Logo

Description automatically generated

**Datamining Phase 1**

|  |  |
| --- | --- |
| **ID** | **Name** |
| 172739 | Omar Atef |
| 171820 | Karim Mohamed |
| 166436 | Mansour Wael |

* **Introduction**

Clustering is popular technique for grouping data of same attributed or similar to each other same group(cluster). Clustering algorithms are classified under two different labeled 1) partitional 2) hierarchical. The purpose of clustering is to find valuable data from unsupervised data. In addition, this technique became applied in very application areas like biology, image processing, etc... In our case the clustering used in data mining to increase the learning ratio of data. After finished the cluster phase and disturbed the data according to their similarity there is second phase called “Cluster Analysis” this phase purpose is to observe the characteristics of each cluster and retrieve information from it like measure the dissimilarity between data.

* **Dataset**
* **Dataset Description**

The aim of the dataset is to highlight the fact that stars in space conform with a pattern illustrated with the diagram proposed by the two scientists Ejnar Hertzsprung and Henry Norris Russell. This enables us to make use of different features of stars in space in order to classify them into their different star types.

A screenshot of a computer

Description automatically generated with medium confidence

Figure (1): Hertzsprung-Russell Diagram. Depicts the correlation between the magnitude of the star and the temperature and divides each portion of the correlation into a spectral class.

The features of the different stars in the dataset were collected from the web, specifically the book named “Stars and Galaxies” by Michael Seeds and Dana Backman and Wikipedia. Some missing data was deduced from the equations of astrophysics like:

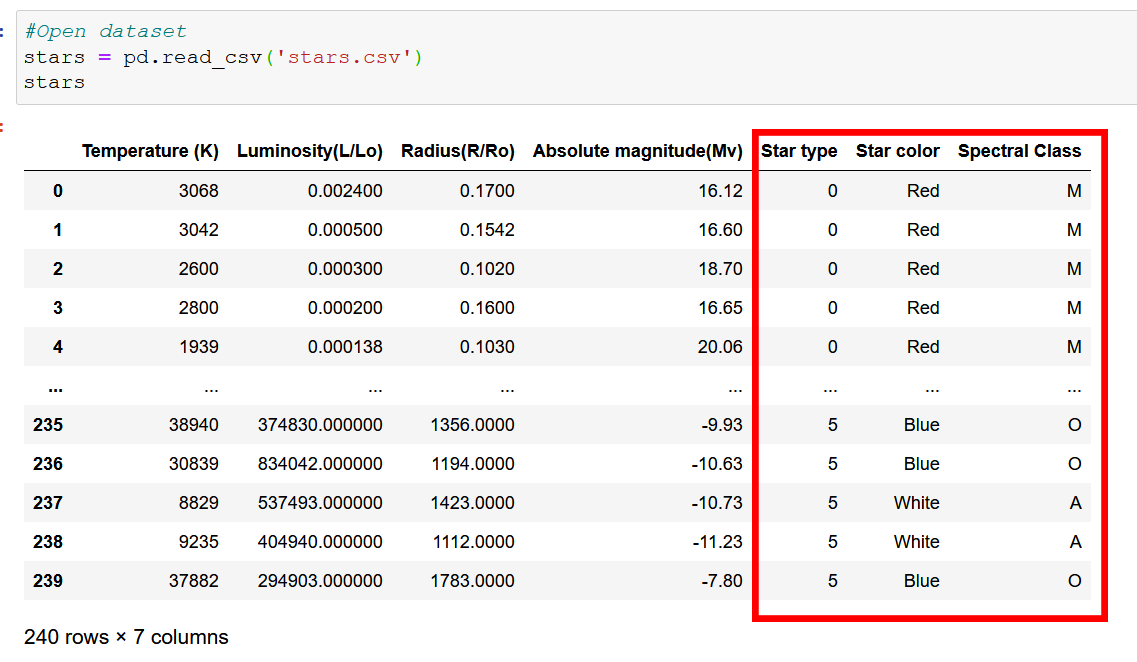
1. **Stefan-Boltzmann's law of Black body radiation:** In order to deduce the luminosity of the star.
2. **Deducing the radius of the star using parallax:** (The parallax is the shift in the position of an object observed by the eye when the object is viewed along two different lines of sight)
3. **Wienn's Displacement law:** In order to deduce the temperature on the surface of the star by the aid of the wavelength of thermal radiation.
4. **Absolute magnitude relation:** the absolute magnitude is the observed size of an object if viewed exactly 10 parsecs away (32.6 light-years) and it used to measure luminosity as it also reflects how much energy is emitted by the star.

The dataset consists of 240 samples and 7 features or attributes including one target column. The 7 attributes are:

* **Absolute magnitude (Mv -** **absolute visual magnitude of a star-):** the absolute magnitude is the observed size of an object if viewed exactly 10 parsecs away (32.6 light-years).
* **Star Color:** The observed color of the star.
* **Absolute Temperature (measured in K or Kelvin):** thermodynamic temperature on a scale of absolute zero which is the lowest temperature that a gas thermometer can measure.
* **Spectral class:** The types depicted in the Hertzsprung-Russell Diagram (O,B,A,F,G,K,M) that organizes stars in a sequence of temperatures from hotter stars to cooler stars. It is also related with the magnitude of a star.
* **Relative Luminosity (L/Lo):** Ratio of the star’s luminosity to the sun’s luminosity.
  + **Luminosity:** the true brightness of a celestial star in space (not the apparent visible/observed brightness affected by distance in light years).
* **Relative Radius (R/Ro):** Ratio of the star’s radius to the sun’s radius.
* **Star Type:** The target which is type of the star from the various types defined by scientists of astronomy.
* **Data pre-processing**

To start with, the stars dataset has no null values in any of its attributes so we will neither need to fill missing values nor drop records with null values.

