Exploring the star's classification via neural networks

Michele Santoni

Dipartimento di Fisica e Astronomia, Università di Bologna, via Irnerio 46, 40126 Bologna, Italy

Stellar ages play a crucial role in understanding the formation and evolution of stars and Galaxies, which pose many challenges while determining in practice. In this paper, we will expose the analysis of a set of data of the GAIA telescope, taking advantage of two complementary approaches. The first part is dedicated to the study of the graphs generated by the relation between the different parameters: we will analyze relationships among stellar parameters such as distance, luminosity, age and mass, investigating connectivity, distribution and communities. In second part, we employ neural networks to develop predictive models, showcasing the processes of data preparation, model training and optimization. This dual approach highlights the potential of combining graph analysis and machine learning for advancing our understanding of stellar evolution and clustering.

Key words: Graph Analysis Distance, Luminosity, Mass, Age, Connectivity, Core-Periphery Structure, Community Detection, Louvain Algorithm, Degree Distribution, Degree Centrality, Threshold Effects Galactic Structures, Stellar Populations, Heterogeneous Data, Graph Density, Modularity Score Peripheral Stars, Stellar Evolution, Neural Networks, Predictive Models, Feature Selection, Data Preprocessing, Machine Learning, Model Training, Optimization

1 Introduction

The Gaia telescope is creating an extraordinarily precise three-dimensional map of more than a thousand million stars throughout our Milky Way galaxy and beyond, mapping their motions, luminosity, temperature and composition. This huge stellar census provides the data needed to tackle an enormous range of important questions related to the origin, structure and evolutionary history of our galaxy. Taking advantage of this, it is possible to perform an intense and deep analysis via the Neural Networks technique, exploring both the graph representation and the implementation of predicting models.

In this paper: Sec.(2) is dedicated to an overview of the full dataset since it crucial knowing the data under consideration before getting into the subject, with all the outcomes sum up in Sec.(2.4); without this analysis, one will proceed via several blind attempts, increasing the amount of work and probably failing in the study. Sec.(3) is focused on the graph study for 4 features: distance, luminosity, age and mass, which are combined in couples and all together; the results are reported in Sec.(3.4). Sec.(4) instead presents the neural network analysis, passing through simple and multilayer methods and using all the main strategies for improving the network. Lastly, Sec.(5) briefly summarize then content and the results, opening the road for some extra possible work.

2 Explanatory Analysis of Dataset

2.1 Dataset description

The dataset under consideration comes from the DR3[1] of the "Gaia Telescope" and it is composed of over 50 columns, with 626 016 rows of data. However, not all the rows are full filled and some missing values are present. Specifically speaking, the columns contain all types of information about stars: position, magnitude, star's characteristics, errors, spectral classification etc... which permits to have a complete overview of the stars in our galaxy and beyond. If one is interested in more, the complete description of the dataset can be found Gaia Dataset.

[1]Data Release 3: 13 June 2022

Starting from this, the dataset has been modified: most of the features (columns) are unnecessary for the work, hence they have been reduced to a lower numbers: 12 total columns. Precisely speaking:

1. RA_{ICRS}: Right ascension in the ICRS[2] coordinate system;
2. DE_{ICRS}: Declination in the ICRS coordinate system;
3. $log(g)$: Surface gravity ;
4. $[Fe/H]$: Abundance of neutral iron, called metallicity ;
5. $Dist$: Distance to the celestial object: inverse of the parallax, in parsece;
6. T_{eff}: Estimated effective temperature of the celestial object by Gaia in Kelvins;
7. Rad[3]: Object radius estimate in terms of solar radius;
8. $Rad - Flame$[4]:Object radius estimate in terms of solar radius;
9. Lum: Estimated object luminosity in terms of solar luminosity (via Flame);
10. $Mass$: Mass estimate in terms of solar mass(via Flame);
11. Age: Celestial object age in giga years (via Flame);
12. z: Redshift in km/s (via Flame);

which are the variables most related to the age of a star, except for the ones connected to the position. Notice also the presence of 2 different estimation of the radius.

[2]International Celestial Reference System

[3]Radius from GSP-Phot Aeneas best library using BP/RP spectra

[4]FLAME is a machine-learning algorithm designed to fit Voigt profiles, which is a probability distribution given by a convolution of a Cauchy-Lorentz distribution and a Gaussian distribution, to H_1 Lyman-alpha ($Ly\alpha$) absorption lines using deep convolutional neural networks

2.2 Data transformations

In the study of the graph, all over the total amount of data, only the initial 32000 rows of data have been selected, ignoring each line which has some null values; as a consequence of this, a total number of 2508 rows have been obtained.

Index	2508 entries, 26017 to 31998	
Columns	**Non-Null Count**	**Dtype**
RA_{ICRS}	2508	float64
DE_{ICRS}	2508	float64
$log(g)$	2508	float64
$[Fe/H]$	2508	float64
$Dist$	2508	float64
T_{eff}	2508	float64
$Radius$	2508	float64
$RadiusFlame$	2508	float64
$Luminosity$	2508	float64
$Mass$	2508	float64
Age	2508	float64
z	2508	float64

Table 1: Data types

There are several reasons for the choice of using a smaller sample: the amount of data is really huge and, having previously tested out several larger samples, no significant improvements, in the quality of the graphs, have been noticed when a larger number was used. Besides, for a number of non-null rows greater the ten thousands, the computing time was too high to permit a manageable study. As a consequence, it was opted to get a smaller, but meaningful, subset.

2.3 Data exploration

Before doing any kind of graphs construction, machine learning and data analysis it is important to understand the data with one is dealing. For that purpose, it is essential to first focus on the characteristics of the dataset and then to the implementation of any the neural network types of work.

2.3.1 Generic statistics

First of all, it is useful to have a generic overview on the statistics of the data:

	RA_ICRS	DE_ICRS	log(g)	[Fe/H]
mean	76.27	35.47	3.83	-0.73
std	8.17	4.63	0.19	0.45
min	26.19	1.24	2.91	-2.28
25%	76.13	34.02	3.69	-0.99
50%	78.61	36.23	3.83	-0.72
75%	80.43	38.39	3.99	-0.43
max	84.33	41.63	4.34	0.80

Table 2: Summary of the statistics for RA_ICRS, DE_ICRS, log(g), and [Fe/H]. It is clear that the statistics of "RA_ICRS" and "DE_ICRS" are not meaningful at all, since they lack of additional (on their own) spatial context.

The dataset predominantly consists of young, hot and massive stars with an average effective temperature of $\sim 10\,000K$ and a mean mass of ~ 2.73 solar masses. The population shows high luminosities, averaging ~ 89.67 solar units, with extreme

	Dist [Parsec]	Teff [K]
mean	3530.46	10207.66
std	1589.86	500.27
min	53.48	9600.12
25%	2529.64	9815.59
50%	3624.40	10025.86
75%	4497.41	10505.16
max	11245.61	11708.81

Table 3: Summary of the statistics for distance and effective temperature

	Rad [Solar Radius]	Rad-Flame [Solar Radius]
mean	3.18	2.86
std	1.01	0.81
min	1.41	1.72
25%	2.36	2.24
50%	3.02	2.67
75%	3.76	3.29
max	10.56	13.08

Table 4: Summary of the statistics for radius and radius flame

	Lum. $[L_{Sun}]$	Mass $[M_{Sun}]$	Age [Gy]	z [Km/s]
mean	89.67	2.73	0.30	0.48
std	68.33	0.33	0.06	0.09
min	22.95	2.07	0.20	0.20
25%	45.50	2.44	0.25	0.43
50%	72.93	2.68	0.29	0.47
75%	114.78	2.97	0.34	0.52
max	2054.97	3.73	0.52	0.89

Table 5: Summary of the statistics for luminosity, mass, age and redshift

cases reaching over 2 000 $solar\ units$, indicating the presence of evolved giants alongside main-sequence stars. Most stars are relatively young, with an average age of $0.3Gy$, and display low metallicities ([Fe/H] mean: ~ 0.73), suggesting they belong to early stellar populations or recently formed regions. The distances span a wide range, from nearby stars at $\sim 53\,parsecs$ to distant ones over $\sim 11\,000\,parsecs$, highlighting a diverse sample across evolutionary stages and environments.

2.3.2 Space configuration

The Fig.(1)(2) provide a visual representation of the spatial distribution of stars based on their right ascension (RA), declination (Dec) and distance, in parsecs. These plots help in understanding the clustering of stars, detecting outliers and assessing whether the dataset exhibits any structured patterns. Precisely:

- Fig.(1): the 3D scatter plot visualizes the full spatial distribution of stars using RA, Dec and distance. The color gradient represents distance, with closer stars appearing in darker shades and distant ones in brighter tones. The distribution suggests a dense concentration of stars at lower distances, while farther objects appear more scattered, indicating possible observational biases or physical clustering of stars in space;
- Fig.(2): the left panel displays the "RA vs Dec" scatter plot, where the color scale represents the distance (par-

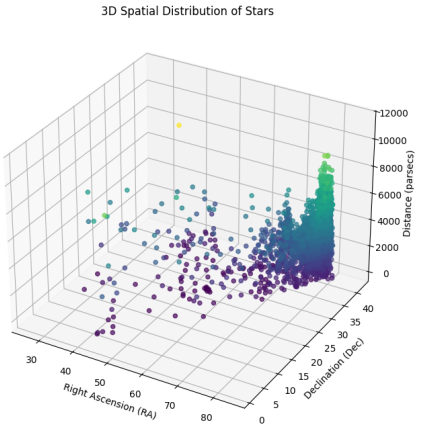

Figure 1: 3D Spatial Distribution of Stars: The plot represents stellar positions in Right Ascension (RA), Declination (Dec), and Distance (parsecs). Color intensity indicates distance, highlighting clusters and distribution patterns.

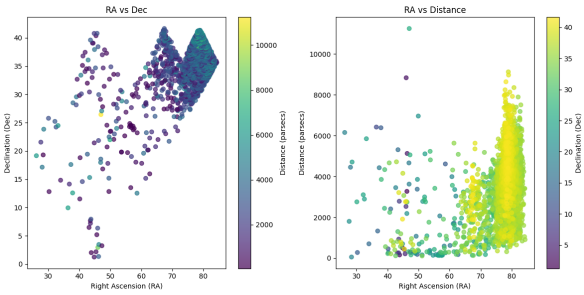

Figure 2: 2D Projections of Stellar Distribution: (Left) Right Ascension vs Declination, colored by Distance. (Right) Right Ascension vs Distance, colored by Declination.

secs) of each star. This helps in identifying whether certain regions of the sky have a higher density of nearby or distant stars. The right panel plots "RA vs Distance", with colors representing the Declination (Dec);

The clustering of points at specific RA values suggests that the dataset might be biased toward certain regions of the sky, possibly due to selection effects in data acquisition.

These visualizations are crucial in determining whether further data preprocessing is needed, such as normalization or subsampling, to ensure representative analysis for graph-based clustering or neural network predictions.

2.3.3 Histograms

In a second place, a very important aspect to observe are the distributions of the data, in order to "capture" their nature. In this direction, looking at the histograms distribution is a powerful tool to notice the presence of regularities, irregularities and outliers. By means of a simple python code, it is possible to generate the histograms in Fig.(3).

The data associated with the individual Gaussian are:

It is possible to notice how all the columns have a different order of magnitude and of variation; these aspects will be taken into account during the elaboration of the dataset, with a crucial role in the construction of the model.

The plots reveal some interesting facts: most of the features are peaked around a certain value, even if the distributions can not be truly called "Gaussian"; the two histograms for the

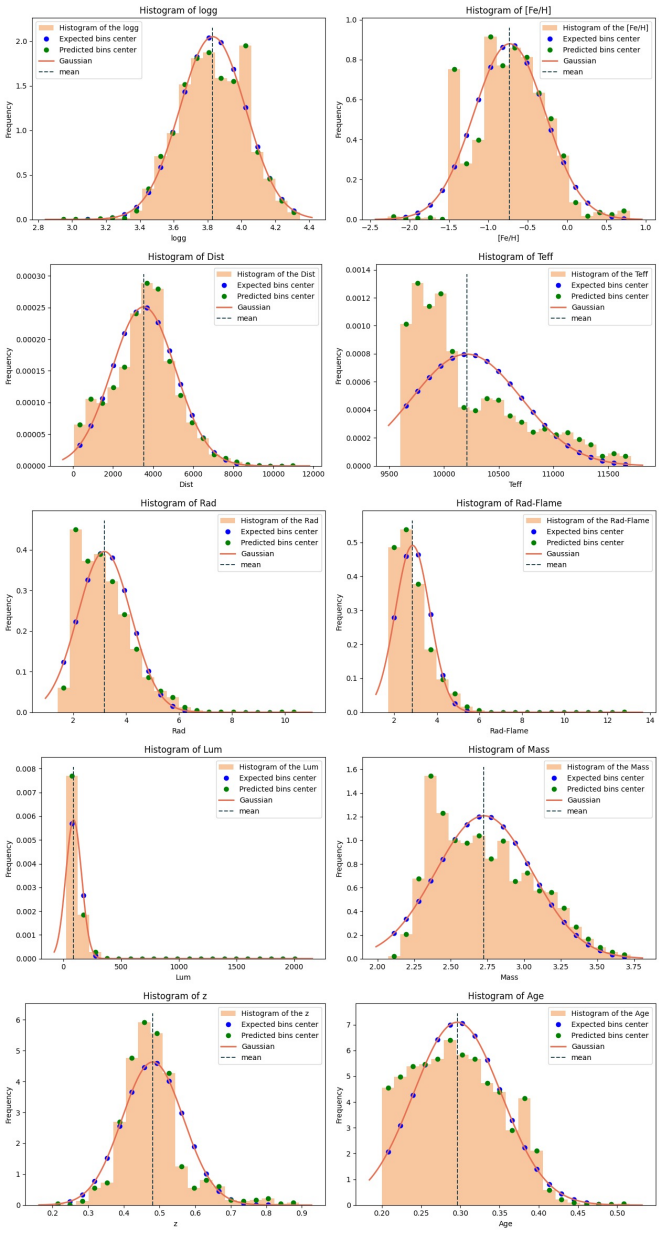

Figure 3: The image presents the histograms of the features of the data in the dataset, with a Gaussian distribution overlapping them; the colors refers to: orange to the histogram, red to the Gaussian, green to predicted bind center, blue to expected bins center and black to the vertical line of the mean value. The histograms are in order (from the top one on the left): log(g), [Fe/H], distance, effective temperature, radius, radius flame, luminosity, mass, redshift, age. For a full description of these features Sec.2.1.

radius are pretty similar in shape, but the "radius flame" distribution is more narrow; the effective temperature histogram is the only not peaked. The temperature is inherently not expected to follow a Gaussian distribution because of the logarithmic nature of stellar classifications (e.g. Hertzsprung-Russell diagram[5]). The temperature distribution is more log-normal or skewed, reflecting the exponential drop in the number of massive, hot stars compared to cooler stars.

