Machine Learning Classifier for the Large Magellanic Cloud using Gaia EDR3 data

Author: Arnau Muros Sanchez
Facultat de Física, Universitat de Barcelona, Marti i Franquès 1, 08028 Barcelona, Spain.

Advisor: Xavier Luri Carrascoso and Óscar Jiménez Arranz

Abstract: In this paper, we develop a method to classify the stars from the Gaia Early Data Release 3 (Gaia EDR3) in order to distinguish the Large Magellanic Cloud stars from the Milky Way stars. To do so, several machine learning algorithms have been adapted, tested and applied.

I. INTRODUCTION

The European Space Agency (ESA) Gaia [1] astrometric mission is mapping the positions and motions of almost two billion stars in our galaxy and neighbouring galaxies, producing the most detailed 3-D map of our cosmos ever made. The data is being used to study the structure, evolution, and characteristics of the Milky Way (hereinafter MW), as well as other galaxies.

However, in order to do so, the application of analysis methods adapted to the large data volume is required. In this work, we start from the paper [2], in which the Magellanic Clouds – the Large Magellanic Cloud (LMC) and the Small Magellanic Cloud (SMC) – are studied. The first step for the analysis of these objects is to characterize which stars in this region of the sky are in fact from the Magellanic Clouds and which are foreground stars belonging to the Milky Way. This process in [2] is done following a movement-based approach in the orthographic plane (proper motion analysis). This criterion, however, is applied with the aim of removing only stars clearly belonging to MW. It is therefore more focused on completeness (do not lose LMC and SMC stars) than on purity (have a minimum presence of MW stars in the samples).

In this context, this work seeks to define an optimal method of star classification, which allows us to select a purer sample of LMC stars (we leave the study of the SMC for future work) without losing many in doing so. To this end, different classifiers based on Machine Learning are applied and tested, and the best performing algorithm is identified.

For the application of these classifiers, a training dataset representative of the problem is needed. In this case, we have used the data provided by the Gaia Object Generator simulator [3] (hereinafter GOG).

Once we have selected and trained a classification algorithm, we apply it to our Gaia dataset [4] and we verify the results by comparing them with previous independent classifications made with other authors (StarHorse [5], RR Lyrae [6], and Cepheids [7]).

II. DATASETS

LMC Base Sample

The base sample we will be working with is a 10^{o} circular selection of the EDR3 catalogue [4] centred at $(\alpha, \delta) = (81.28^o, -69.78^o)$ and with a limit G magnitude of 20.5 obtained by querying the Gaia EDR3 archive using the following ADQL query.

```
SELECT *
FROM gaiaedr3.gaia_source as g
WHERE 1=CONTAINS(POINT('ICRS',g.ra,g.dec)
CIRCLE('ICRS',81.28,-69.78,10))
AND g.parallax IS NOT NULL
AND g.phot_g_mean_mag < 20.5
```

The resulting sample contains 15.501.760 stars. It slightly differs from the one used in [2] because the larger selection radius of 20^{o} used in that paper includes a part of the SMC, and we are interested in avoiding it in order to specifically study the LMC.

Simulation

To carry out the training process and comparison of the different classifiers, we will use data provided by A.C. Robin et al. 2012: Gaia Universe Model Snapshot [8].

The Gaia Simulator was developed to simulate the huge amount of data provided by Gaia. This simulator is organized around one tool box called GaiaSimu, which contains a template universe, an instrumentation model, numerical methods, astronomical tools, etc.

We will use a specialized component called Gaia Object Generator (GOG) to generate realistic EDR3-like simulations.

This is very useful to us, as GOG is specifically designed to simulate catalogue data and provides objects with position, kinematics, photometry, and spectrum, as in the Gaia EDR3 (Early Data Release 3) archive.

StarHorse

As presented in [5], taking advantage of the improvements in accuracy and number of objects, this paper re-

fines the StarHorse code previously used by StarHorse Gaia DR2 [9].

This code uses the Gaia observables and seeks to obtain additional parameters of the EDR3 stars such as the distance d, the age τ, the effective temperature T_{eff}, the metallicity $[M/H]$, the AV extinctions (at $\lambda = 542nm$) and the surface gravity $log\, g$ from isochronal fitting. We will use this catalogue to verify our classification results, we will use the StarHorse distances to define an LMC vs MW separation based on a cut-off distance. To determine this threshold distance, we use the distance histogram to define a value that correctly separates two the MW from the LMC based on its spatial distribution (Fig.1). We can see an approximately Gaussian form centred on 50kpc corresponding to LMC and a decreasing exponential distribution identified as the MW. The chosen threshold distance is 25kpc.