When observing the dataset, we find that some of the dataset attributes are somehow contiguous; meaning that for example in the Star Type attribute rows containing the value 0 exist one after another at continuous row indexes and this consequently results in the lack of randomization in the dataset and this makes the K-Means clustering algorithm in this case work on a much easier task rather than leveraging its full potential in clustering samples belonging to the same class that are scattered all over the dataset rows. In order to solve this, we must add some randomization to the dataset which makes samples of the same class scattered across rows of the dataset in a non-contiguous manner. The randomization was applied by the shuffle function of the sklearn library.



Looking at the distributions for the 4 core numeric attributes we have (excluding Star type as it a column containing only 6 possible values ranging from 0 to 5, we figure that the 4 attributes Temperature (K), Luminosity(L/Lo), Radius(R/Ro), and Absolute magnitude(Mv) have wide distributions and lie on different scales and thus, these 4 attributes need to be normalized in order to be on the same scale ranging from 0 to 1 rather than different scales.

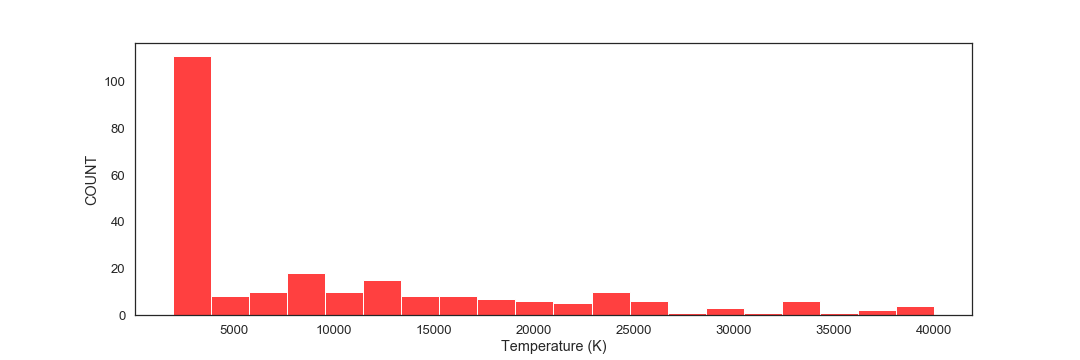


Figure (2): Distribution of Temperature (K) attribute

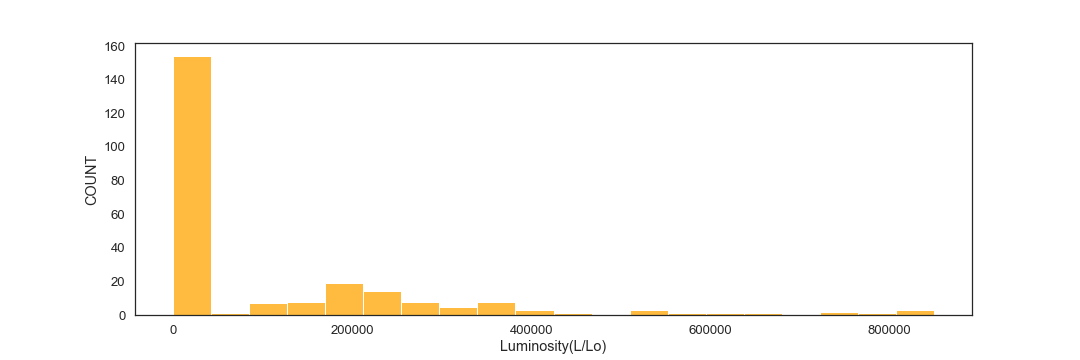


Figure (3): Distribution of Luminosity(L/Lo) attribute

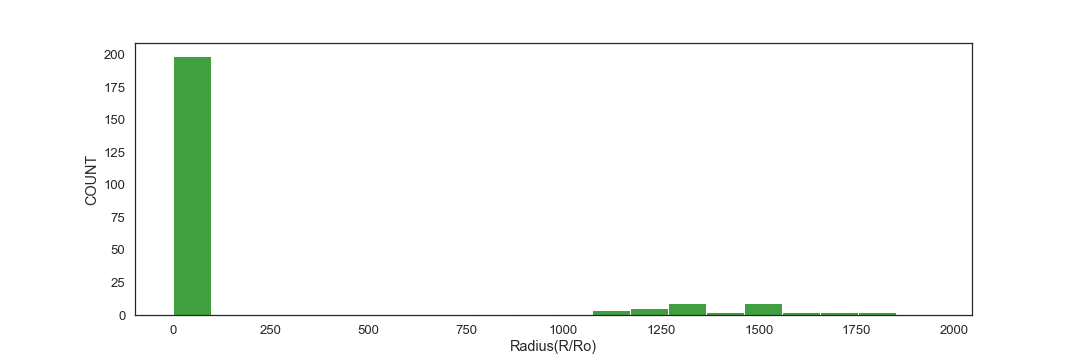


Figure (4): Distribution of Radius(R/Ro) attribute

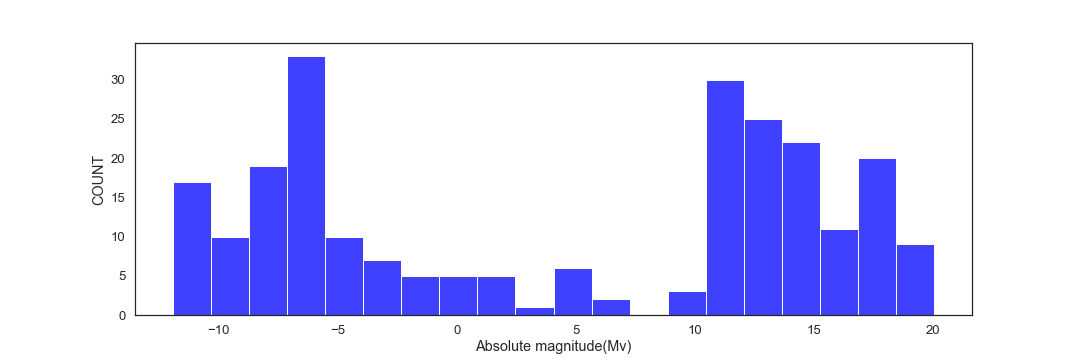
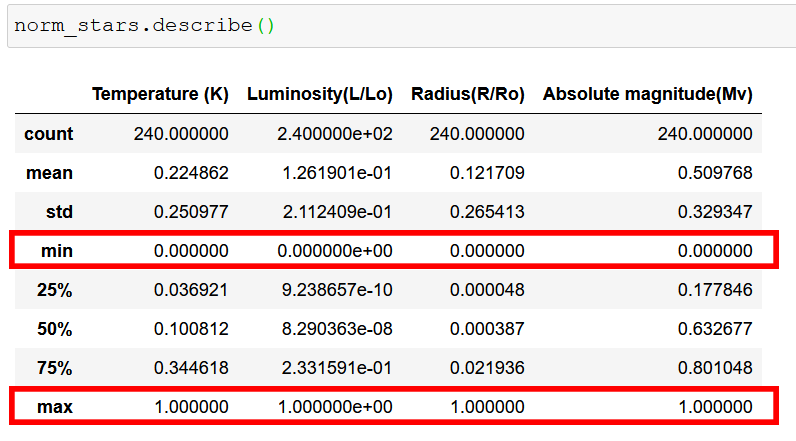


Figure (5): Distribution of Absolute magnitude(Mv) attribute

We apply the following equation to do the normalization:

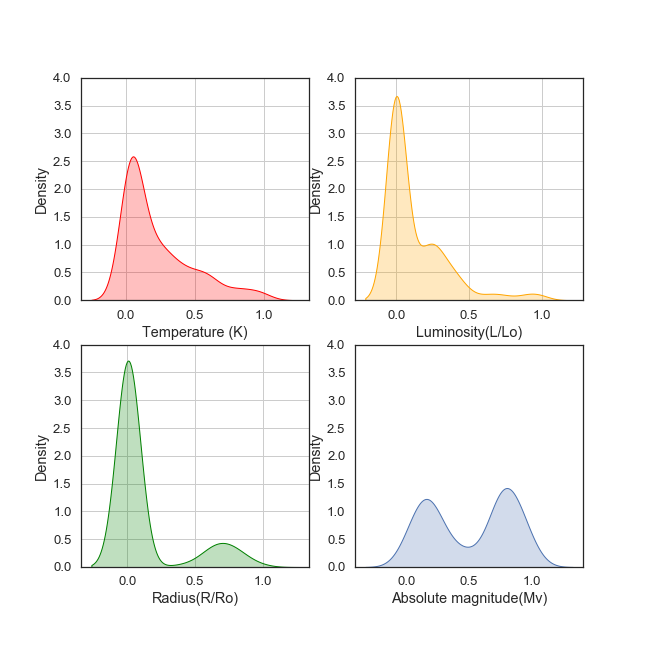
**Xnormalized =**

Where X represents a sample value in a specific column of the dataset, X min represents the minimum value in a specific column and X max represents the maximum value in a specific column. After applying the normalization function on the 4 core numeric attributes, the result that was displayed by the describe method of pandas verified that all the values of each of the 4 core numeric columns are now ranging from 0 to 1.