[5]The Hertzsprung–Russell diagram (abbreviated as HR diagram) is a scatter plot of stars showing the relationship between the stars' absolute magnitudes or luminosities and their stellar classifications or effective temperatures.

	Mean value	Standard deviation
$log(g)$	3.82982	0.194539
$[Fe/H]$	-0.727394	0.453826
$Distance$	3530.46	1589.54
T_{eff}	10207.7	500.172
$Rad_{expected}$	3.17911	1.00847
Rad_{star}	2.86439	0.812786
$Luminosity$	89.6664	68.3152
$Mass$	2.72563	0.330604
Age	0.296203	0.0562484
$Redshift$	0.481135	0.0861393

Table 6: Standard deviation and mean value of the features in the dataset.

2.3.4 Further on Histograms

Generically speaking, it is always not sufficient to study the data with respect to a normal distribution, since it is not known a priori whether the underlying data truly follows a Gaussian distribution or exhibits significant deviations due to multi-modality, skewness or heavy tails. To further inspect these aspects, high-order moments of the distributions, such as the "kurtosis"[6] and the "skewness"[7], offer a deeper perspective, helping to understand the distribution shape and detect deviations from normality.

	Kurtosis	Skewness
log(g)	-0.147297	-0.16534
[Fe/H]	0.214384	0.161179
Dist	0.0863102	0.0741425
Teff	0.0286673	0.972001
Rad	2.29297	1.0864
Rad-Flame	10.4243	1.7902
Lum	272.316	10.2404
Mass	-0.604993	0.498078
Age	-0.678504	0.255766
z	3.25888	1.20538

Table 7: The table shows the kurtosis and skewness of the histogram distributions of the data under consideration.

From the descriptive statistics provided in Tab.(7), it is possible to observe:

- **log(g)**: the numerical values of kurtosis (-0.15) and skewness (-0.16) indicates a fairly symmetrical and normally distributed dataset;
- **[Fe/H]**: the slightly positive kurtosis (0.21) and skewness (0.16) suggests a mostly normal distribution, but with a slight bias toward higher metallicity values;
- **Distance**: it presents a low kurtosis (0.086) and skewness (0.074), such that one can affirm that the distribution is close to a normal one, meaning distance values are evenly spread;
- **Teff**: The mildly positive skewness (0.97) indicates that a few very hot stars are pushing the mean upward. The kurtosis (0.03) suggests the temperature distribution is not significantly peaked;
- **Rad**: it shows a moderate positive skewness (1.09) and kurtosis (2.29). A possible explanation is that some large values (possibly giant stars) are affecting the mean;
- **Rad-Flame**: the high kurtosis (10.42) and skewness (1.79) are strong evidences that a few very large stars are causing an asymmetry. The dataset may contain a mix of main-sequence stars and evolved giants;
- **Luminosity**: The extremely high kurtosis (272.32) and skewness (10.24) suggests a highly skewed distribution with extreme outliers. Again, one can suppose that few very luminous stars (massive giants) are pushing the mean up significantly;
- **Mass**: the values indicate a mild negative kurtosis (-0.60) and positive skewness (0.50), hence a close to normal distribution, but slightly skewed toward higher-mass stars;
- **Age**: the negative kurtosis (-0.68) and low skewness (0.26) points directly to a fairly normal and symmetrical distribution;
- **z**: the moderate kurtosis (3.26) and skewness (1.20) suggesta a slightly peaked and right-skewed distribution. This means that more stars have low redshift values with a few high-redshift outliers;

Overall, this analysis highlights the importance of considering deviations from normality when studying astrophysical data, as it allows for a better understanding of the underlying distribution patterns and the presence of extreme values.

To go further in the study, one can consider to log-transform to normalize highly skewed distributions and not only run a dip test to check for an effectively "bi-" or "multi-" modality, but also confirm it visually with KDE[8] plots.

2.3.5 Bi-modularity

Hartigan's Dip Test is used to detect deviations from unimodality in a dataset. in short, a low p-value (< 0.05) suggests that the distribution is likely not unimodal, and hence either bimodal or multimodal.

The results of Tab.(8) are pretty clear: most columns (Distance, Teff, Luminosity, Mass, Age, Redshift) have high p-values ($>> 0.05$), directly proving that they are unimodal. However, some features has shown interesting outcomes: [Fe/H] shows strong evidence of bimodality in the metallicity distribution, probably related to the presence of two distinct stellar populations: older ones, metal-poor stars (formed early in galactic evolution), and younger ones, metal-rich stars (formed from enriched gas clouds); log(g) shows a moderate evidence of bimodality, that could be due to different stellar evolutionary stages (e.g., main-sequence stars vs. giants); lastly, the radius presents a moderate evidence of bimodality, likely to be a consequence of the separation between main-sequence stars and evolved stars (e.g., giants).

[6] Kurtosis measures the presence of outliers and the sharpness of the peak of a distribution compared to a normal distribution.

[7] Skewness quantifies the asymmetry of the distribution, indicating whether data is concentrated more on one side of the mean.

[8] In statistics, kernel density estimation (KDE) is the application of kernel smoothing for probability density estimation, i.e., a non-parametric method to estimate the probability density function of a random variable based on kernels as weights. *from Wikipedia*

	Dip Statistic	p-value
log(g)	0.0114	3.0983e-02
[Fe/H]	0.0265	0.0000e+00
Dist	0.0051	9.7759e-01
Teff	0.0078	4.2497e-01
Rad	0.0114	3.0168e-02
Rad-Flame	0.0047	9.9125e-01
Lum	0.0043	9.9375e-01
Mass	0.0047	9.9094e-01
Age	0.0075	4.8686e-01
z	0.0053	9.6304e-01

Table 8: Hartigan's Dip Test Results: Dip Statistic and p-values for various features. Low p-values (e.g., [Fe/H], log(g), Rad) indicate possible bimodality.

2.3.6 KDE

Kernel Density Estimation (KDE) is a non-parametric way to estimate the probability density function (PDF) of a dataset. It smooths data points by applying a kernel function (typically Gaussian) to approximate the underlying distribution. KDE is particularly useful for identifying modal structures (unimodal, bimodal or multimodal distributions), for detecting outliers and skewness in data and for comparing the empirical distribution with a theoretical one (e.g., Gaussian distribution).

From the behavior of the plots in Fig.(4), one concludes:

- **log(g)**: the gaussian fit aligns closely with the KDE, confirming near-normality;
- **[Fe/H]**: KDE reveals a subtle secondary mode, confirming Hartigan's Dip Test result. Probably, this indicates the presence of two stellar populations: one metal-poor and one metal-rich;
- **Distance**: right-skewed but approximately normal-like, even if there are some minor deviations from Gaussian, but overall unimodal;
- **Teff**: right-skewed distribution, with a long tail of hotter stars. KDE deviates significantly from Gaussian, indicating non-normality and suggesting a mix of main-sequence and evolved stars;
- **Rad** and **Rad-Flame**: right-skewed, with a long tail of larger stars. KDE highlights the presence of many small-radius stars and a few very large stars;
- **Luminosity**: highly right-skewed, with an extreme peak at low luminosities and a long tail. Confirms what we saw in the kurtosis value (272.32), indicating a few highly luminous stars;
- **Mass**: KDE suggests a slightly heavier tail toward high-mass stars;
- **Age**: symmetric distribution with KDE closely follows Gaussian;
- **Z**: near-normal with a slight skew;

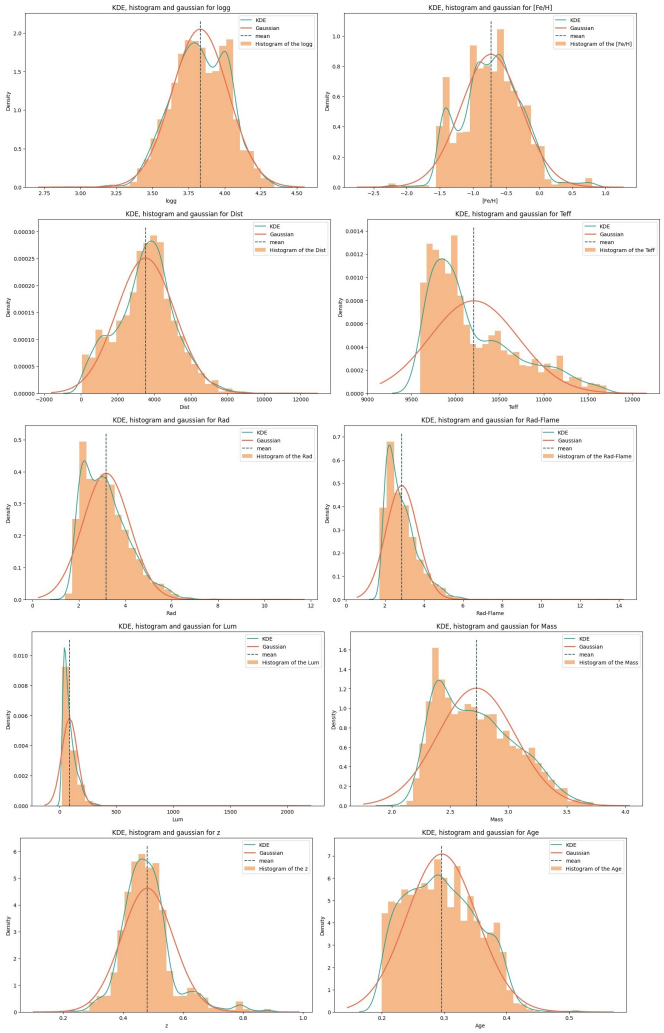

Figure 4: This image contains multiple KDE plots overlaid on histograms for different the studied parameters: log(g), [Fe/H], Distance, Teff, Radius, Luminosity, Mass, Redshift (z), and Age. Each plot has in orange the histogram, in light blue the KDE curve, in red the Gaussian fit and in black the a dashed line in correspondence with the center of the gaussian.

2.3.7 Box Plots

Box plots provide insights into the distribution, central tendency, variability and presence of outliers in the dataset. The blue box represents the interquartile range (IQR) (25th to 75th percentile), the horizontal line inside indicates the median, and the whiskers extend to 1.5 × IQR from the quartiles. Any points beyond the whiskers (red dots) are considered outliers.

Considering Fig.(5), one can extract some key features based on the box plot characteristics:

- **log(g)**: the distribution is fairly symmetrical, but some outliers exist on the lower side. This suggests a consistent range for most stars, with a few stars having significantly lower gravity (possibly evolved stars);
- **[Fe/H]**: several outliers on both sides, indicating the presence of metal-poor and metal-rich stars. The core of the distribution is balanced, but bimodal behavior (already seen in KDE plots) could be confirmed;
- **Distance**: Strong right-skewness with many outliers at higher distances. Most stars are concentrated at lower dis-

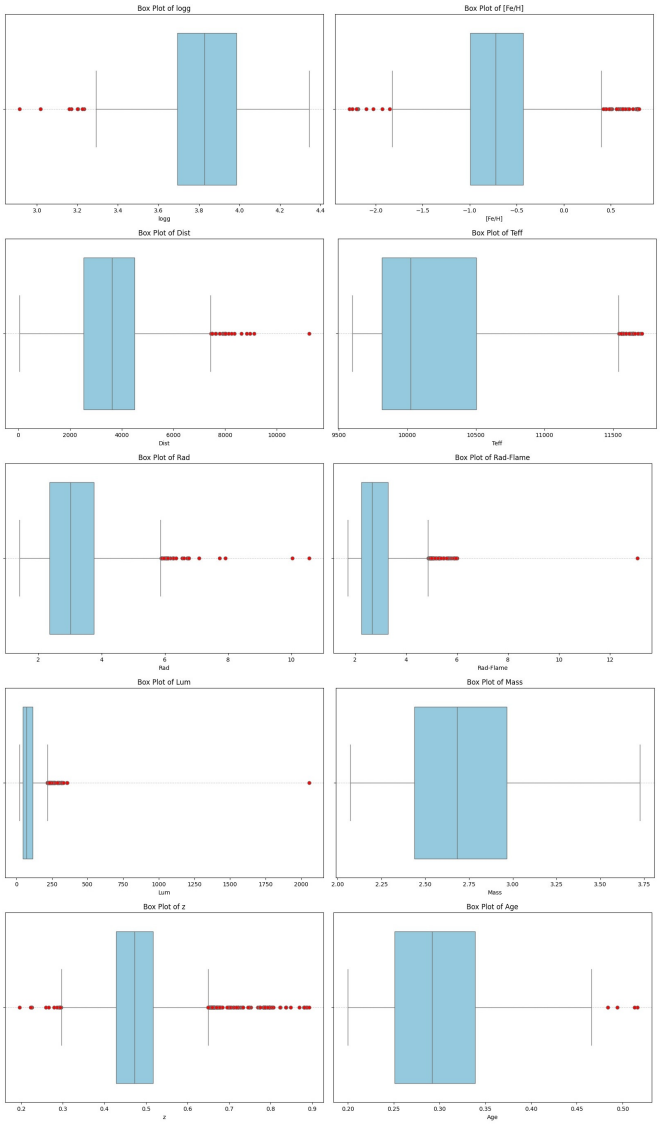

Figure 5: Box plots of the dataset variables.

tances, while a few stars extend to ∼10,000 parsecs, making them possible outliers or distant extragalactic sources;

- **Teff**: mild skewness to higher temperatures with some hot outliers. The distribution suggests that most stars have similar temperatures, with a few extremely hot ones (possibly early-type or giant stars);
- **Rad, Rad-Flame**: right-skewed distribution with significant outliers. Many small-radius stars exist, but several very large stars are pushing the tail. The outliers likely represent giant or supergiant stars;
- **Luminosity**: highly right-skewed, with many extreme outliers at high luminosities. The core data consists of low-luminosity main-sequence stars, while the outliers represent highly luminous stars (giants or supergiants). A log-transformation may help normalize this distribution;
- **Mass**:mostly normal-like, but with a single significant high outlier. The outlier may be an extremely massive star or a misclassified binary system;
- **Age**: mildly right-skewed, with several high-redshift outliers.This suggests most stars are relatively nearby, but some may belong to distant extragalactic populations;
- **z**: almost symmetrical distribution, but a few outliers at higher ages. This shows that most stars have similar ages, but some very old stars could be low-mass long-lived stars or remnants of ancient populations;

2.3.8 Correlation matrix

Another relevant statistical technique is the correlation matrix[9], which investigates the linear correlation between the variables, Fig.(6).

