.

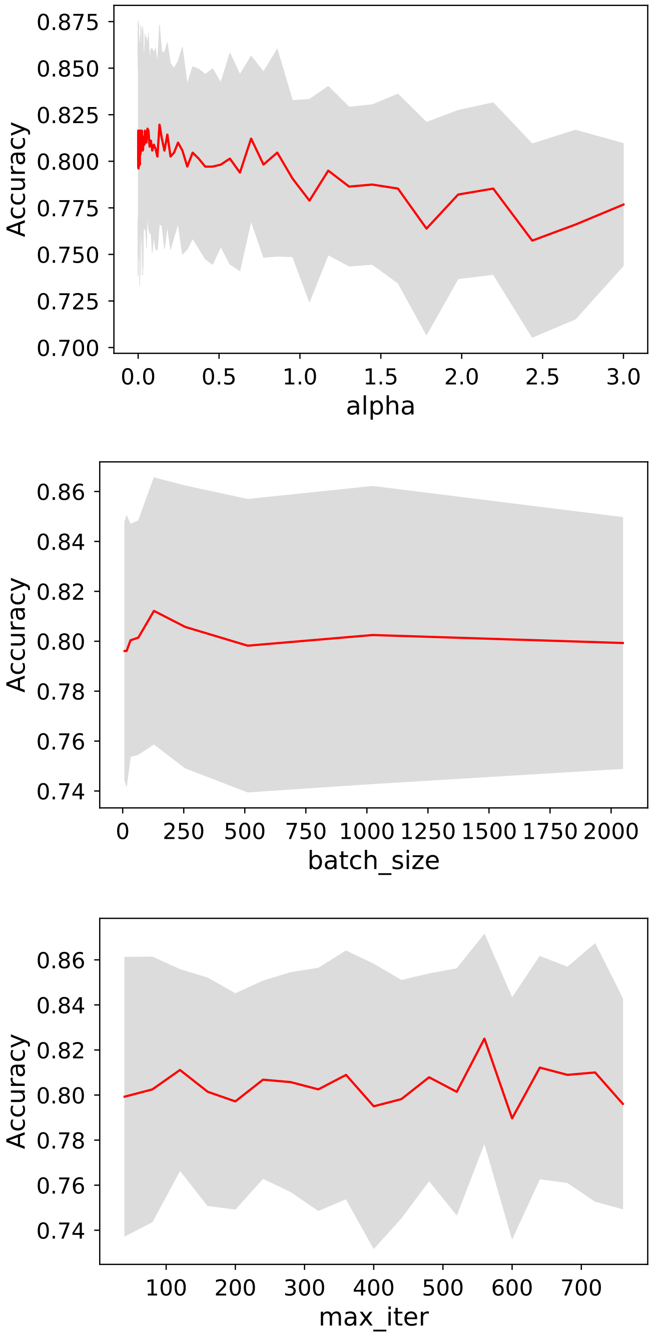


Fig. C.4. Validation curves for the (64,64) architecture of the MLP, examining `alpha` (L2 regularization parameter), `batch_size` (number of samples to estimate the gradient), during the weight optimization of the MLP), and `max_iter` (maximum number of epochs during training), with optimal values defived as ∼ 0.04, 128, and 160, respectively. The other hyperparameters are left to their default values.

Appendix D: Metrics with sample volume

Complementary to Fig. 10 we plot all metrics (recall, precision, f1-score) in Fig. D.1.

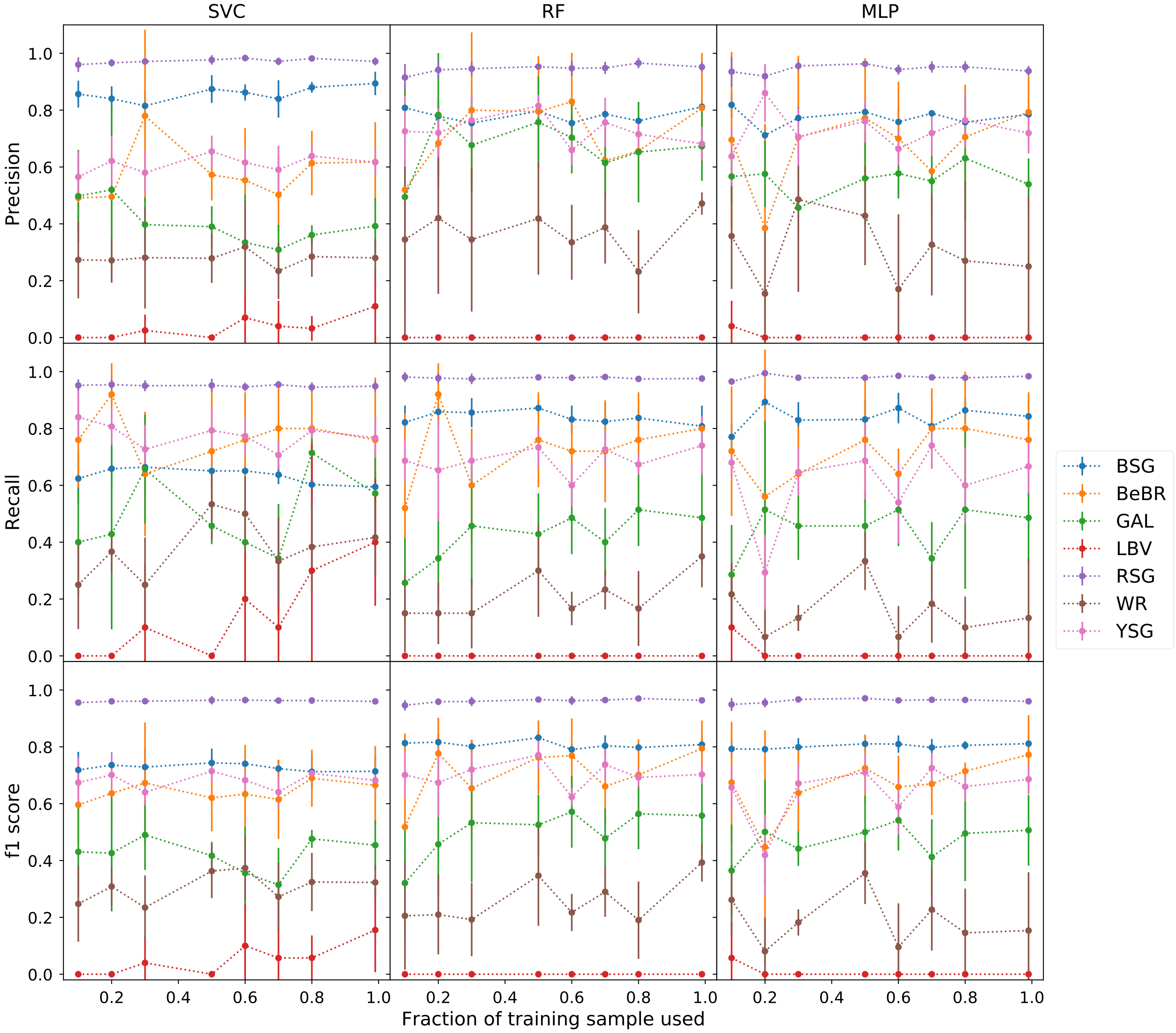


Fig. D.1. Precision, recall, and f1-score variation by adjusting the used training sample (per class). We notice that the metrics improve when we have a significant increase of the sample used, such as for BeBR and YSG. In cases where the samples sizes are already adequate (BSG and RSG) the maximum possible value is achieved faster.