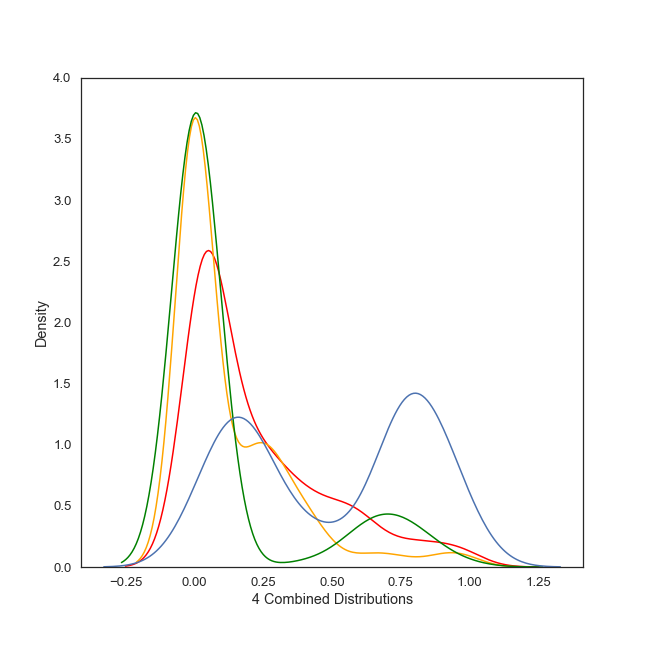


Figure(6): The output displayed by the describe method after applying normalization

After normalization we look at the KDE (Kernel density estimation) to observe the difference in the distributions of the probability density functions of the 4 core numeric attributes. It can be observed from the 4 plots below that the plots for the attributes Temperature (K), Luminosity(L/Lo) , and Radius(R/Ro) are right skewed with the peaks of the distributions on the left while the distribution of the Absolute magnitude(Mv) attribute has two peaks and none of them lies in the center. The skewness of the distribution of the Absolute magnitude(Mv) attribute was estimated by the skew function of scipy library to be -0.1207786427322446 which is negative or left skewness.

Figure (6): Separate KDE distributions for the 4 core numeric attributes

When observing the plot below for the combined KDE distributions for the 4 core numeric attributes, it can be deduced that the 4 distributions when being overlayed on top of each other, form two peaks. For the purpose of easier processing, a standardization process should be applied in order to bring the two peaks closer as much as possible and thus, be in an even more similar distribution and similar scale. The standardization process of bringing the two peaks closer must be done in such a way that the mean μ becomes 0 and the standard deviation σ becomes 1.

Figure (7): Combined KDE distributions for the 4 core numeric attributes after applying normalization

The standardization was done using the equation:

**Z =**

Where X represents a sample value in a specific column of the dataset, represents the mean of a specific column in the dataset, and represents the standard deviation – from the mean - of a specific column in the dataset. After standardization we observe that the mean of each column in the 4 core numeric attributes is now very near to 0 and the standard deviation of each column is now equal to 1.

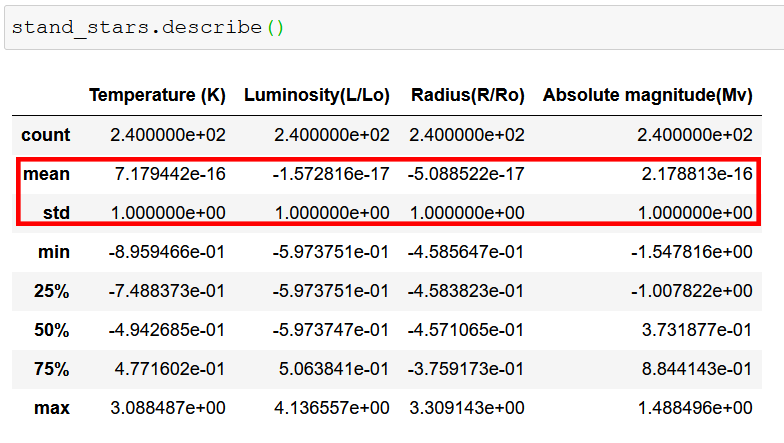


Figure (8): The output displayed by the describe method after applying standardization

Plotting the 4 attributes after being standardized results in the shape shown below. The 4 KDE plots are now closer to each other in general and they are all oriented and centered around the mean which is 0 as shown in the plot.

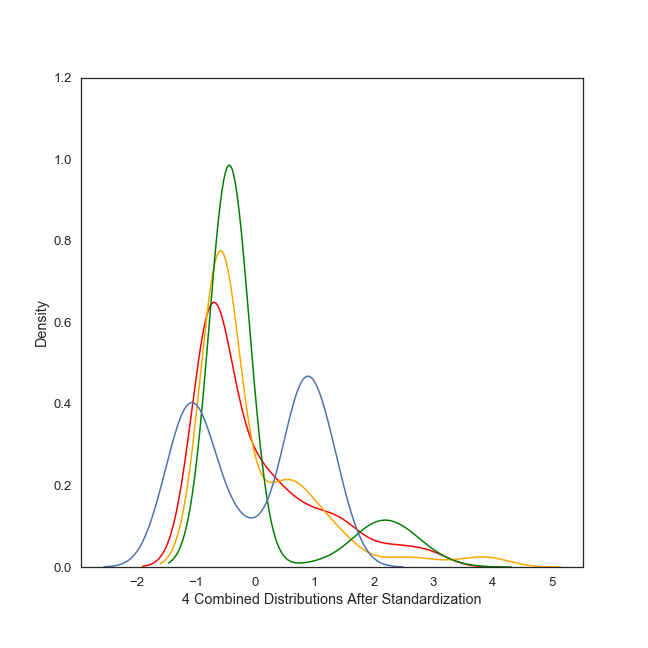


Figure (9): Combined KDE distributions for the 4 core numeric attributes after applying standardization

Next, we should convert string elements inside any attribute column into numbers to be suitable for training a neural network. Fortunately, the output class was already converted into numbers by the creator(s)/collector(s) of the dataset, namely 6 classes with numbers ranging from 0-5 which are: Brown Dwarf -> Star Type = 0, Red Dwarf -> Star Type = 1, White Dwarf-> Star Type = 2, Main Sequence -> Star Type = 3, Supergiant -> Star Type = 4, Hypergiant -> Star Type = 5. There are two other attribute columns that contain string elements and were not addressed by the author(s) of the dataset and therefore must be converted into numbers by us. To do this, we created two functions that correspond each class of each column into a fixed number on the column level.