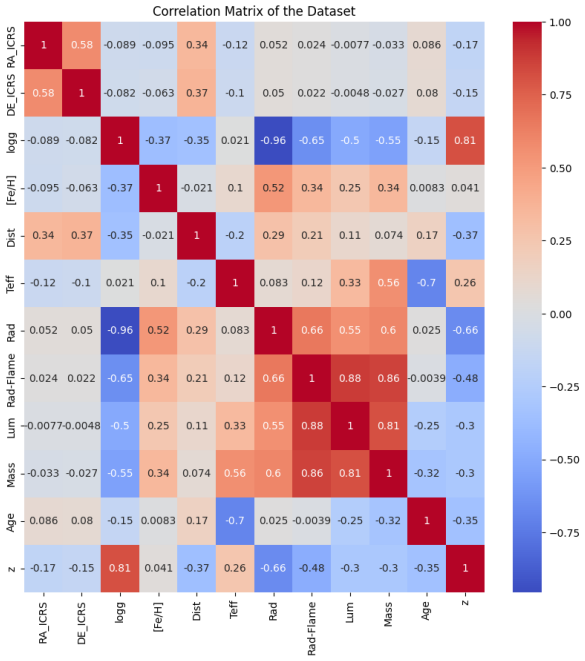

Figure 6: The image shows the correlation matrix of the 12 columns of the dataset. Higher positive value (red) means that there is a linear correlation; high negative value (blue) means there is anti-linear correlation; zero or near zero values (neutral color) correspond to the absence of correlation.

From the image is possible to deduce the presence of strong positive correlations (like luminosity and mass), strong negative correlations (like log(g) and radius) and weak correlations (like age and radius).

These insights help understand how variables are related and can be useful in identifying which variables might influence each other. However, it is not enough to complete see their mutual relation as the correlation matrix makes explicit only linear relations.

2.3.9 Density plot

The model is trained only on the features which are meaningful to estimate the age, which is the final goal: mass, luminosity, effective temperature, radius flame, log(g) and [Fe/H]. Thus, knowing how the variables are distributed with respect the desired one is a precious hint to predict the "weights" of each column, meaning the quantities which are more incisive, in the implementation of the model.

[9]the correlation matrix is a table that displays the correlation coefficients between variables in a dataset. Each cell in the matrix shows the correlation between two variables, ranging from -1 to 1: +1 is a perfect positive correlation (as one variable increases, the other variable increases), -1 is A perfect negative correlation (as one variable increases, the other variable decreases), 0 means no correlation. One must to be very careful since the correlation matrix displays only linear correlation

Looking in this direction, it is instructive to observe the distributions between one of the relevant quantity and the stellar age, Fig.(7).

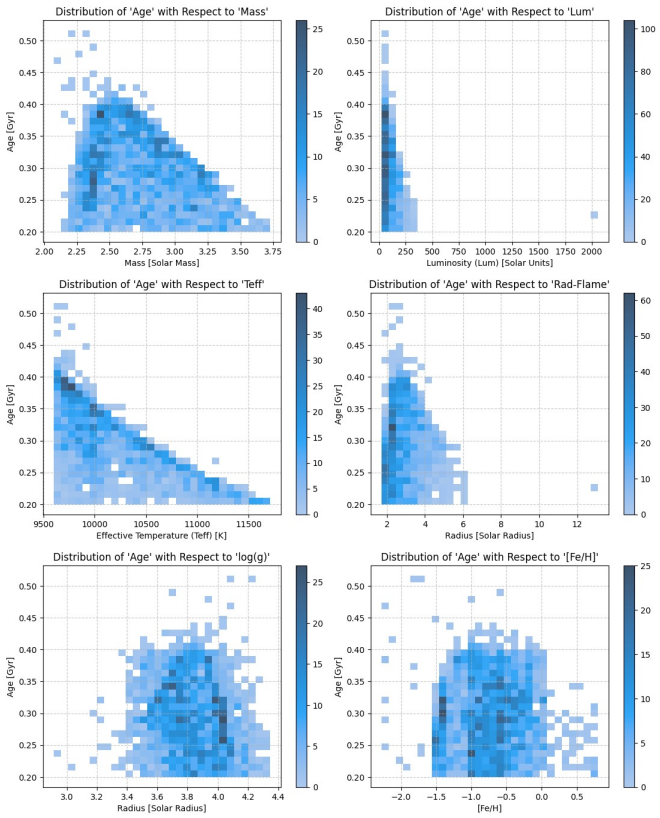

Figure 7: The image presents the density plot of the different features with respect to the age.

Generically speaking, these plots provide insight into the life cycles of stars, showing how physical properties like mass, luminosity, and temperature change with age.

Specifically speaking, from the series of plots one can learn:

- **Age vs. Mass:** The distribution shows that more massive stars are generally younger. Stars with higher mass (around 3.5 solar masses) have a shorter lifespan, which is why they tend to be younger. As mass decreases, the stars can be older, which follows known stellar evolution principles;
- **Age vs. Luminosity**: There is a trend where younger stars are more luminous and, as stars age, their luminosity tends to decrease. This distribution reflects the idea that young, massive stars are very bright but have shorter lifespans;
- **Age vs. Effective Temperature** : Higher temperatures correspond to younger ages. Stars cool down and become less hot as they age, which aligns with stellar cooling processes;
- **Age vs. Radius Flame**: Larger radii are associated with younger stars, and as stars age, their radius generally shrinks or stabilizes, depending on their stage of evolution;
- **Age vs. log(g):** The distribution between age and surface gravity (log(g)) indicates that older stars have higher surface gravity, which is consistent with the idea that stars contract over time, increasing their surface gravity;
- **Age vs. [Fe/H]:** The spread across metallicity indicates diversity in stellar populations. Older stars generally have lower metallicity because they formed when the universe had fewer heavy elements, while younger stars have higher metallicity due to the enrichment of the interstellar medium over time;

Note that the relationships depicted are consistent with established astrophysical principles: massive, hot stars being younger and more luminous. Besides, the diversity in metallicity highlights different generations of star formation, with older stars having formed from a less metal-rich environment; this is coherent with the fact that not all the stars collected by Gaia come from the same space region (in our galaxy or beyond).

2.4 Sum up on data exploration

Taking into account all the analysis, one can conclude that the dataset is composed mostly by young, hot and massive stars, with the presence of some evolved Giants and Supergiants alongside the main-sequence. The former are probably born from an enriched gas cloud because of their high metallicity; the latter, on the contrary, are born at the early stages of the galaxy evolution. Lastly, those two groups, as showed in Sec.(2.3.2), are predominantly concentrated in a dense region at lower distance, even if the RA values suggests a possible presence of a bias in the data acquisition.

2.5 Methodology

The study has been performed by the programming language *Python*, using *Google Colab*. The full and described code is available link.

The improvement and correction of the code have been made with the support of AIs, specifically *Gemini* and *ChatGPT*.

2.6 Useful definitions

Some useful definitions, for the first part, are reported here. They will be cited frequently in the graphs' study and, since they permits to explore deeply the data, it is necessary to have a clear definition:

- **Degree Distribution:** A representation of the frequency of nodes having a particular degree in a network. It helps in understanding the overall connectivity pattern, indicating whether the network is more centralized or decentralized.
- **Degree Centrality:** A measure that indicates how many direct connections a node has. Nodes with high degree centrality are highly connected and can influence or access many other nodes quickly.
- **Louvain's Community (Detection):** A method used to partition a network into communities by maximizing the modularity. It identifies groups of nodes that are more densely connected to each other than to the rest of the network, revealing underlying network structures.

3 Graphs study

In next sections, the study of the graphs, constructed with 4 relevant characteristics, will be fully implemented. This analy-

sis is essential for identifying hidden relationships and patterns among stars. Moreover, not only it permits to predict location of possible stellar clusters, such as star-forming regions or globular clusters, but also, via the analysis of central properties, permit to highlight stars with greater influence.

In order to build the graph, it has been assumed that each node represents a star and the connections (edges) between stars depend on a certain features, explored in the related section. Precisely, in the code:

- **Nodes:** Each star is represented as a node, with attributes such as age, distance, luminosity and mass;
- **Edges:** These are added between stars that meet some specific conditions, which are given by the difference in the features;
- **Weights:** Weights represent numerical values assigned to edges, indicating the strength, the cost, the similarity or distance between connected nodes;
- **Visualization:** The graph is visualized using *"matplotlib"*, where nodes are small blue circles, and edges are gray lines. The type of representation chosen is called *"Spring Layout"*;

Aiming to have a meaningful study, one needs to choose the threshold for the creation of the connections which are reasonable, and not some random values. The initial considered choice was to use 4 values: half of the mean, the mean, 110% of the mean and two times the mean. This should have allowed to explore cases far from the mean and near it.

Specifically, the corresponding thresholds were:

- **Distance thresholds:** 1765.23, 3530.46, 3883.51, 7060.92 parsecs;
- **Luminosity thresholds:** 44.83, 89.67, 98.63, 179.33 solar units;
- **Age thresholds:** 0.15, 0.30, 0.33, 0.59 Gy;
- **Mass thresholds:** 1.36, 2.73, 3.00, 5.45 solar mass;

But this approach has resulted in graphs with "trivial" patterns, always composed of a dense inner region (point-like), surrounded by a ring, as one can immediately observe from Fig(8)(9)(10)(11).

This bimodal pattern could be caused by several reasons, but as it was shown in Sec.(2.3.4)(2.3.5)(2.3.6), the only quantity with a true bi-modularity was the metallicity. It is probably the presence of the bias to effect the graph, generating the ring. Moreover, it is straightforward to suppose that the origin of the overall triviality is related to the procedure used for creating the network:

1. It has been used only one feature per time;
2. The thresholds were probably in lower density zones, maybe capturing the rare areas;
3. No weights were introduced for the construction of the edges;

Therefore, it has been decided to use a different approach, where the features have been combined with each other, choosing the couples which are more connected; specifically: distance-luminosity, mass-age and all together. In addition,

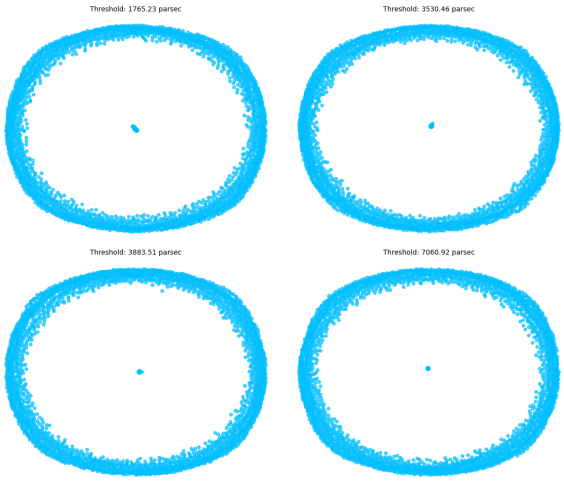

Figure 8: The image shows the graphs for the four thresholds of distance (from the top left to the bottom right) 1765.23, 3530.46, 3883.51, 7060.92 parsecs

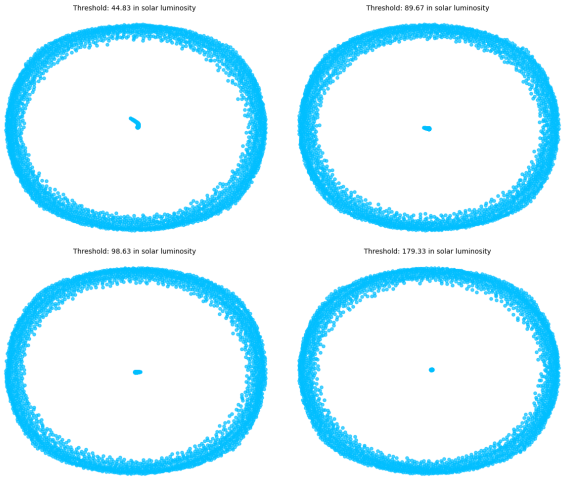

Figure 9: The image shows the graphs for the four thresholds of luminosity (from the top left to the bottom right) 44.83, 89.67, 98.63, 179.33 solar units.