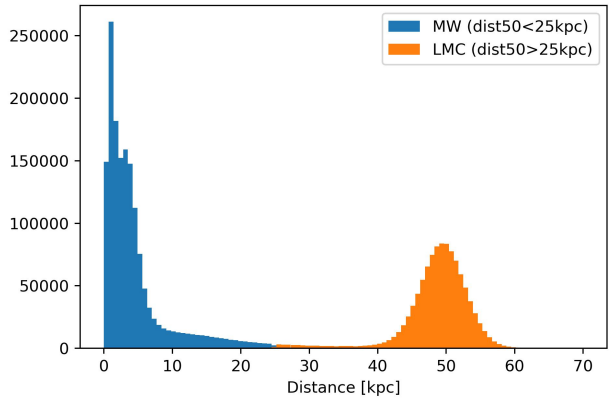

FIG. 1: Star distance distribution of StarHorse dataset in LMC sky section. In blue, stars treated as MW and in orange those treated as LMC according to the threshold distance.

RR Lyrae and Cepheids

Two additional catalogues will be used to verify the classification results, one containing RR Lyrae stars and the other containing Cepheids. Taking advantage of the physical properties of these variables (the existence of period-luminosity relations, described below), the authors can accurately and independently determine stellar distances, and thus build samples that only contain LMC objects. Therefore, we can use these samples as "pure LMC samples" for our verification.

Cepheids stars are variable stars that change in brightness periodically, a fact that we observe as changes in their luminosity. The period of change of a Cepheid star is related to its luminosity; with this relationship, the absolute magnitude of these stars can be obtained. By measuring their apparent magnitude, the distance at which it is located can be calculated from the relation between magnitudes. We can compute the distance by applying the following formula: Eq. 1.

$$d = 10^{\frac{m-M+5+A_o}{5}} \quad (pc) \tag{1}$$

Where M is the absolute magnitude, A_o is the interestellar absorption, m the apparent magnitude and d is the distance in parsecs.

RR Lyrae variable stars are also undergoing changes in their radius, and are stars of spectral type between A and F. They are abundant in the spherical region of galaxies and in globular clusters.

These stars are located on the horizontal branch of the HR diagram and have all a similar absolute magnitude, close to 0.50 (also defined by a period-luminosity relation). In the same way that with the Cepheids, known the absolute magnitude and by measuring the apparent magnitude the distance can also be calculated with the Eq.1.

Proceeding in the same way as with StarHorse data, these Cepheid and RR-Lyrae datasets contain only stars that belong to LMC. That allows us checking if the classifier has characterized them correctly.

III. CLASSIFIERS

Parameters Selection

In order to differentiate between the stars belonging to the MW and those belonging to the LMC, a set of parameters must be selected that can be used for the classification. Based on the available EDR3 data, we have chosen the orthographic projection both in position and velocity (x, y, μ_x, μ_y) – calculated from the celestial coordinates (α, δ) and proper motions using Eq.(2) and (3) —, parallax π, magnitude G and colour $(G_{BP}-G_{RP})$.

$$\begin{aligned} x &= \cos(\delta) \cdot \sin(\alpha - \alpha_c) \\ y &= \sin(\delta) \cdot \cos(\delta_c) - \cos(\delta) \cdot \sin(\delta_c) \cdot \cos(\alpha - \alpha_c) \end{aligned} \tag{2}$$

$$\begin{aligned} \mu_x &= \mu_\alpha \cdot \cos(\alpha - \alpha_c) - \mu_\delta \cdot \sin(\delta) \cdot \sin(\alpha - \alpha_c) \\ \mu_y &= \mu_\alpha \cdot \sin(\delta_c) \cdot \sin(\alpha - \alpha_c) \; + \\ &\quad + \mu_\delta \cdot (\cos(\delta) \cdot \cos(\delta_c) \qquad + \\ &\quad + \sin(\delta) \cdot \sin(\delta_c) \cdot \cos(\alpha - \alpha_c)) \end{aligned} \tag{3}$$

With $\alpha_c = 81.28^o$, $\delta_c = -69.78^o$ as the LMC's centre coordinates.

Random Forest

Random Forest (RF) is a supervised Machine Learning algorithm that is used in classification and regression problems. It consists of a set of decision trees; each tree

is trained with one of the subsets of the training data. The classifier is formed by combining the results of these trained decision trees, which are then weighted in a way that minimizes error, this technique is called *Bagging*. In our case, the criterion that determines the result is *Majority Voting*.