In the Star color attribute, as soon as we used the value\_counts() method to find the unique values in the Star color column, it can be deduced that there are some repeated classes that are in their essence in fact the same. For example, in the Stars color column, Red color is unique with 112 samples, Blue is almost unique with 55 samples except for another value ‘Blue’ existing in only one record. The value ‘Blue-white’ must be treated the same as ‘Blue White’, ‘Blue white’, and ‘Blue-White’ should be considered the same. Moreover, ‘White’, ‘white’, and ‘Whitish’ should be treated as the same. Furthermore, ‘yellow-white’, ‘Yellowish White’, ‘yellowish’, ‘Yellowish’, and ‘White-Yellow’ should be treated as the same. Lastly, ‘Orange’, ‘Orange-Red’, and ‘Pale yellow orange’ should be grouped. The grouping of these values was decided based on the “Hertzsprung-Russell Diagram”. It is important to provide a unification for values written differently but are the same color as this problem might cause the unbalanced classes problem if not prevented in the pre-processing phase and consequently affect the accuracy later. Thus, it is a must to inspect the unique values in a column and give a fixed number to those values that differ and are unique in terms of wording but mean the same thing or have the same significance. The values in each column containing strings were labeled by numbers only and neither normalized nor standardized as these values always appeared to have narrow range/distribution in our dataset so the number-labelling was enough.

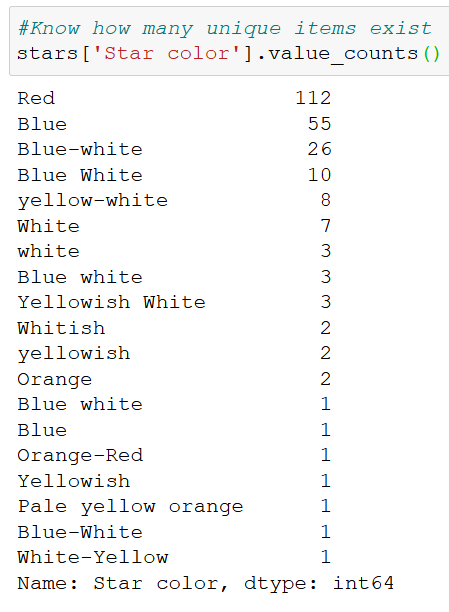


Figure (10): The output of the value\_counts method applied on the ‘Star color’ attribute

After conversion we make sure that the strings were converted to numbers successfully.

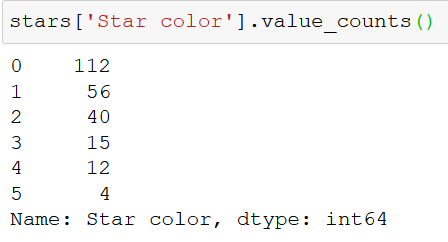
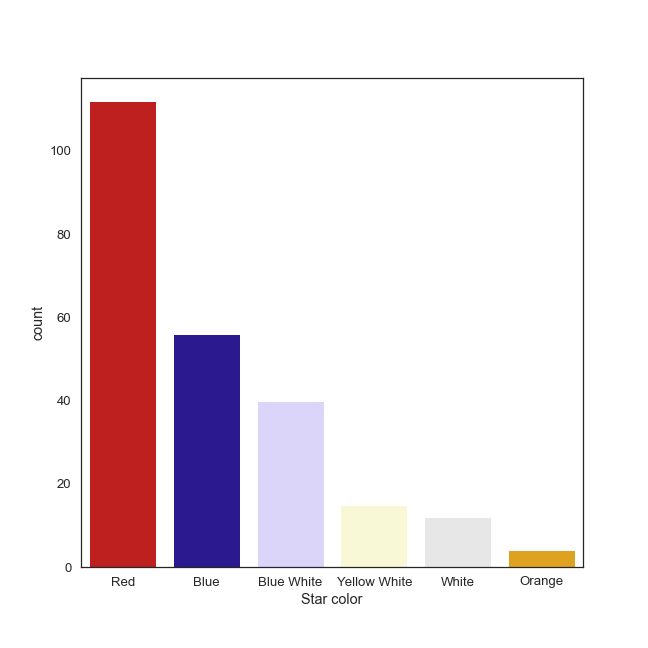


Figure (11): The output of the value\_counts method applied on the ‘Star color’ attribute after converting strings into numbers

Figure (12): A plot of available colors after grouping similar colors together

After that, each unique string value of the last column ‘Spectral Class’ is assigned a fixed number.