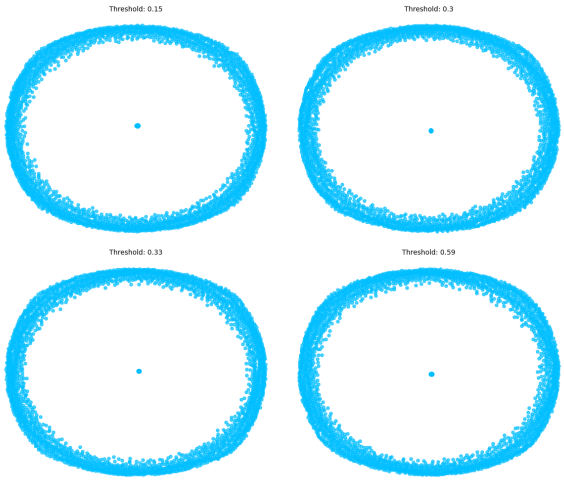

Figure 10: The image shows the graphs for the four thresholds of distance (from the top left to the bottom right) 0.15, 0.30, 0.33, 0.59 Gy.

new ranges of thresholds have been chosen to better fit the

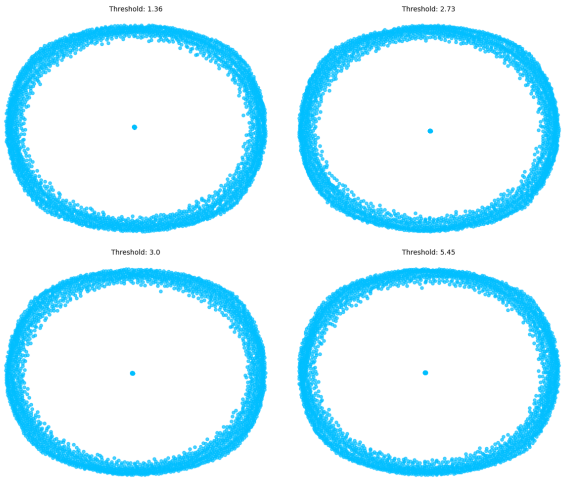

Figure 11: The image shows the graphs for the four thresholds of distance (from the top left to the bottom right) 1.36, 2.73, 3.00, 5.45 solar mass.

data, and appropriate weights, matched with the case, have been implemented. For the former:

- **Distance thresholds:** 500 (for local clustering), 1.500 (capturing most stars within ∼1 std deviation), 3.500 (captures a broad neighborhood) and 5.000 (including more distant stars but increasing connectivity) parsecs;
- **Luminosity thresholds:** 5 (for main-sequence stars), 50 (to exclude extreme giants), 150 (including moderate giants) and 500 (allowing rare supergiants) solar units;
- **Age thresholds:** 0.2, 0.3, 0.4, 0.5 Gy, focused around 1-2σ (see Tab.(6));
- **Mass thresholds:** 2.0, 2.5, 3.0, 3.5 solar mass, focused on mean 1-2σ (see Tab.(6));

And for the latter:

- **Exponential Similarity (Gaussian-based)**
  This method emphasizes close connections and heavily penalizes large differences between stars. It is particularly effective in reducing noise and preserving meaningful links. The weight is defined as:

$$\text{weight} = \exp\left(-\sum_{i}^{N}\left(\frac{\Delta X_i}{T_{X_i}}\right)^2\right) \tag{3.1}$$

  where ΔX_i is the difference between the 2 features considered and T_{X_i} is the characteristic scaling factors for the quantity.

- **Inverse Euclidean Distance**
  This method assigns higher weights to closer stars and lower weights to distant ones, making it useful for networks where similarity is determined purely by distance. The weight is computed as:

$$\text{weight} = \frac{1}{1+\sqrt{\sum_i^N \Delta X_i^2}} \tag{3.2}$$

  This approach ensures that stars with similar properties are assigned stronger connections.

Notice that there are others possible weights, like the "Manhattan similarity", but these are not interesting for the undergoing case.

Each of these weighting methods provides different insights into the network structure, hence the choice of method depends on the scientific question at hand.

Taking into account all of these considerations, it has been possible to perform a deep study of the graphs. In detail, one can see that: in Sec(3.1) the relation "Luminosity-Distance" is explored, in Sec(3.2) the "Mass-Age" features are analyzed and in Sec(3.3) all the characteristics are used to have a whole picture.

3.1 Graph: Luminosity and Distance combined

The first graph under study is the one relating "Luminosity" and "Distance", two quantities that are strictly connected. In this case, the inverse Euclidean weight method is applied, to emphasize stronger relationships between stars that are both close in distance and in luminosity.

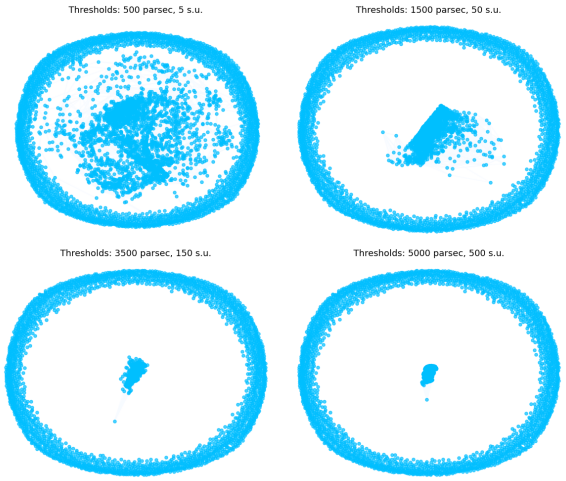

Figure 12: The image shows the graphs for the four thresholds of ddistance (parsecs) and luminosity (solar units), from the top left to the bottom right. The graphs illustrate the effect of thresholding on network connectivity.

The generated graphs, Fig.(12), represent a network of stars where connections are established based on spatial proximity and luminosity differences. The main observations coming from the graphs are:

- **Threshold: 500 parsecs, 5 s.u.:** the graph is densely connected in the central region, indicating that nearby stars tend to cluster together due their similar luminosity;
- **Threshold: 1500 parsecs, 50 s.u.:** The central structure becomes more defined, with an elongated pattern in the core of the network. This could suggest an underlying stellar substructure, possibly related to a galactic disk or a distinct stellar stream; however, considering the results in Sec.(2.3.2), it is also possible that this strange form is a consequence of the collecting procedure of the Gaia Telescope;
- **Threshold: 3500 parsecs, 150 s.u.:** the network topology changes drastically , where the main structure becomes more isolated and fragmented. The central cluster

remains dominant, while fewer connections persist in the full periphery. This may indicate the presence of a prominent stellar group surrounded by a more diffuse population;

- **Threshold: 5000 parsecs, 500 s.u.:** At the largest distance and luminosity threshold, only a small number of highly connected nodes remain: these are massive, high-luminosity stars that maintain connectivity over large distances. The majority of fainter stars at lower distances appear disconnected;

These results suggest that stars form well-defined structures when limited to lower distance and luminosity thresholds. The fragmentation observed at higher thresholds implies a bimodal distribution in the dataset, where a concentrated stellar population exists separately from more massive, bright and spatially extended stars.

This fact gives more strength to the conclusions of the data exploration and analysis, Sec.(2.4), where we have clearly distinguished two main groups of stars. Therefore, the behavior of the network is consistent with expectations for a galactic disk, which is the object studied by GAIA.

3.1.1 Degree of distribution: Luminosity and Distance combined

The degree distribution histograms provide insight into the connectivity properties of the network, highlighting the number of connections each node (star) has within different distance and luminosity thresholds.

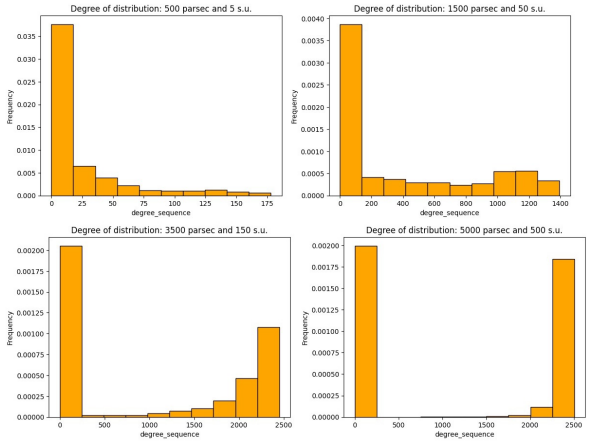

Figure 13: The image shows the histogram plot related to the degree of distribution of the nodes for the networks based on different thresholds of distance (parsecs) and luminosity (solar units).

Considering Fig.(13), the following key observations can be made:

- **Threshold: 500 parsecs, 5 s.u.**: the degree distribution is highly skewed, with most stars having very low connectivity: a few nodes present higher degrees, indicating small clusters of highly connected stars. As said in the previous section, this suggests that stellar groups are predominantly localized in small regions;
- **Threshold: 1500 parsecs, 50 s.u.**: the histogram still exhibits a strong right skewness, but with more nodes having moderate degrees. An expansion in connectivity arises, meaning more stars are linked;
- **Threshold: 3500 parsecs, 150 s.u.**: the degree distribution becomes divided in 2 sections: while many stars still have low connectivity, a secondary peak appears at high degrees, suggesting the presence of a few very highly connected hubs. Based on the previous observations, this indicates massive, bright stars acting as central nodes in stellar structures;
- **Threshold: 5000 parsecs, 500 s.u.**: the degree distribution becomes nearly bimodal: the network consists of a dominant, well-connected core and a sparse periphery, consistent with a hierarchical structure of star distributions in a galaxy.

On the whole, the outcomes indicate that as the two thresholds increase, the network transits from a collection of small local groups, to a system dominated by a few highly connected stars. This points into the direction that there is a structural segregation in the dataset, where massive and bright stars form the central hubs while the majority of fainter stars remain on the outskirts.

3.1.2 Degree of centrality: Luminosity and Distance combined

The degree centrality analysis provides valuable insights into the connectivity structure of stars in the dataset, measuring how well-connected a node (star) is.

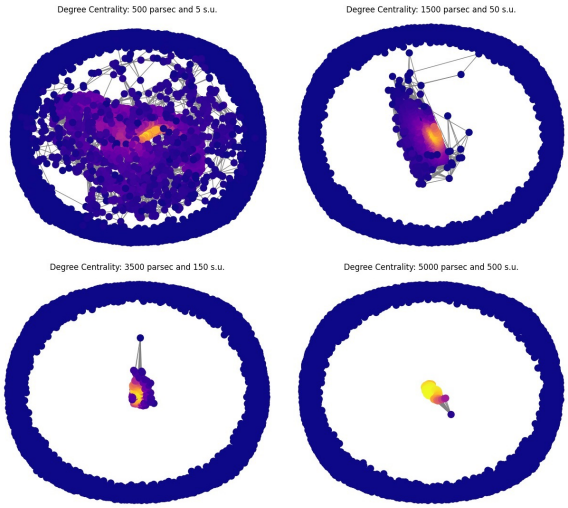

Figure 14: : The image shows the degree of centrality (via colors) for the different thresholds of distance (parsecs) and luminosity (solar units). The transition from a highly clustered, dense stellar region to a sparse, hierarchical structure dominated by a few high-degree nodes is clearly visible.

Threshold (parsec)	Graph Density	Average D. C.
500.00	0.004524	0.0045
1500.00	0.073946	0.0739
3500.00	0.205021	0.2050
5000.00	0.243491	0.2435

Table 9: Graph Analysis for different threshold values, showing graph density and average degree centrality (D. C.).

The corresponding graphs, Fig.(14), and statistical tables, Tab.(9)(10) offer a deeper understanding of the structural evolution of the stellar network as the threshold varies:

Threshold (parsec)	N. of Clusters	Modularity Score
500.00	2550	0.7557
1500.00	2515	0.5166
3500.00	2513	0.4416
5000.00	2514	0.4246

Table 10: Graph Analysis for different threshold values, showing the number of clusters and modularity score. A higher modularity score indicates stronger community structure.

- **Threshold: 500 parsecs, 5 s.u.**: the network exhibits a dense core, where numerous stars have high degree centrality. The high modularity score (0.7557) suggests that the network is highly clustered, proving the presence of a local stellar cluster. The low graph density (0.004524) and average degree centrality (0.0045) confirm that;
- **Threshold: 1500 parsecs, 50 s.u.**: the network undergoes an evident reorganization, with a more elongated core forming. High-degree nodes are concentrated in a single well-defined structure, possibly representing a stellar stream or association. The graph density increases significantly to 0.073946, while the modularity score decreases to 0.5166, indicating a transition from a highly modular system to a more integrated structure;
- **Threshold: 3500 parsecs, 150 s.u.**: At this threshold, the network becomes sparser, with only a few highly connected central nodes;
- **Threshold: 5000 parsecs, 500 s.u.**: The network is dominated by a small number of central nodes with very high degree centrality, while the majority of the stars remain isolated. The low modularity score (0.4246) and the stable number of clusters (2514) suggest that, at this scale, only the most luminous stars retain significant connectivity;

The degree centrality analysis reveals a hierarchical structure in the stellar network, which evolves with the values of the thresholds. This is exactly what it was expected from the outcomes of Sec.(3.1) and Sec.(3.1.1).

3.1.3 Louvain Communities: Luminosity and Distance combined

The Louvain community algorithm is applied to the networks at different distance and luminosity thresholds, aiming to analyze the modular structure of stellar connections. The results reveal how stellar associations are grouped based on similarity in spatial and luminosity properties.