We use the *Scikit-learn* library [10] to build our model. The parameters of the algorithm selected are: *n_estimators = 100*, *max_depth = 2* and *random_state = 0*. The other parameters are set to the default.

In order to determine how good the classifiers work on the simulation, we will use a metric called Confusion Matrix. The results are compared one on one with the input (simulated) data; as in the simulation we know which stars are MW or LMC, the amount of correctly or incorrectly predicted results of each class can be measured. These results are presented in a matrix, the Confusion Matrix. The Confusion Matrix for the results obtained with *Random Forest* on the GOG simulation is shown in Fig.2.

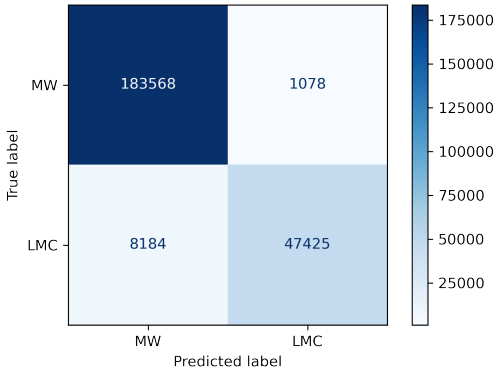

FIG. 2: Confusion Matrix for RF classification applied to GOG data.

K-Nearest Neighbors

The K-nearest Neighbors (KNN) is an instance-based machine learning algorithm. It follows a supervised learning model, in which the input is a sample that represents a percentage of the data to be studied and which it is used to later classify the data we pass on to it. The classifier works in three stages. [11]

First, it reads the training dataset and then, stores the D parameters and class of each element, which it places in a D-dimensional space. With this step, the trained classifier is created.

Next, a new dataset will be introduced with the objects that one wants to classify. The classifier will place each element in a D-dimensional space and identify its k nearest neighbors among the training points.

Finally, it uses the types of these k neighbors that it has identified to determine the type of each element, using a voting scheme based on majority rule. All neighbors could have uniform weight or could have weights inversely proportional to their distance to the element one wants to identify.

One of the features to keep in mind for this classifier is that it requires a calculation of the distance between all points in D-dimensional space. This calculation scales as $O[DN^2]$, so it is inefficient for large datasets.

The confusion matrix of the results obtained with KNN classifier on the GOG simulation is shown on Fig.3.

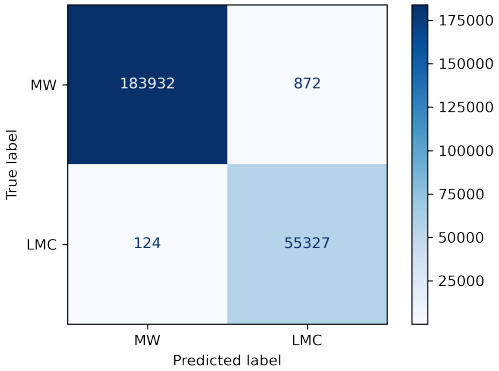

FIG. 3: Confusion Matrix for KNN classification applied to GOG data.

Neural Network

A Neural Network (NN) is a type of supervised learning algorithm that builds a nonlinear function from a set of training data that determines from the parameters what is the type of an element we want to characterize.

The structure of the neural network is made up of a number of neurons arranged in layers. The first layer (called *input layer*) contains D neurons that represent the input parameters and are connected to each one of the neurons in the next layer. The layers between the *input layer* and the *output layer* are called *hidden layers*, the value of each neuron in these layers is a linear combination of the values of the neurons in the previous layer.

With this configuration, however, only linear adjustments could be made. Therefore, the result of each neuron goes through an activation function, which allows it to make nonlinear adjustments. The activation function used is the default one: the function *'relu'*, $f(x) = max(0, x)$. This function is the usual recommendation for classification problems. The value of each neuron a_i in the first hidden layer will have the value:

$$a_i = f\left(\sum_j^D \omega_{ij} x_j + b_i\right) \tag{4}$$

This recurrence formula is used in every layer of the neural network. The coefficients ω_{ij} and b_i are those that are adjusted during the training phase of the model based on the selected data.[12]

The structure of the hidden layer used is (6,3,2), where the numbers represent the number of neurons in each layer. This configuration has been chosen after some different tries and keeping the one with better results and gives a confusion matrix on the GOG simulation that is shown on Fig. 4.