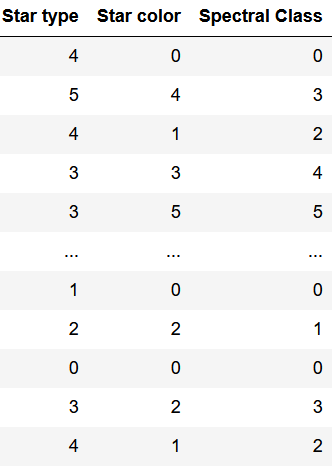


Figure (13): The 3 other attributes of the dataset after converting Star color and Spectral class into numbers

At the end of the preprocessing, a new data frame is created to hold all the 7 features which are the 4 normalized and standardized features and the 3 number-labeled features and then this newly created data frame is saved as a csv file.

* **Comparison between the 2 algorithms**
  + **K-Means clustering:**

K-mean is one of the cluster techniques that was proposed in 1967.it is unsupervised, numerical, iterative method of clustering. And its famous technique between the clustering types. As mentioned above there is 2 different type of cluster K-means is considered as partitional that utilize iteration techniques to group the data. K-mean has 2 phases to implement as below.

**Phase 1:**

Select number of groups to be clustered where is will be fixed in the initial step (usually takes K symbol)

**Phase 2:**

Each object goes to the nearest centroid to it by using “Euclidean distance” as traditional measurements.

The entire life Cycle of the Algorithm is shown as below.

1) Choose K value and the Dataset.

2) Distribute each object to the nearest centroid.

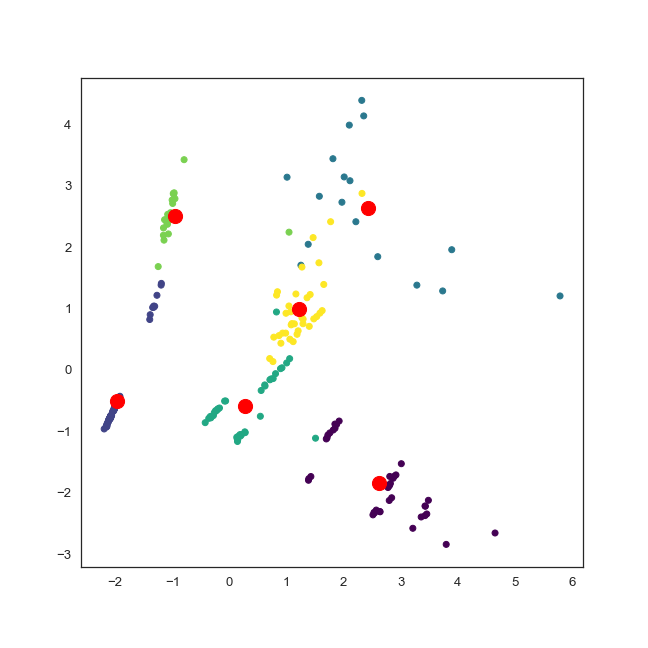
3) Calculate the new mean value of each object and update the mean value.

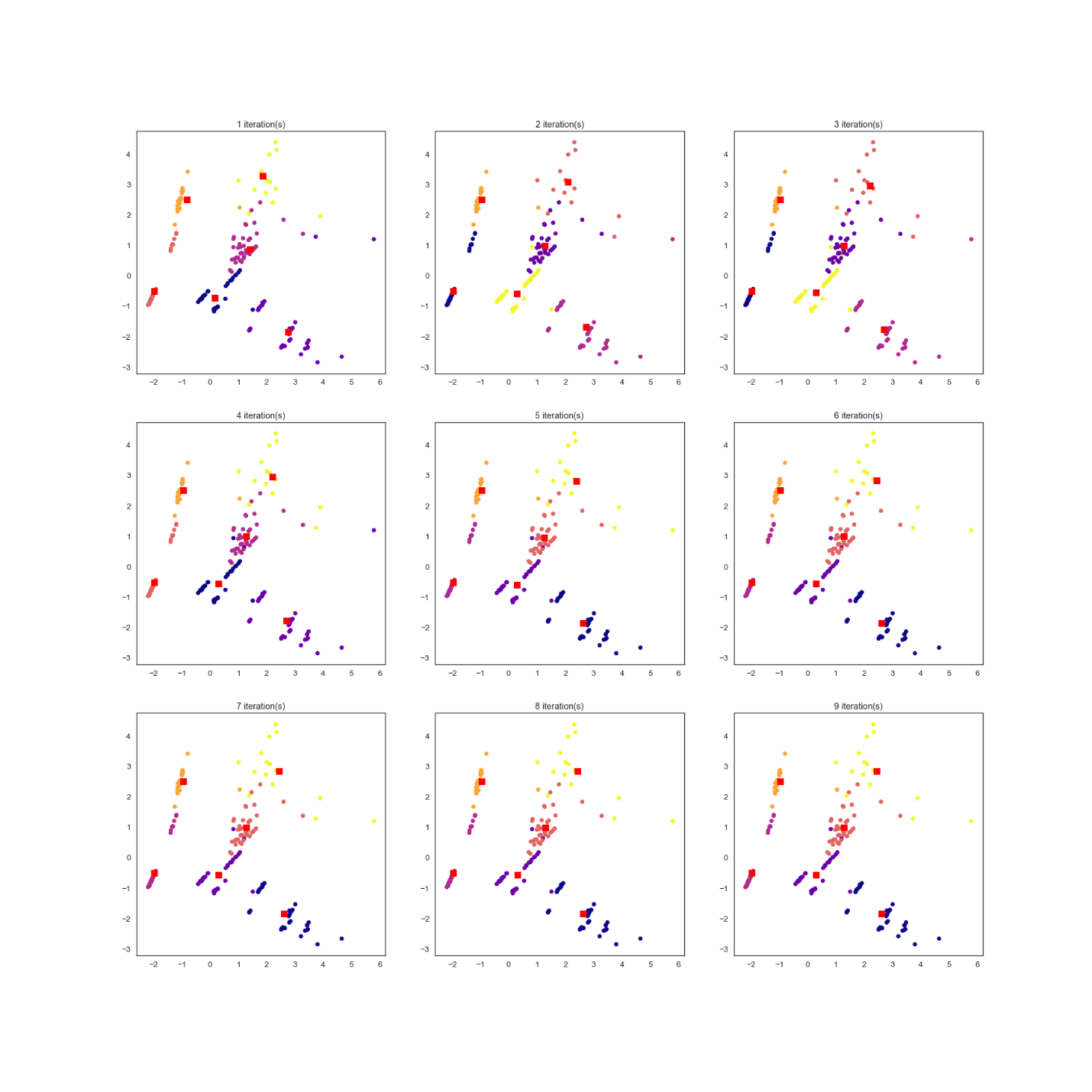
4) Repeat step 2.