Following the increase of thresholds from Fig.(15), one observes:

- **Threshold: 500 parsecs, 5 s.u.**: the network exhibits a densely connected central region, forming a large and compact cluster. Moreover, the strong interconnections mean the existance of a local stellar association, where stars are closely related in both spatial proximity and luminosity;
- **Threshold: 1500 parsecs, 50 s.u.**: the network structure evolves, forming a more segregated but still strongly connected core: the modularity score decreases (0.5166), indicating that the community structure is still present but less pronounced. Hence, at this scale, the influence of a single dominant stellar cluster starts to diminish, leading to more fragmented communities;

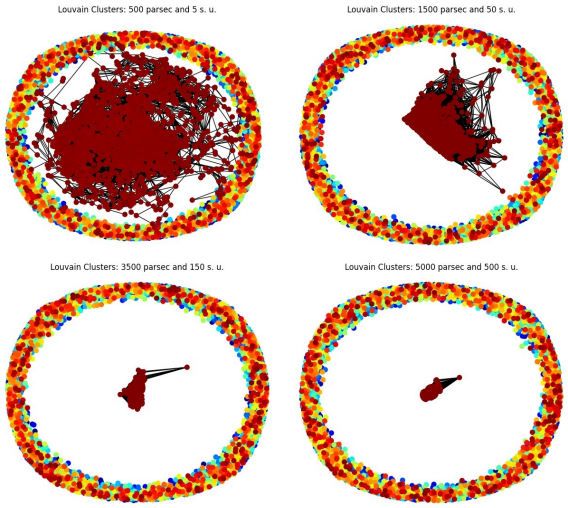

Figure 15: The image shows the graph of Louvain's communities in the case of thresholds of distance (parsecs) and luminosity (solar units), where each color represents a distinct community identified by the Louvain algorithm.

- **Threshold: 3500 parsecs, 150 s.u.**: the network becomes more sparse, with a few highly connected nodes forming a small, central cluster. The modularity score (0.4416) continues to drop, confirming the transition toward a network with weaker community structures;
- **Threshold: 5000 parsecs, 500 s.u.**: the graph is dominated by a single small, highly connected component surrounded by a vast number of isolated nodes. The modularity score (0.4246) reaches its lowest value, indicating that the community structure has nearly vanished: only the most luminous stars retain strong interconnections.

3.2 Graph: mass and age combined

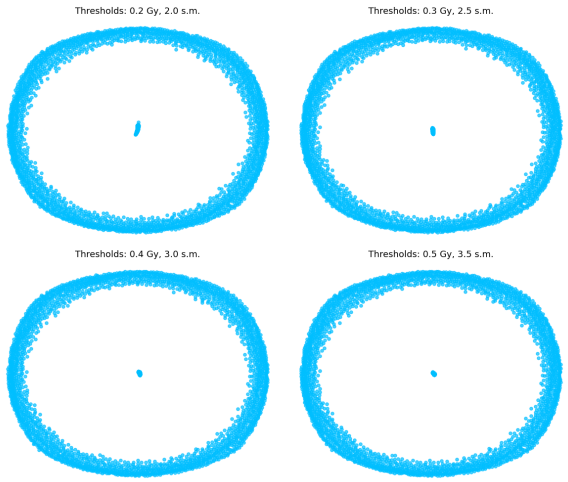

Figure 16: The image shows the graphs for the four thresholds of mass (solar mass) and age (Gyr), from the top left to the bottom right. The graphs illustrate the effect of thresholding on network connectivity, using as weight the "inverse euclidean similarity".

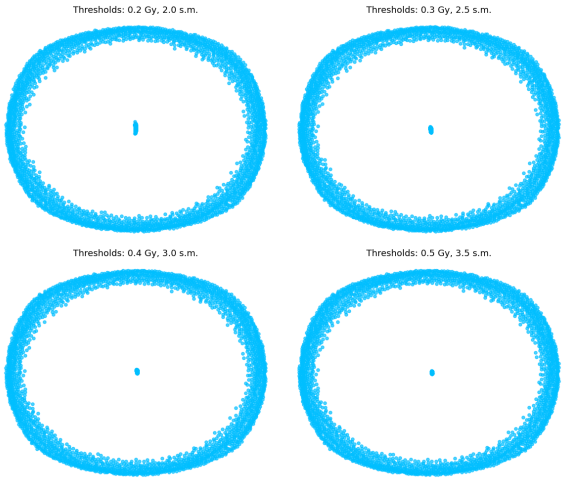

Figure 17: The image shows the graphs for the four thresholds of mass (solar mass) and age (Gyr), from the top left to the bottom right. The graphs illustrate the effect of thresholding on network connectivity, using as weight the "exponential similarity".

The two presented graph, Fig.(16)(17), presents a completely trivial pattern, even if a different weight has been used. One can conclude that the nucleus is totally unaffected by the thresholds changing, making evident the presence of two groups of stars in the dataset. Knowing so, doing degree of centrality and the others analysis will not reveal any new and additional information.

Hence, it has been decided to use a different layout to see if it was possible to gain new characteristics: the spectral layout, because it effectively positions nodes to reveal global connectivity patterns.

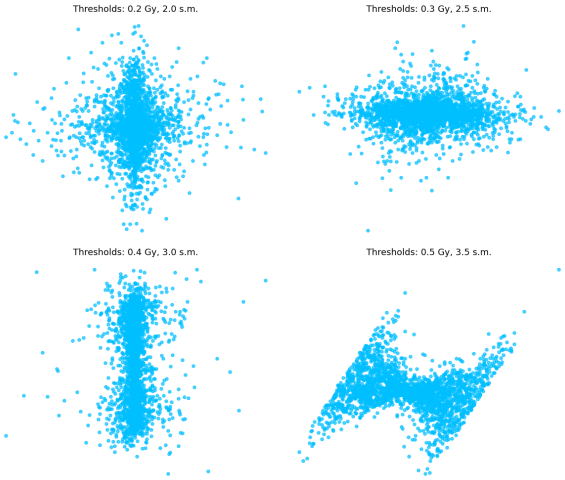

Figure 18: Network representation of stars from the Gaia telescope using an Exponential Similarity function based on stellar mass and age. Each panel corresponds to different mass and age thresholds. The spectral layout highlights the connectivity structure, revealing compact clusters at lower thresholds and elongated anisotropic distributions at higher thresholds.

Considering the new visualizatin in Fig.(18), one observes that the clustering behavior changes: for low thresholds (e.g., 0.2 Gy, 2.0 s.m.), the network forms a dense central region where many stars are connected due to their similar mass and age; then, as the threshold increases, the structure stretches, forming anisotropic distributions. The globsl network shape and structure varies significantly:

- The top-left graph (0.2 Gy, 2.0 s.m.) shows a globular shape, suggesting that many stars share very close properties;
- The top-right graph (0.3 Gy, 2.5 s.m.) reveals an elongated structure, suggesting a progressive variation in stellar properties;
- The bottom-left graph (0.4 Gy, 3.0 s.m.) presents a vertically elongated shape, potentially indicating a non-uniform spread of stellar masses in that age range;
- The bottom-right graph (0.5 Gy, 3.5 s.m.) exhibits a zig-zag pattern, possibly due to stars being grouped in distinct mass-age sequences;

From the Astrophysical point of view, the interpretation is clear: the compact nature of the lower-threshold networks suggests a homogeneous cluster of stars in the dataset; at the opposite, the anisotropic stretching in higher-threshold networks might point to substructures, such as: young stars (forming along specific mass sequences), old stars (well-separated in their mass distribution) and a possible age gradient in the dataset.

3.3 Graph: all the four features combined

The last case is dedicated to combine all the 4 features under consideration. Here, the exponential weight has been chosen since it emphasizes close connections and heavily penalizes large differences between stars; therefore, one better distinguishes similar stars, which are close.

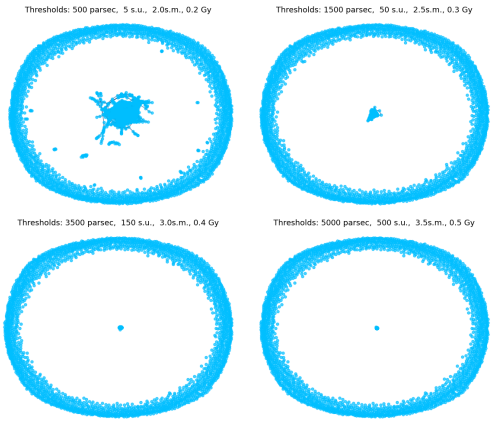

Figure 19: Graph representations of stellar networks derived from Gaia data. The networks are constructed based on varying distance, luminosity, mass, and age thresholds. The graphs illustrate the effect of thresholding on network connectivity, showing a dense core at small distances and a progressive fragmentation as thresholds increase.

The Fig.(19) provides some useful information:

- **Network connectivity:** at the lowest threshold (500 parsecs), the network exhibits a highly connected core, indicative of local stellar clustering in the Milky Way, probably the nucleus. Stars in close physical proximity tend to form dense sub-networks;

- **Threshold effect on structure:** as the threshold increases (from 1500 to 5000 parsecs), the network progressively fragments. The largest connected component becomes more isolated, suggesting that stellar properties diverge significantly beyond a certain spatial scale;
- **Outer ring structure:** the presence of an outer ring in all four panels suggests that a significant fraction of stars are not well-connected under the chosen thresholds. As previously pointed out in Sec.(2.4), this is probably a consequence of a systematic observational selection bias in the Gaia dataset;

3.3.1 Degree of distribution: All combined

The degree distribution of a network provides insight into its structural characteristics and the connectivity of its nodes.

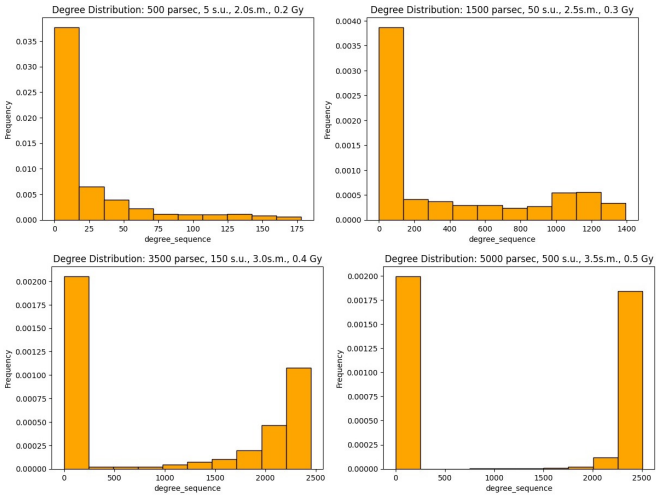

Figure 20: Degree distribution for networks constructed using different distance, luminosity, mass and age thresholds. The x-axis represents the degree (number of connections per node), while the y-axis represents the frequency of nodes with a given degree.

The histograms shown in Fig.(20) represent the degree distributions for different distance and stellar property thresholds. Evidently, it exhibits a highly skewed nature, with the majority of nodes having a low degree, while a few nodes possess exceptionally high connectivity. A structure where most stars have limited local connections is dominant, while a small subset forms highly connected hubs. Precisely speaking, as the distance (and the others) thresholds varies, one observes:

- **Small threshold:** the degree distribution is concentrated at very low values, implying that most stars in the sample have very few direct connections. This suggests a fragmented structure, where only the densest central region forms connected clusters;
- **Intermediate thresholds:** as the threshold increases, the degree distribution starts to exhibit a broader range, with some stars gaining a significantly higher number of connections. This indicates the emergence of a more interconnected structure, where the network transitions from a sparse, local connectivity regime to a more globally connected system;
- **Large threshold:** the presence of extreme values in the degree distribution suggests the formation of a highly centralized network, with certain nodes acting as dominant hubs;

The evolution of the degree distribution with increasing thresholds reflects the transition from local stellar clustering, to a more extended galactic-scale network. The exponential weighting method used in the graph construction reinforces this effect by preferentially linking similar stars, emphasizing structures formed by the chosen astrophysical properties.

3.3.2 Degree of centrality: All combined

The degree centrality in a network measures the fraction of nodes it is directly connected to, highlighting the most influential or well-connected stars in the dataset.

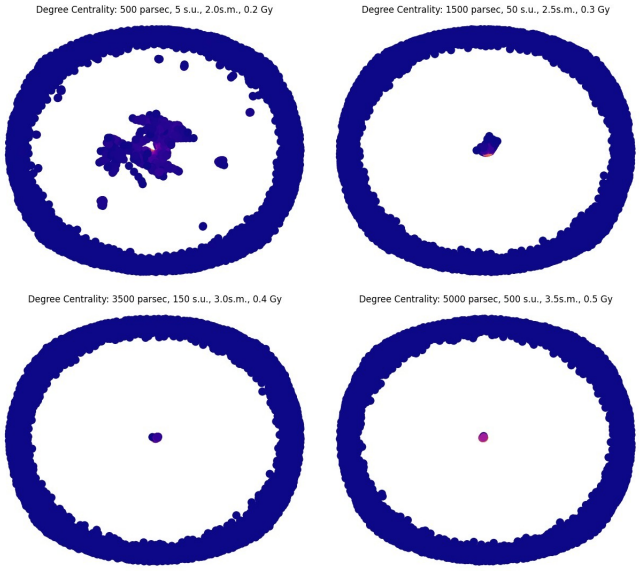

Figure 21: Degree centrality distribution for networks constructed using different distance, luminosity, mass, and age thresholds. Higher degree centrality values (lighter colors) indicate stars that act as hubs, with significant connectivity across the dataset.

Threshold (parsec)	Graph Density	Average D. C.
500.00	0.004508	0.0045
1500.00	0.073940	0.0739
3500.00	0.205021	0.2050
5000.00	0.243491	0.2435

Table 11: Graph Analysis for different threshold values, showing graph density and average degree centrality (D. C.).

