1/18/2022

Unsupervised learning – Stars clustering

Olivier FONTAINE

TABLE OF CONTENTS

ANALYSIS OBJECTIVE

- ➔ Main objective of the analysis that also specifies whether your model will be focused on clustering or dimensionality reduction and the benefits that your analysis brings to the business or stakeholders of this data.

The objective of this analysis is to use a set of clustering methods to classify stars in groups of stars with same characteristics and compare it to existing scientific groupings.

So, here we are focusing on clustering rather than dimension reduction.

I am an astrophysics enthusiast but I am no science expert on this field. I will try to find insights from my work with my current scientific knowledge.

DATA SET DESCRIPTION

➔ Brief description of the data set you chose, a summary of its attributes, and an outline of what you are trying to accomplish with this analysis.

The data set name is Star Dataset.

The data set is available on KAGGLE: https://www.kaggle.com/deepu1109/star-dataset

The data set has **240 rows** (stars) and **7 columns** (stars characteristics).

Column	Features	Data type	Data type	Values
0	Absolute Temperature	int64	Numerical	Surface temperatures of stars in Kelvin
1	Relative Luminosity	float64	Numerical	Luminosity of stars calculated with respect to sun(L/Lo)*
2	Relative Radius	float64	Numerical	Radius of stars calculated with respect to sun(R/Ro)*
3	Absolute Magnitude	float64	Numerical	Absolute Visual magnitude(Mv) of several stars
4	Star Type	object	Categorical	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
5	Star Color	int64	Numerical	Colors of each star after Spectral Analysis
6	Spectral Class	object	Categorical	spectral classes of each star(O,B,A,F,G,K,,M)

* The Luminosity and radius of each star is calculated w.r.t. that of the values of Sun.

- Lo = 3.828 x 10^26 Watts (Average Luminosity of the Sun)
- Ro = 6.9551 x 10^8 m (Average Radius of the Sun)

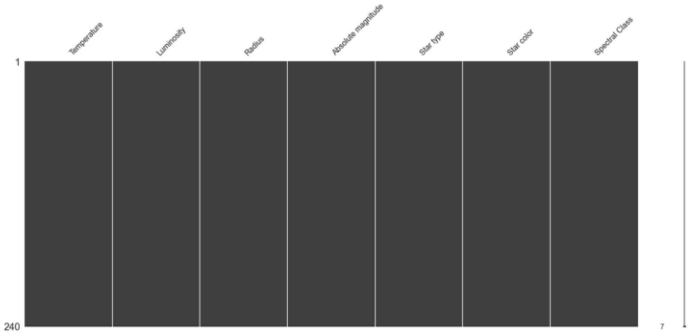
So, in this project we will try different clustering techniques to identify stars groups and compare them to the current "Star type" field in the dataset.

DATA EXPLORATION, DATA CLEANING & FEATURE ENGINEERING

➔ Brief summary of data exploration and actions taken for data cleaning and feature engineering.

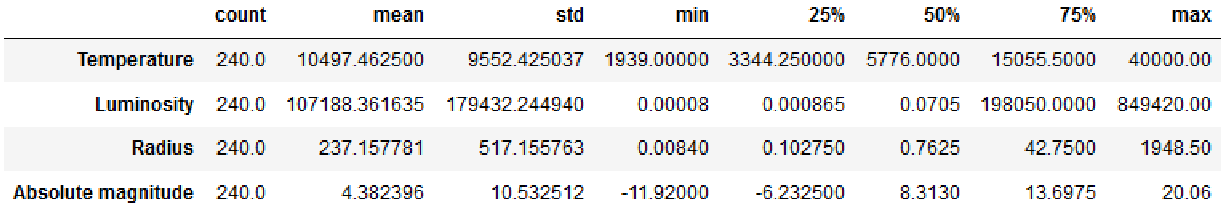
MISSING VALUES

In this dataset, there are no missing values.