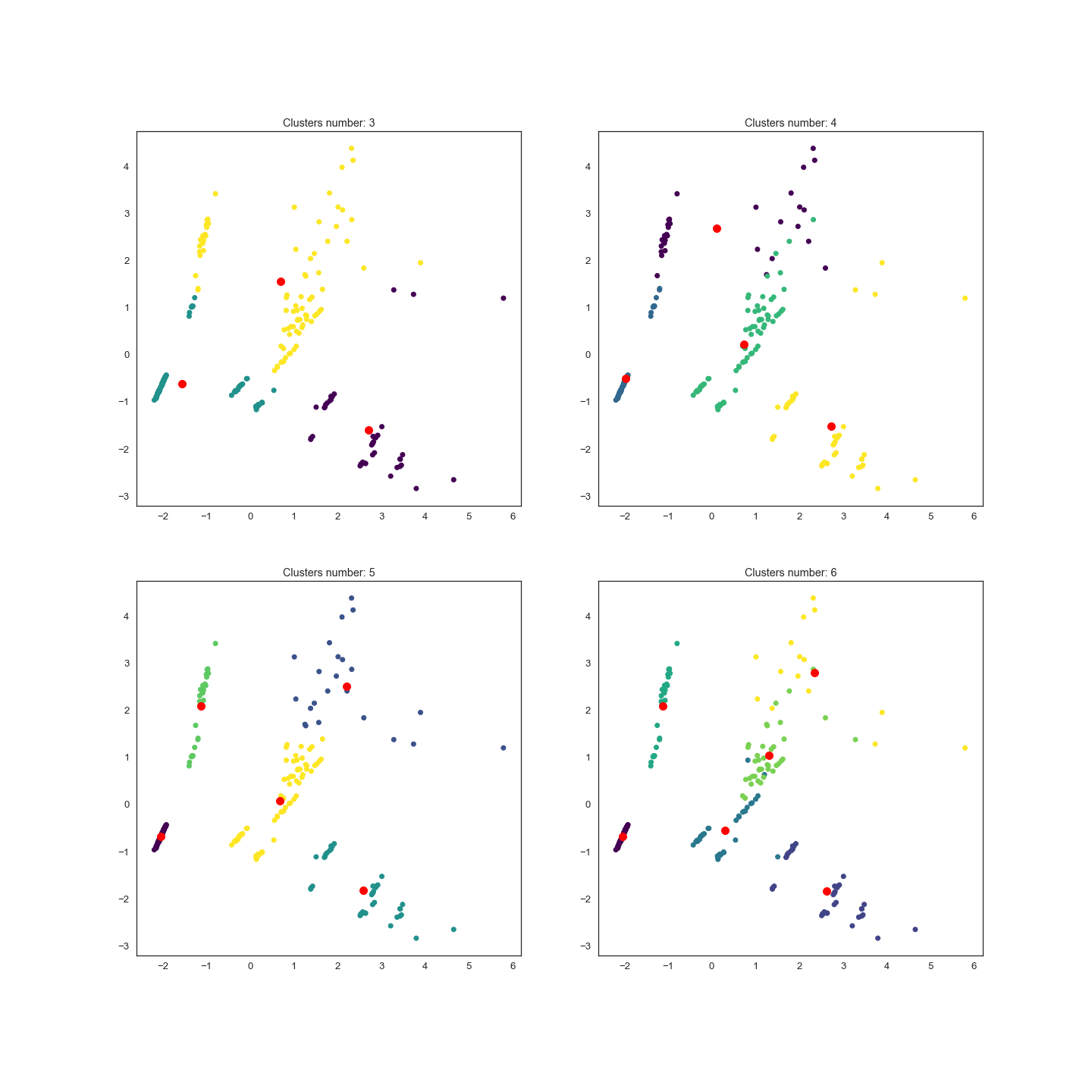
5) Until no change in mean values of each object

This algorithm is restricted by numerical values only but there is some modified algorithm that handle both numerical and categorical values. Also,