Threshold (parsec)	N. of Clusters	Modularity Score
500.00	2540	0.6509
1500.00	2511	0.2715
3500.00	2511	0.0639
5000.00	2511	0.0186

Table 12: Cluster analysis for different graph thresholds, showing the number of clusters detected and the modularity score. Higher modularity values indicate well-separated structures, while lower values suggest a more interconnected network.

The visual representation of degree centrality in Fig.(22) allows us to examine how connectivity evolves as different selection thresholds are applied.

- **Small threshold:** The degree centrality is concentrated in a dense core region, where a subset of stars is significantly more connected than the rest. This is consistent with the low graph density (0.0045), reported in Tab.(11), indicating a fragmented structure where only a few stars exhibit strong connectivity;

- **Intermediate thresholds:** As the distance threshold increases, the number of connections per node increases, leading to a rise in the average degree centrality. The network transitions from isolated clusters to a more cohesive structure, although modularity scores (from the cluster analysis) indicate a significant drop, suggesting the dissolution of well-defined clusters;

- **Large threshold :** At this threshold, a dominant central node emerges, indicating that a few stars act as massive hubs connecting the majority of the dataset. The network has lost most of its modularity (0.0186), indicating that the galaxy-wide network is largely interconnected with little structural separation;

From Tab.(11), one sees that as the threshold increases, graph density rises and degree centrality follows a similar trend. However, this comes at the cost of modularity: while the lowest threshold exhibits a well-separated clustering structure (0.6509 modularity score), at highest values, modularity drops to nearly zero. Thus, the distinct stellar groupings observed at lower thresholds gradually merge into a large-scale, highly connected network.

These results suggest that stars in the Gaia dataset exhibit different levels of clustering at varying scales. Small thresholds isolate local stellar associations, such as open clusters, while large thresholds lead to a highly interconnected graph, possibly representing galactic-scale structures. The presence of dominant hubs at higher thresholds suggests that some massive or luminous stars serve as central points in the connectivity landscape.

3.3.3 Louvain Communities: All combined

The Louvain algorithm has been applied to detect communities in the stellar network at different threshold values. The results can be analyzed in relation to graph density, degree centrality and modularity scores.

- **Low threshold**: an highly clustered central region appears, indicating a dense stellar grouping. The high modularity score (0.6509) suggests that well-defined communities exist;

- **Intermediate thresholds**: the modularity score drops significantly, indicating the merging of several smaller clusters into larger and approaching a more homogeneous configuration;

- **High thresholds**: the network is nearly fully connected, and the modularity score drops further to 0.0186. Most nodes are part of a single large community, reflecting a highly interconnected structure;

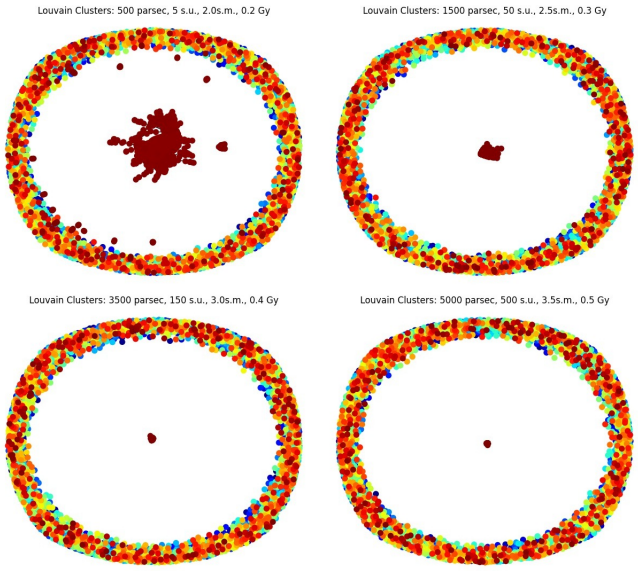

Figure 22: Louvain community detection applied to the stellar network at different distance thresholds (500, 1500, 3500, and 5000 parsecs). Each node represents a star, and colors indicate different detected communities. At small thresholds, tightly clustered communities are evident, with high modularity scores indicating well-separated groups. As the threshold increases, communities merge into a more interconnected structure, reducing modularity. This reflects the hierarchical clustering nature of the stellar dataset, where local stellar associations transition into a large-scale galactic structure.

The Louvain clustering results indicate that the network exhibits hierarchical clustering, with a dense core and a more sparsely connected outer shell. The transition from multiple small communities to a nearly uniform structure suggests that stars initially form local associations before merging into larger structures at greater distance thresholds. The clustering structure may reflect the underlying astrophysical distribution of stars, where gravitationally bound systems are evident at smaller thresholds, while larger-scale distributions become apparent at greater distances.

3.4 Conclusions on graph analysis

The Sec.(3) has explored stellar clustering and connectivity using network science techniques applied to Gaia telescope data. The stellar parameters: mass, age, distance and luminosity were used to construct graphs with different threshold values, aiming to deep in the relationships between stars. Two different weighting methods were employed: *Exponential Similarity*, which enhances local clustering by penalizing large differences, and *Inverse Euclidean Similarity*, which prioritizes close stellar connections.

On the whole, the analysis has revealed appealing outcomes: the stellar distribution is shaped by a combination of local clustering and large-scale hierarchical structures. At small thresholds, dense stellar groupings are evident, likely corresponding to star-forming regions or open clusters. As the threshold increases, stars transition into a highly interconnected network, possibly reflecting the large-scale structure of the Milky Way, even if the small sample (around 2500 stars) is not capable of reproduce a full galaxy. The degree distribution and centrality analysis confirm the presence of massive and luminous

stars as key network hubs, underlining their role in shaping the connectivity of the stellar population.

The decrease in modularity at higher thresholds shows that small-scale stellar groups dissolve into broader galactic structures, reinforcing the idea that the Milky Way's stellar distribution exhibits both local and global clustering behaviors. Lastly, the Louvain community analysis brings to the idea that local stellar environments are crucial for understanding the fundamental structure of stellar systems, while large-scale networks reveal global galactic trends.

The results suggest that the Milky Way exhibits both localized clustering and large-scale interconnectivity, driven by stellar dynamics and evolutionary processes. Further studies, integrating additional parameters like metallicity and spatial velocity, could provide deeper insights into the evolutionary processes governing these stellar networks. Besides, a focused study on the central hub could be gives details information about the center of our galaxy, highlight the main features of this special regions, which is characterized by an intense stellar activity and stellar evolution.

4 Data analysis and Neural Network

A "Neural Network" is a computational model inspired by the human brain. It consists of interconnected layers of neurons (nodes) that process data by learning patterns and relationships. These models are trained using data, where weights and biases are adjusted to minimize errors and make accurate predictions.

There are many advantages in using Neural Networks in data analysis: ability to learn complex patterns, high accuracy in predictions, scalability, adaptability are only few of them.

Looking in this direction, the following paragraphs explore these aspects, aiming to express most of the strengths and weaknesses of the Neural Network approach.

4.1 Simple Perceptron

A simple Perceptron is the most basic form of a neural network, used primarily for binary classification tasks. It is a linear classifier that models how a single neuron in the human brain might operate. Even its simplicity, it is able to show a great power in studying the data.

The inital step of the analysis is splitting the total data into 2 subgroups, corresponding to 20% and 80% of the total amount. The first set is called *"test set"* and the second one is called *"train set"*; the names are connected to their function: the train set is used to train the model and the test set is reserved to evaluate the model's performance on unseen data. However, the split is not a simple 20-80 split (e.g. taking the first 20% of the data and then the remaining 80%), but it is implemented by a function, *train_test_split*, which permits to keep the same distribution in the 2 sub-datasets. Otherwise, it is highly possible to miss the main characteristics of the data, obtaining 2 (sub)datasets whit different behavior; hence the model will not be able to produce predictions. As matter of facts, the goal of the splitting is to ensure the model's ability to generalize effectively, making possible to prevent overfitting and permit a fair evaluation.

Moreover, all over the columns, it is reasonable to take into account only the features which are really related to the one to estimate. In this specific case, the relevant columns of the dataset are: log(g), [Fe/H], effective temperature (Teff), radius flame (Rad-Flame), luminosity (Lum), mass (Mass). The explanation for these choices is written in Appendix, Sec(6.1). It has been chosen to use the radius estimated by "Flame" since also mass, luminosity and age are computed by means of it.

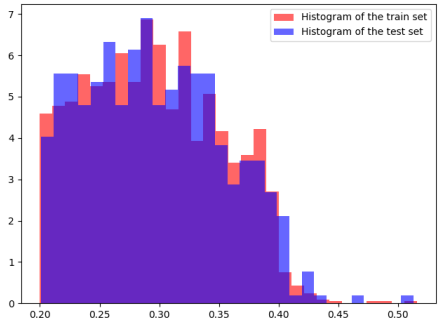

Figure 23: The image shows the 2 histograms for the train and test sub-datasets

In the image is presented an example of the 20-80 split; it is simply an example since the *train_test_split* function creates the sub-datasets randomly each time.

It is clearly possible to claim that the two datasets are well balanced, with a major concentration a lower values, which are the younger stars.

4.1.1 Architecture, training and performance

After doing so, the S.P. model can be implemented. The full description of the Neural Network is given in Table(4.1.1).

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	224
dense_1 (Dense)	(None, 1)	33
Total params:		**257 (1.00 KB)**
Trainable params:		**257 (1.00 KB)**
Non-trainable params:		**0 (0.00 B)**

Table 13: Model architecture and parameters

The model chosen has a single layer, without considering the input and output ones, and it is composed by thirty two neurons, with a total number of trainable parameters of 257.

The number of parameters in the first dense layer is calculated as follows:

- The dense layer has 32 output units (or neurons).
- The input to this layer is connected to each of these 32 neurons.
- If the input features are "n", the number of weights for each output unit is "n"
- Additionally, each neuron has one bias parameter.

Thus, since the input layer has 7 features, one get:

$$\underbrace{(7 \times 32)}_{weights} + \underbrace{(32 + 1)}_{biases} = 257 \tag{4.1}$$

where with *"weights"* one means the parameters that determine the importance of each input feature in a neural network, and with *"biases"* one means the additional parameters added to the input of a neuron, allowing the model to shift the activation function left or right.

The model is trained for 10 *"epochs"*, which is one full cycle through the training dataset, and, in this specific case, one gets the values reported in Tab.(14).

Model	Architecture	Final Loss	Final MSE	R^2
Model SP	[32]	0.0045	0.0231	-6.04

Table 14: Final values after training the simple perceptron for 10 epochs.

where *"Final loss"* is the value of the loss function[10] at the end of the training process of a model, the *"Final MSE (Mean Squared Error)"* is a metric used to evaluate the performance of a trained model on a separate validation dataset and the *"R^2"* is a statistical measure that indicates how well a regression model fits the data.

The simple perceptron demonstrates significant limitations in modeling the dataset, as reflected by its high final MSE (0.0231) and a negative R^2 score (-6.04). This suggests the model not only struggles to generalize but performs worse than predicting the average of the target variable.

Moreover, the associated *"scatterplot"*[11]: We can conclude

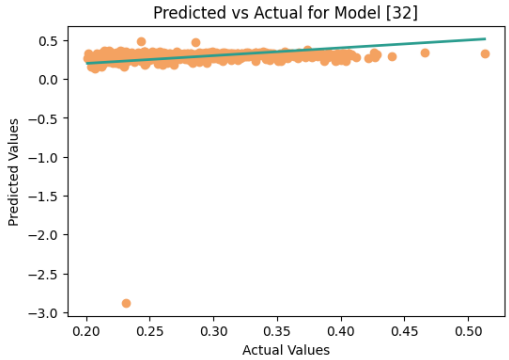

Figure 24: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data predicted by the model.

without any doubt that the simple perceptron model with a single dense layer of 32 units fails at capturing and accurately predicting the data. This fact is a direct consequence of SP model having several limitations, mainly related to the model complexity: limited neurons may not be complex enough to capture non-linear relationships in the data. Moreover, the restricted neurons in the layer brings to an insufficient learning capacity, that is to say: the architecture has a limited number of parameters (257), which restricts its capacity to model complex relationships.

One could be tempted to increase the number of neurons: this could help, but it should be part of a broader approach, in which one should expand architecture, "play" with the number of neurons and manipulate both the data and the learning process. As matter of facts, adding more hidden layers or using a more complex neural network architecture is crucial for capturing rightfully the complexity of the data[12].

This issue is not a singular case: by ri-generating the train and test sets multiple times, the result is invariant or simply slightly better. Some final values are reported to prove it:

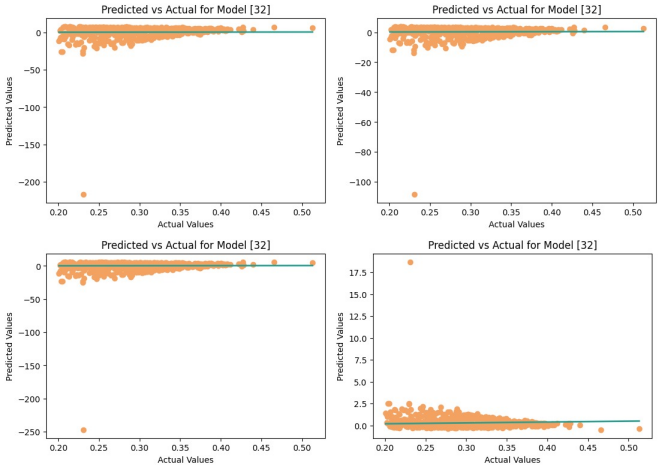

Figure 25: The image shows the scatterplots of for 4 new trainings of the SP model

Tanking into account only the R^2 score for example, the 4 presented models shows (in order from the top left to the bottom right: $\sim$ -42, $\sim$ -10, $\sim$ -47 and $\sim$ -307. Clearly, no one of them is a good model for making predictions

4.2 Multilayer perception

A simple perceptron has no hidden layers, which limits its ability to capture complex patterns in data. More advanced neural networks, like "multilayer perceptrons", use hidden layers to learn more complicated relationships. The multilayer perceptrons consist of fully connected neurons with non-linear activation functions (in this case ReLu[13]), organized in layers, notable for being able to distinguish data that is not linearly separable.