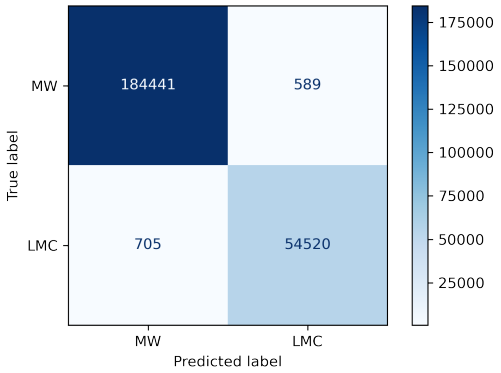

FIG. 4: Confusion Matrix for NN classification applied to GOG data.

IV. RESULTS ON GAIA'S DATASET

Using the algorithms trained with the simulation data, we classify the stars of the Gaia dataset in the selected region, obtaining for each star a probability of being MW and a (complementary) probability of being LMC; as we focus on LMC, we keep only the probabilities of being LMC. A cut-off probability must be chosen to complete the classification. The higher the cut-off probability, the higher the purity of LMC, but at the same time, the lower the completeness. After studying different values and as we are more interested in purity, a cut-off probability P = 0.5 has been chosen. It is also the optimal cut-off value based on the Receiver Operating Characteristic curve (ROC curve), a graph showing the performance of a classification model at all classification thresholds.

The results obtained with the NN classifier are shown in receiver operating characteristic Fig.5, the ones obtained with KNN are shown in Fig.6 and the results of the RF in the figure Fig.7.

V. COMPARISON OF RESULTS

In order to verify the results obtained with the classifier, we have compared them with the catalogues described in Section II, doing a crossmatch and evaluating

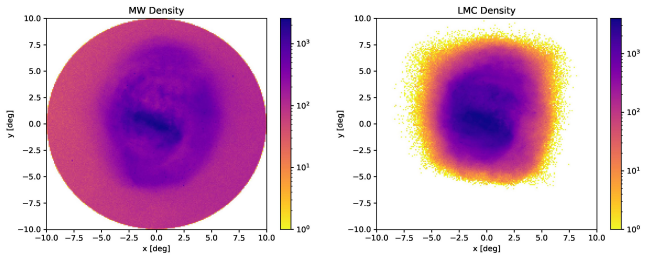

FIG. 5: Density map of LMC (right) and MW (left), classification made with NN. Obtained 8075976 LMC stars and 7425784 MW stars.

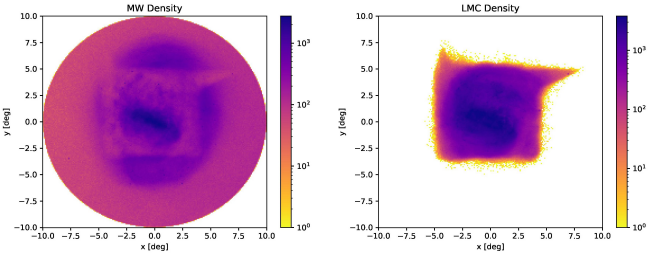

FIG. 6: Density map of LMC (right) and MW (left), classification made with KNN. Obtained 9131253 LMC stars and 6370507 MW stars

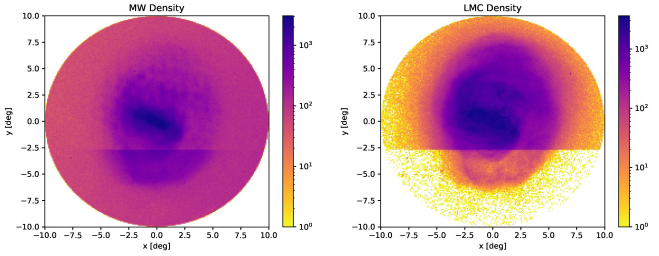

FIG. 7: Density map of LMC (right) and MW (left), classification made with RF. Obtained 8849906 LMC stars and 6651854 MW stars

the resulting completeness, purity, and accuracy. The first thing that is needed is the number of coincidences: for a given number of stars predicted as LMC, the amount of them that are characterized as LMC in the SH, RR Lyrae and Cepheids catalogues. These are considered as *True LMC*. The ones predicted as MW by the classifier but characterized as LMC in SH, RR Lyrae and Cepheids are the *False MW*.