This algorithm faced some problems as defining similarity measurement to measure the distance between the object by this measure. In addition, conclude information of characteristics of object in the cluster in succinct and representative way**.**

**** Figure (14): The final output when applying K-means clustering on the stars dataset along with the final centroids of clusters

 Figure (15): The first 9 iterations of the K-means clustering algorithm applied on the stars dataset

 Figure (16): K-means clustering algorithm applied on the stars dataset using different values for K ranging from 3 to 6

* + **Hierarchical clustering:**

It is an algorithm that is used to collect a group of data according to some similarities using some mathematical formula and according to their similarity and differences and puts them in group called cluster to narrow them down. Hierarchical clustering is considered as nested clusters that is pictured as a tree, the decision of clustering some data and merging them is done according to the distance between them (dissimilarity) and this distance can be measured in various ways such like Euclidean distance, Squared Euclidean distance, Manhattan distance and Cosine distance. Euclidean distance equation considered the most common between the other it’s the direct straight line between the two points and its equation = square root of (point x – point y). while Manhattan distance measure is simple equation by of difference between two points (∑ ∣ a - b∣).

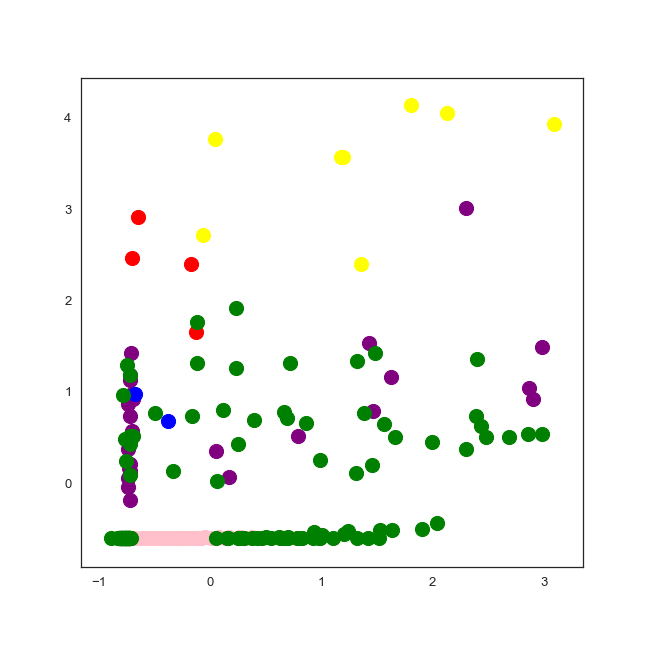


Figure (17): Agglomerative clustering applied on the stars dataset

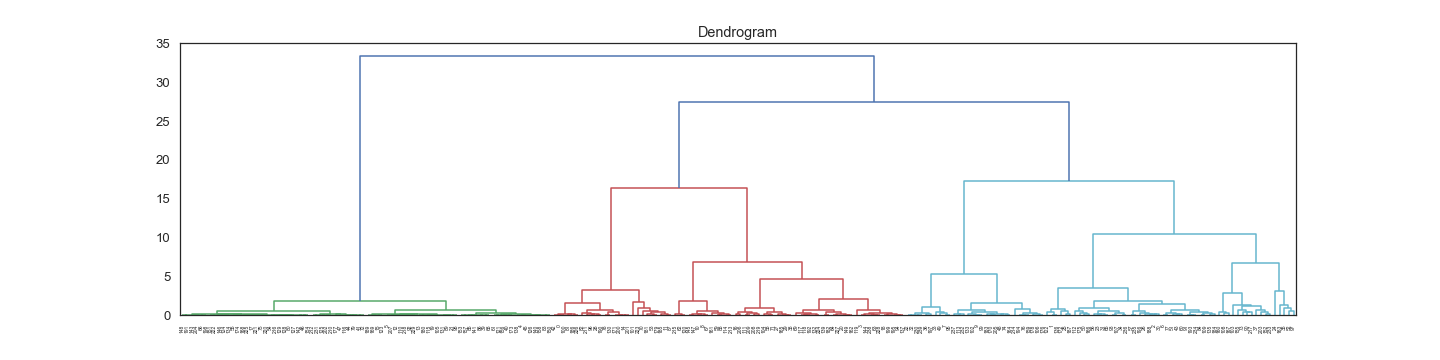


Figure (18): The dendrogram of hierarchical clustering applied on the stars dataset

* **Conclusion**

In conclusion, the K-means clustering algorithm performed better than the hierarchical algorithm as was able to cluster different groups that appeared closer in distance to each other in the scatter plot. The K-means algorithm could arrive at the final state in just few steps in python. For the next phase, some sort of generation of new samples will be needed in order to work efficiently with a neural network.

**References**

[1] M. S. Jyoti Yadav, "A Review of K-mean Algorithm," International Journal of Engineering Trends and Technology, vol. 4, no. 7, July 2013.

[2] L. D. Amir Ahmad, "A k-mean clustering algorithm for mixed numeric," Data & Knowledge Engineering, vol. 63, no. 2, p. 503–527, 2007.

[3] S. SakshiPatel, "A Study of Hierarchical Clustering Algorithms," International Conference on Computing for Sustainable Global Development, no. 2, pp. 537-541, 2015.