4.2.1 Architectures, training and performance

In order to capture the advanced complexity, one should extend the number of layers and neurons. However, there are no universal guidelines for determining the optimal architecture, as it depends on the specific dataset and problem at hand. Hence, mainly determined through experimentation.

In this specific case, it has been chosen to use a multiple of 32 neurons and 4 possible architectures; precisely:

1. **architecture #1:** [128, 256, 512];
2. **architecture #2:** [512, 256, 128];
3. **architecture #3:** [256, 512, 512, 256];
4. **architecture #4:** [768, 128];

The associated final outcomes are presented in Tab.(15).

[10]The loss function is a mathematical function that quantifies the difference between the predicted output of a model and the actual target value.

[11]The "scatterplot" is a graph in which the values of two variables are plotted along two axes, the pattern of the resulting points revealing any correlation present.

[12]To be fair, this is not the only way: 1) one could consider using additional features or transforming existing ones to help the model learn; 2) one could add regularization (or dropout) to control overfitting when one increases model complexity; 3) one could experiment with other models (like decision trees, random forests) or more advanced deep learning architectures (like CNNs or LSTMs).

[13]In the context of artificial neural networks, the rectifier or ReLU (rectified linear unit) activation function is an activation function defined as the non-negative part of its argument: ReLU (x) = max (0 , x) where x is the input to a neuron.

Architecture	Final Loss	Final MSE	R^2
$[128, 256, 512]$	0.2308	2.8200	-857.4032
$[512, 256, 128]$	0.1824	0.2570	-77.2217
$[256, 512, 512, 256]$	0.0116	0.2325	-69.7731
$[768, 128]$	0.0318	0.0471	-13.3417

Table 15: Final statistics in the training of multi-layer perceptron.

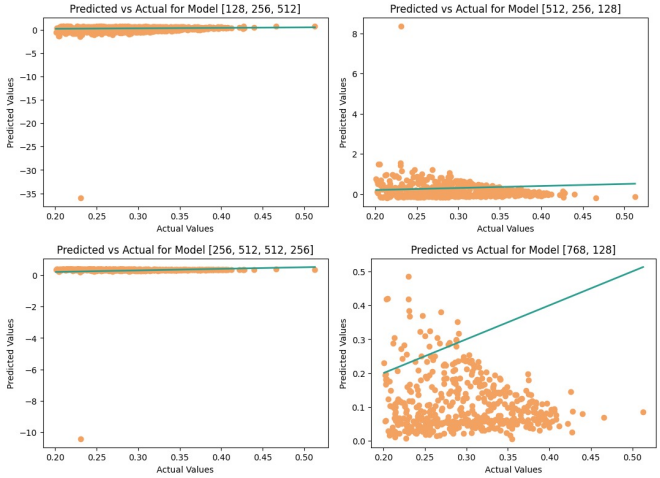

Figure 26: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

From the 4 scatterplots, Fig.(26), and the final parameters,Tab.(15), it is evident that the network is still failing to predict the data accurately. This is reflected in the consistently poor R^2 scores across all tested architectures, indicating that the models do not capture the underlying patterns in the dataset effectively.

This outcome should not be surprising. As noted in Sec.(4.1), improving performance often requires more than just varying the architecture and the number of neurons. Additional considerations such as data preprocessing or incorporating regularization techniques may be necessary.

4.2.2 Data scaling

Data scaling is the process of transforming the values of the features of a dataset till they are within a specific range. This is to ensure that no single feature dominates the distance calculations in an algorithm, and can help to improve the performance of the algorithm.

As a matter of facts, when working with machine learning models, especially neural networks, having features on different scales can significantly impact the model's performance due to:

1. **Convergence issues:** Features with larger scales can dominate the learning process, causing the model to converge slowly or get stuck in suboptimal solutions;

2. **Biased importance:** The model might incorrectly assign more importance to features with larger magnitudes, even if they are not actually more predictive;

3. **Gradient descent problems:** During optimization, features with larger scales can cause larger gradients, leading to unstable updates and potentially overshooting optimal values;

4. **Activation function saturation:** For neural networks, large input values can push activation functions like ReLU or sigmoid into saturation regions, slowing down learning. These issues can result in poor model performance, reflected in low R^2s;

By scaling your data, one can ensure that all features contribute more equally to the learning process, potentially leading to better model performance and higher R^2s. To address this, the *"StandardScaler"*[14] will be used in the code. This will standardize the data features, giving them zero mean and unit variance, which should help the model learn more efficiently from all available features.

It is important to fit the scaler only on the training data and then use that same scaler to transform both training and test data. This prevents information leakage from the test set into the scaling process.

Architecture	Final Loss	Final MSE	R^2
$[128, 256, 512]$	0.0192	0.1753	0.8328
$[512, 256, 128]$	0.0101	0.1763	0.8318
$[256, 512, 512, 256]$	0.0100	0.1446	0.8621
$[768, 128]$	0.0129	0.2639	0.7483

Table 16: Final statistics in the training of multi-layer perceptron, after having scaled the data.

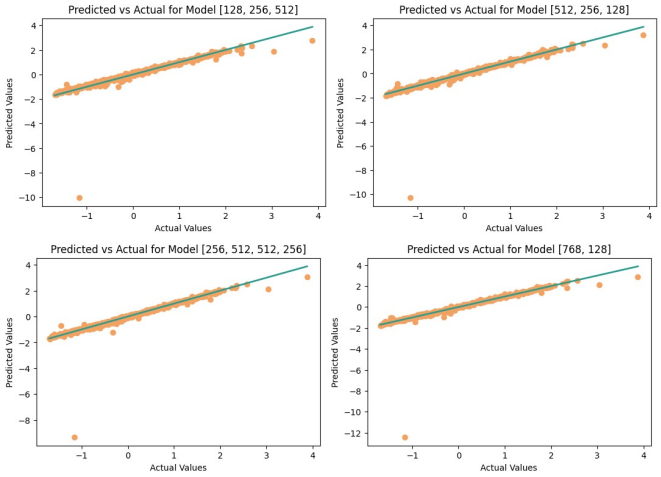

Figure 27: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

From the results in Table (16) and the scatterplot in Fig.(27), the scaling process has completely changed the outcomes: the final R^2 scores are now positive and much closer to 1, indicating that the models are capturing the underlying patterns in the data effectively.

This improvement highlights the importance of feature scaling in neural networks. Having all input features within a comparable range prevents issues such as vanishing gradients or bias in weight updates caused by disproportionate feature magnitudes.

Due to all these advantages, we will work only with scaled data from now on.

[14] The StandardScaler" (from scikit-learn) is a preprocessing tool that standardizes features by removing the mean and scaling to unit variance. This is particularly useful when the data features have different scales, which can significantly impact the performance of many machine learning algorithms.

4.2.3 Helpfull layers

It is important to note that the efficiency of these layers can vary significantly depending on the specific dataset and model architecture. What works well for one person may not work as well for another, or vice versa. This variability does not indicate that anyone's approach is inherently "wrong" it simply reflects the complex nature of neural network optimization.

1. **Dropout Layer:** Dropout is a regularization technique used to prevent overfitting in neural networks. It randomly sets a fraction of input units to 0 at each update during training time, which helps prevent units from co-adapting too much;

2. **Batch Normalization Layer:** Batch normalization is a technique used to improve the stability and performance of neural networks. It normalizes the inputs to a layer for each mini-batch, which can speed up training and act as a regularizer;

In the following, these 2 procedure has been study in several cases, aiming to explore their relations and effects, combined and not. Specifically:

A) Dropout after the first Dense layer, experimented with a rate from 0.2 to 0.5;

B) Batch Normalization before the first Dense layer;

C) Batch Normalization after the first Dense layer;

D) Combining Dropout and Batch Normalization;

E) Combining Dropuout and Batch Normalization;

The detailed form is explained in Appendix, Sec.(6.2).

It has been chosen to use as test architecture for this study the one with best R^2; thus in the example case, the best R^2 is (0.8621) with the associated architecture [256, 512, 512, 256].

Results for Case A:

Architecture	Final Loss	Final MSE	R^2
[256, 512, 512, 256]	0.0365	0.1859	0.8227
[256, 512, 512, 256]	0.0459	0.1649	0.8427
[256, 512, 512, 256]	0.0615	0.1182	0.8873
[256, 512, 512, 256]	0.0677	0.1692	0.8387

Table 17: Final statistics in the training of the best architecture, for Case A.

Considering Tab.(17) and Fig.(28), the dropout was applied after the first dense layer with varying rates (0.2 to 0.5), getting back as best R^2 score (0.8873), with an MSE of 0.1182. The performance varied with dropout rates, with some settings showing moderate results. All in all, dropout successfully reduced overfitting by randomly deactivating neurons during training. However, higher dropout rates can disrupt learning since a large fraction of neurons is deactivated, whit a consequence loss in the network's ability to learn meaningful representations, especially in smaller or less complex datasets.

Results for Case B:

Considering Tab.(18) and Fig.(29), the batch normalization was added before the first dense layer to standardize inputs. As a result of this, the R^2 score was 0.6250, and the MSE was 0.3932, showing a decline compared to Case A. Batch normalization stabilized the input distribution and accelerated

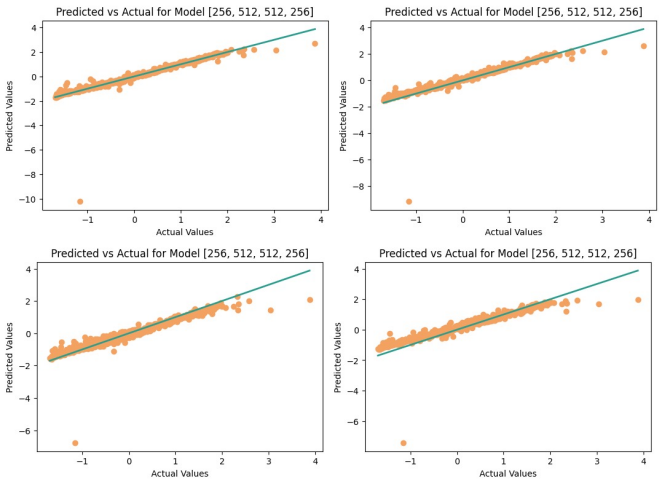

Figure 28: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

Architecture	Final Loss	Final MSE	R^2
[256, 512, 512, 256]	0.1768	0.3932	0.6250

Table 18: Final statistics in the training of the best architecture, for Case B.

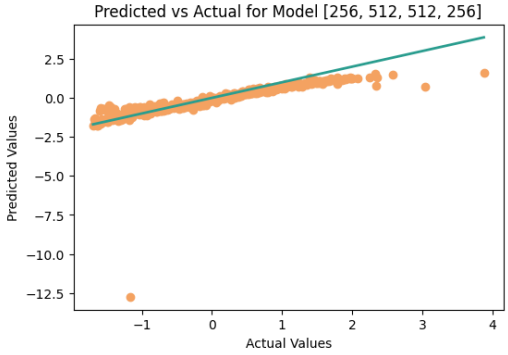

Figure 29: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

convergence, but failed to improve predictive performance significantly. It is possible to say that this placement of batch normalization might not align with the network's architecture, since it reduces its effectiveness.

Results for Case C:

Architecture	Final Loss	Final MSE	R^2
[256, 512, 512, 256]	0.0464	0.6057	0.4223

Table 19: Final statistics in the training of the best architecture, for Case C.

Considering Tab.(19) and Fig.(30), the batch normalization was applied after the first dense layer. The outputs are critically worst: the R^2 score dropped further to 0.4223, with an MSE of 0.6057. One can directly conclude that by applying batch normalization after the dense layer, the normalization is introduced at an inappropriate stage, disrupting the training process.

Results for Case D:

Considering Tab.(20) and Fig.(31), both dropout and batch normalization were combined. As a result, the R^2 scores ranged from 0.1969 to 0.7200, with the best MSE at 0.2936; the out-

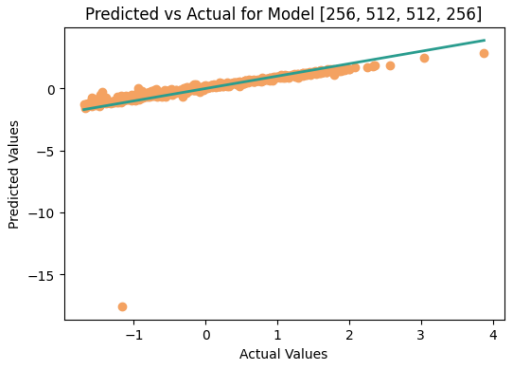

Figure 30: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

Architecture	Final Loss	Final MSE	R^2
[256, 512, 512, 256]	0.0735	0.8421	0.1969
[256, 512, 512, 256]	0.0672	0.6729	0.3582
[256, 512, 512, 256]	0.0939	0.4078	0.6111
[256, 512, 512, 256]	0.0844	0.2936	0.7200

Table 20: Final statistics in the training of the best architecture, for Case D.