As SH also give us a dataset characterized as MW, it can be used to calculate the coincidences in these results: *True MW* are the predicted as MW in the classifier and in SH, *False LMC* are the stars classified as LMC, but are MW in the SH dataset.

Doing the previous calculations, the results obtained for each classifier are:

RF	***T LMC***	***F LMC***	***T MW***	***F MW***
RR Lyrae	14987	-	-	6332
Cepheids	4081	-	-	184
StarHorse	535008	338727	601768	80073

TABLE I: Crossmatch of the RF results with catalogues to obtain coincidences as True LMC, False LMC, True MW and False MW.

KNN	***T LMC***	***F LMC***	***T MW***	***F MW***
RR Lyrae	15646	-	-	5673
Cepheids	4025	-	-	240
StarHorse	494440	329568	610927	120641

TABLE II: Crossmatch of the KNN results with catalogues to obtain coincidences as True LMC, False LMC, True MW and False MW.

NN	***T LMC***	***F LMC***	***T MW***	***F MW***
RR Lyrae	15373	-	-	5946
Cepheids	4097	-	-	168
StarHorse	443606	325035	615460	171475

TABLE III: Crossmatch of the NN results with catalogues to obtain coincidences as True LMC, False LMC, True MW and False MW.

VI. CONCLUSIONS

After analysing the results, the *Random Forest* classifier has been discarded, as it gives clearly worse results than the other two. The comparative of the other two classifiers is more even. Since the two gives good results, we have carried out a more detailed analysis.

First, our main goal is to have a classifier that identifies the stars that are LMC with the minimum error possible. The data we want to identify is from Gaia EDR3, therefore, that classifier has to work well on that dataset even if it has been trained on GOG simulation data. It is a key point to take into account, since KNN adapts itself so much to the training dataset that it filters parameters (such as coordinates) too strictly, giving results with undesired characteristics. This can be seen in the figure 6, where the rectangular form of the LMC training dataset clearly shows up in the results with the EDR3.

On the other hand, even though it has been trained by the same training dataset, the neural network seems to be better suited to extrapolation on the Gaia data, giving results free of these particularities. This shows that NN is a more adaptive option and less (although not negligibly) dependent on the shape of the training data.

Also, something to keep in mind is the ease of use. While NN has a relatively long training time and an almost instantaneous prediction time, the KNN classifier has a very fast training time but a significantly long prediction time (48 min). This means that while the NN can be trained once, stored, and then be applied as many times as wanted, KNN needs a large amount of time for every application. This also makes NN a more suitable option. Therefore, we finally discard the KNN classifier and retain the NN classifier as our choice for the MW/LMC separation.

The next step to follow to improve the results would be to train the classifier with some LMC training data with a cut-off shape of the coordinates equal to that of MW, since it has turned out to strongly influence the characterization of the stars.

Finally, as of the delivery date of this TFG (June 16, 2022), Gaia DR3 data was published 2 days ago, it would be of great interest to investigate the operation of the classifier on the new data.

The selected classifier: the *Neural Network* is being used by my advisors in the writing of a new paper for the kinematic analysis of the LMC using EDR3. The MW/LMC separation provided by this classifier is essential for the study being carried out. In that paper, other configurations, cut-off probabilities and more sophisticated interpretations of the results are being developed.

Acknowledgments

This final project that culminates my Physics degree would not have been possible without the guidance and support of my advisors Xavier Luri Carrasco and Óscar Jiménez Arranz, whose councils have allowed me to move forward and lead the project to the end. Special thanks to my family and friends, who supported me and cheered me up during all the process.

[1] ESA - Gaia [En línia]: `https://www.esa.int/Science_Exploration/Space_Science/Gaia`
[2] Gaia Collaboration, X. Luri et al. (2021)
[3] Luri, X., Palmer, M., Arenou, F., et al. 2014, AA, 566
[4] Gaia Collaboration et al. (2020a): Gaia EDR3: Summary of the contents and survey properties.
[5] Anders, Khalatyan, et al. 2022: StarHorse parameters for Gaia EDR3 stars
[6] F.Cusano et al. 2021
[7] V. Ripepi et al. 2022
[8] A.C. Robin et al. 2012: Gaia Universe Model Snapshot
[9] Anders, F., Khalatyan, A., Chiappini, C., et al. 2019
[10] Scikit Learn 1.1.1. Random Forest Classifier. Web-based documentation
[11] Scikit Learn 1.1.1. K Neighbors Classifier. web-based documentation
[12] Scikit Learn 1.1.1. Neural Networks. Web-based documentation