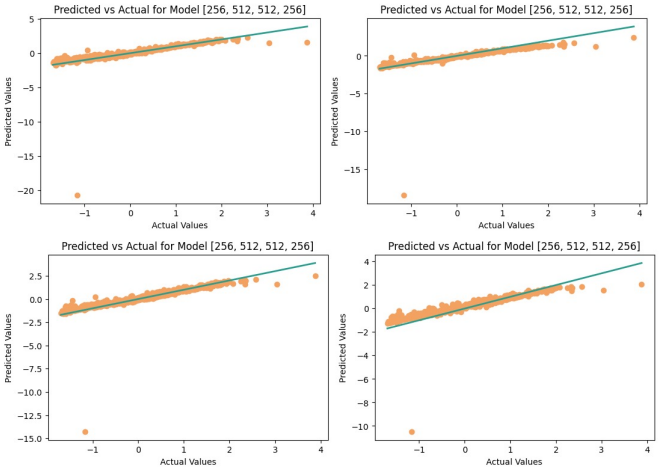

Figure 31: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

comes vary significantly across different configurations. In general, one can interpret that while dropout reduced overfitting and batch normalization improved stability, their combination might have introduced conflicting effects. From the outcome, it is possible to learn that over-complicating the network with multiple techniques requires careful tuning, and it is not adapt for every cases like our one.

Results for Case E:

Architecture	Final Loss	Final MSE	R^2
[256, 512, 512, 256]	0.0496	0.1131	0.8921

Table 21: Final statistics in the training of the best architecture, for Case E.

In the last case, Tab.(21) and Fig.(32), dropout and batch normalization were combined with refined parameters and placement. The R^2 score reached 0.8921, the best among all cases, with an MSE of 0.1131. The refined configuration of both techniques allowed the network to generalize well while

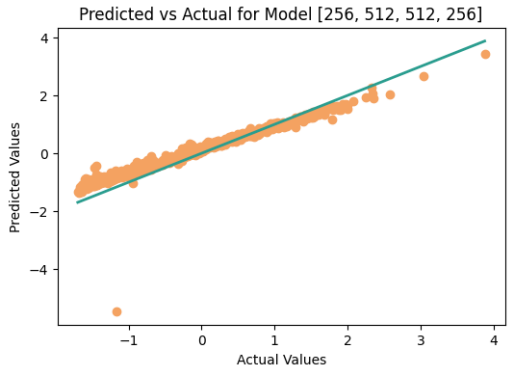

Figure 32: The image shows the "scatterplot" of the data: in green there is the line of perfect prediction, in orange the data

maintaining stability during training. Proper placement and tuning of dropout and batch normalization proved highly effective, yielding the best performance.

Taking into account everything, one can conclude undoubtedly:

- Case A showed promising results, demonstrating that dropout can effectively reduce overfitting when tuned properly;
- Cases B and C highlighted the importance of proper placement for batch normalization, as inappropriate use can degrade performance;
- Case D revealed that combining techniques without refinement may introduce inconsistencies or overcomplicate the training process;
- Case E demonstrated the potential of combining dropout and batch normalization effectively, achieving the best results through careful tuning;

4.2.4 Regularization

Regularization is a technique used to prevent overfitting in neural networks. It adds a penalty term to the loss function to discourage large weights, which can help the model generalize better.

L1 and L2 regularization are two common types of regularization:

- **L1 regularization** adds a penalty term proportional to the absolute value of the weights. This encourages sparsity in the weights, meaning many weights will be exactly zero.
- **L2 regularization** adds a penalty term proportional to the square of the weights. This

encourages smaller weights overall, which can help prevent overfitting.

Here, the implementation of code has taken adavantage again of the best model, based on R^2: architecture [256, 512, 512, 256] with the score of 0.8621.

On the whole, it results:

- **L1 regularization,** Fig.(33): forces sparsity in the weights by penalizing large coefficients. This likely helped the model focus on essential features, resulting in competitive performance with an R^2 score close to 0.891. The scatterplot (Fig. 33) likely shows tighter clusters around

Regularizer	Final Loss	Final MSE	R^2
L1	0.2734	0.1144	0.8909
L2	0.1162	0.2552	0.7566
L1_L2	0.2860	0.1099	0.8952

Table 22: Final statistics of the model training after the introduction of the regularizers.

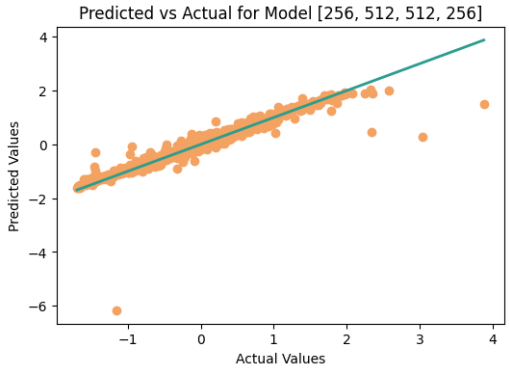

Figure 33: The image shows the scatterplot for the L1 regularizator.

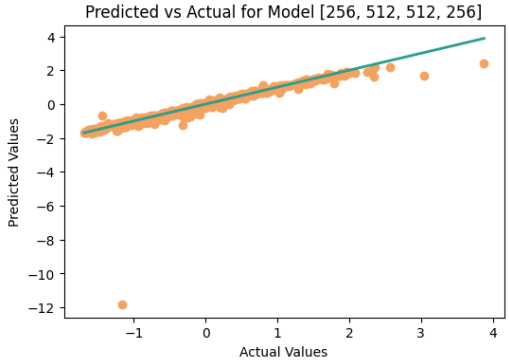

Figure 34: The image shows the scatterplot for the L2 regularizator.

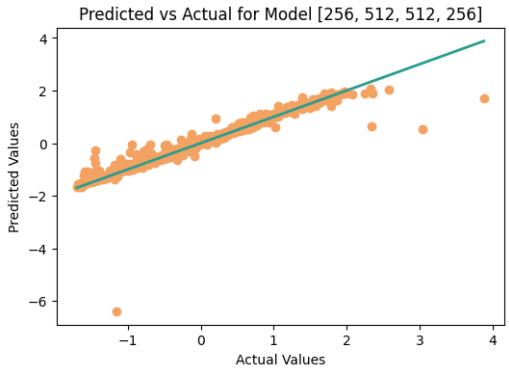

Figure 35: The image shows the scatterplot for the L1 and L2 regularizator combined.

the line of perfect prediction compared to other regularizers;

- **L2 regularization:** Fig.(33): penalizes large weights more uniformly than L1, preventing overfitting but without the same sparsity-inducing effect. While it achieved a reasonable MSE, the R^2 score suggests it struggled to generalize as effectively as L1. This may indicate that the network still relied on irrelevant features or noise;

- **Combining L1 and L2 regularization,** Fig.(35): leveraged the strengths of both: L1's sparsity-inducing properties and L2's overall stabilization. This produced the best R^2 score (0.8952), indicating a strong balance between reducing overfitting and maintaining generalization. The scatterplot (Fig. 35) likely shows the closest alignment with the line of perfect prediction among the three regularizers;

In the undergoing case, one can conclude: L1 Regularization effectively reduced overfitting, leading to high predictive accuracy. L2 Regularization provided moderate improvement but did not perform as well as L1 or L1_L2, possibly due to its inability to induce sparsity. L1_L2 Regularization emerged as the best approach, balancing sparsity and stabilization for optimal generalization and predictive performance.

Stellar ages are critical for understanding the formation and evolution of stars and galaxies, though determining them poses significant challenges. This study leverages data from the Gaia Telescope to explore these challenges through two complementary approaches. First, we apply graph-based methods to analyze relationships among stellar parameters such as distance, luminosity, age, and mass, investigating connectivity, distribution, and community structures. Second, we employ neural networks to develop predictive models, showcasing the processes of data preparation, model training, and optimization. This dual approach highlights the potential of combining graph analysis and machine learning for advancing our understanding of stellar evolution and clustering.

4.2.5 Epochs

The last section is dedicated to train the 2 best models for 100 epochs; this permits to explore how the number of cycles influences the properties of the final model.

The two model consider are: [256, 512, 512, 256] with an R^2 of 0.8621 and [128, 256, 512] with an R^2 of 0.8328.

The final values are reported in Tab.(23).

Architecture	Final Loss	Final MSE	R^2
$[256, 512, 512, 256]$	0.0062	0.1430	0.8636
$[128, 256, 512]$	0.0085	0.1077	0.8973

Table 23: Final statistics of the 2 best models after 100 epochs of training.

The final result are:

1. **Best Architecture:** this architecture provides the best balance between training and validation performance. The scatterplot for this model shows a tighter clustering around the line of perfect prediction, indicating better generalization and accurate predictions;

2. **Second Best Architecture:** Although this architecture achieved a slightly better R^2 score than the first, the scatterplot and validation trends suggest it may be prone to overfitting for some configurations, as indicated by the variability in MSE over epochs. This may point to sensitivity to the dataset or a need for additional regularization;

The $[256, 512, 512, 256]$ architecture is the most reliable choice, with a strong R^2 score and stable generalization.

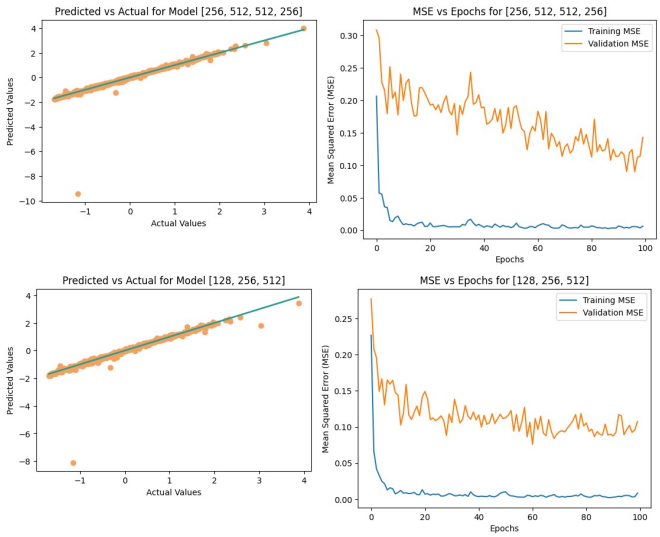

Figure 36: The image shows the scatterplots for the 2 best models and their associated mean square errors with respect to the number of epoches.

In conlusion, it is clear and evident that training for 100 epochs appears unnecessary for both architectures, as the validation MSE stabilizes well before this point (around 40 epochs). Besides, extending training further might lead to overfitting, causing the failure of the model. Hence, in a generic case the best choice is having a sufficient number of epochs to ensure the convergence, without exceeding in the number itself.

5 Conclusion

This study demonstrates the effectiveness of combining graph-based methods and neural networks to analyze stellar datasets.

The graph analysis revealed critical insights into the structural properties of stars, such as clustering, community detection and transition from local to global connectivity. This approach revealed both local clustering and large-scale hierarchical structures within the stellar population. Different weighting methods, such as Exponential Similarity and Inverse Euclidean Distance, demonstrated how stellar connections evolve across different thresholds, influencing the formation of hubs and central nodes. The degree distribution and centrality analysis confirmed that a few highly connected stars dominate the network, while most stars have low connectivity, suggesting a core-periphery structure. The Louvain community detection method showed a progressive decline in modularity at higher thresholds, indicating a transition from localized stellar groups to broader galactic structures. These findings highlight the importance of network-based astrophysical analysis for understanding stellar associations and galactic evolution.

The neural network analysis, while highlighting the potential for predictive modeling, also exposed limitations in handling the complexity of stellar data. Exploring multiple architectures, the simple models struggled to generalize effectively, emphasizing the need for advanced architectures, better feature engineering and optimized training processes.

Overall, this dual approach provides a foundational framework for further exploration of stellar evolution and clustering, illustrating the potential of integrating network science and machine learning in astrophysical research.

6 Appendix

6.1 Features

These features are most correlated to age because they are directly linked to a star's evolutionary stage:

- Mass: Higher-mass stars evolve more quickly due to faster nuclear fusion, making mass a primary determinant of a star's lifespan and age.
- Luminosity: Luminosity increases as a star ages and progresses through its lifecycle, especially during later stages like the red giant phase.
- Effective Temperature : Temperature changes with age, as stars evolve from hotter, younger main sequence phases to cooler, older giant phases.
- Radius: Stars expand as they age, particularly in later stages, causing their radius to increase significantly.
- log(g): Surface gravity decreases as stars age and their radius increases, making log(g) inversely related to age.
- [Fe/H] (Metallicity): Older stars typically have lower metallicity because they formed in the early universe when fewer heavy elements were present, making [Fe/H] a proxy for stellar age.

These features collectively reflect the physical and chemical changes that occur during a star's evolution, correlating strongly with age.

6.2 Specific structure for Dropout and Batch Normalization

In this section, the internal position for implementing Dropout/Batch Normalization is maked explicit:

A) Dropout after the first Dense layer, experimented with a rate from 0.2 to 0.5:

- Input layer
- Dense layer
- Dropout layer (rate: 0.2-0.5)
- ...
- Output layer

B) Batch Normalization before the first Dense layer;

- Input layer
- Batch Normalization layer
- Dense layer
- ...
- Output layer

C) Batch Normalization after the first Dense layer;

- Input layer
- Dense layer
- Batch Normalization layer
- ...
- Output layer

D) Combining Dropout and Batch Normalization;

- Input layer
- Dense layer
- Batch Normalization layer
- Dropout layer
- ...
- Output layer

E) Combining Dropuout and Batch Normalization:

- Input layer
- hidden layer
- ...
- Batch Normalization layer
- Output layer

References

[1] DataSet *Gaia* 626 000 stars from DR3 (Data release 3), 2023, *"Gaia data set for stellar classification(DR3)"* Available Here

[2] Yude Bu, Yerra Bharat Kumar, Jianhang Xie, Jingchang Pan, Gang Zhao, and Yaqian Wu, 2020 June 25, *"Estimation of Stellar Ages and Masses Using Gaussian Process Regression"*, The American Astronomical Society The Astrophysical Journal Supplement Series, Volume 249 [DOI 10.3847/1538-4365/ab8bcd]

[3] Book *Dive into Deep Learning*, 2021, *"Interactive deep learning book with code, math, and discussions"* Available Here