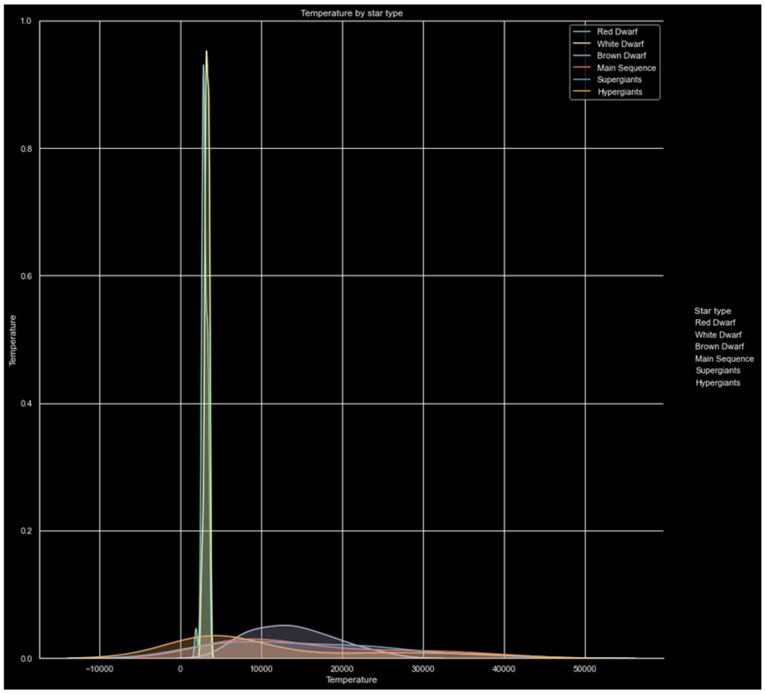
DATA EXPLORATION

Here are the statistics on numerical fields:



	count	mean	std	min	25%	50%	75%	max
Temperature	240.0	10497.462500	9552.425037	1939.00000	3344.250000	5776.0000	15055.5000	40000.00
Luminosity	240.0	107188.361635	179432.244940	0.00008	0.000865	0.0705	198050.0000	849420.00
Radius	240.0	237.157781	517.155763	0.00840	0.102750	0.7625	42.7500	1948.50
Absolute magnitude	240.0	4.382396	10.532512	-11.92000	-6.232500	8.3130	13.6975	20.06

Here is the Temperature distribution by star type:

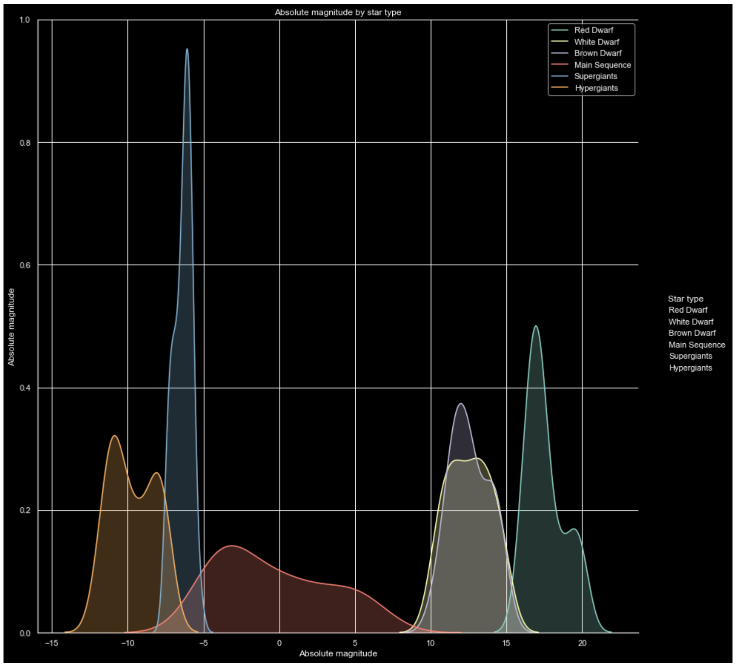


For all type of stars, the temperature distribution is right skewed.

The hottest stars are:

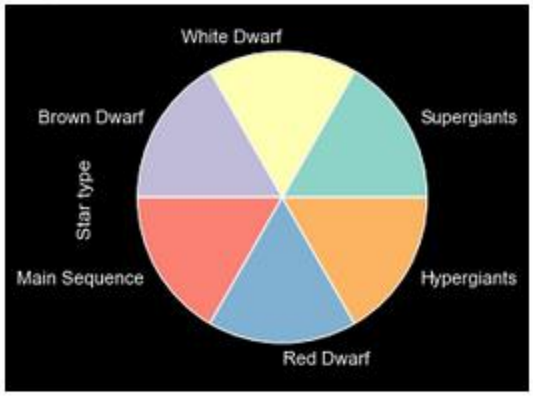
- Main sequence
- Supergiants
- Hypergiants

Here is the Absolute magnitude distribution by star type:



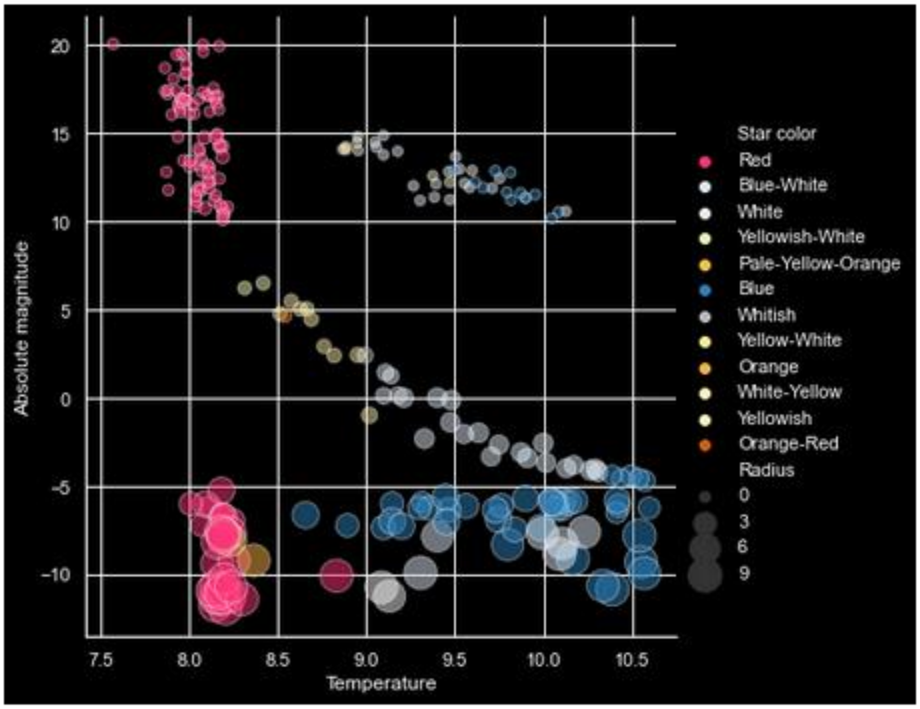
For all type of stars, the absolute magnitude distribution has 2 modes.

Now, the 6 “Star types” are balanced in this dataset:



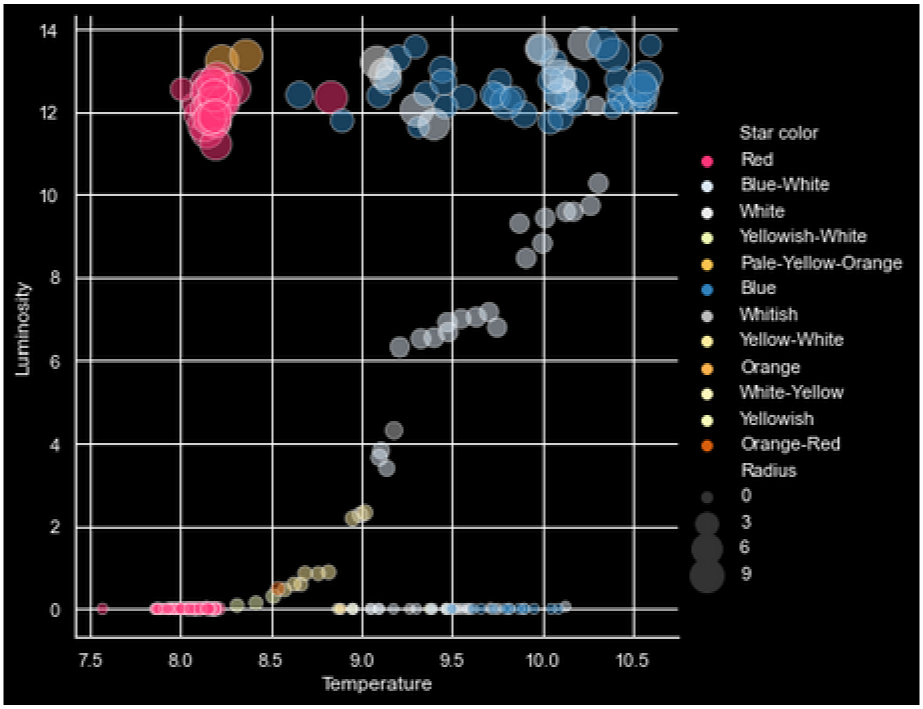
Now, let’s have a look to relationships between certain variables.

Below is displayed the relationship between the “Absolute magnitude” and the “Temperature”. Stars appear with their color and relative size in the scatter plot.



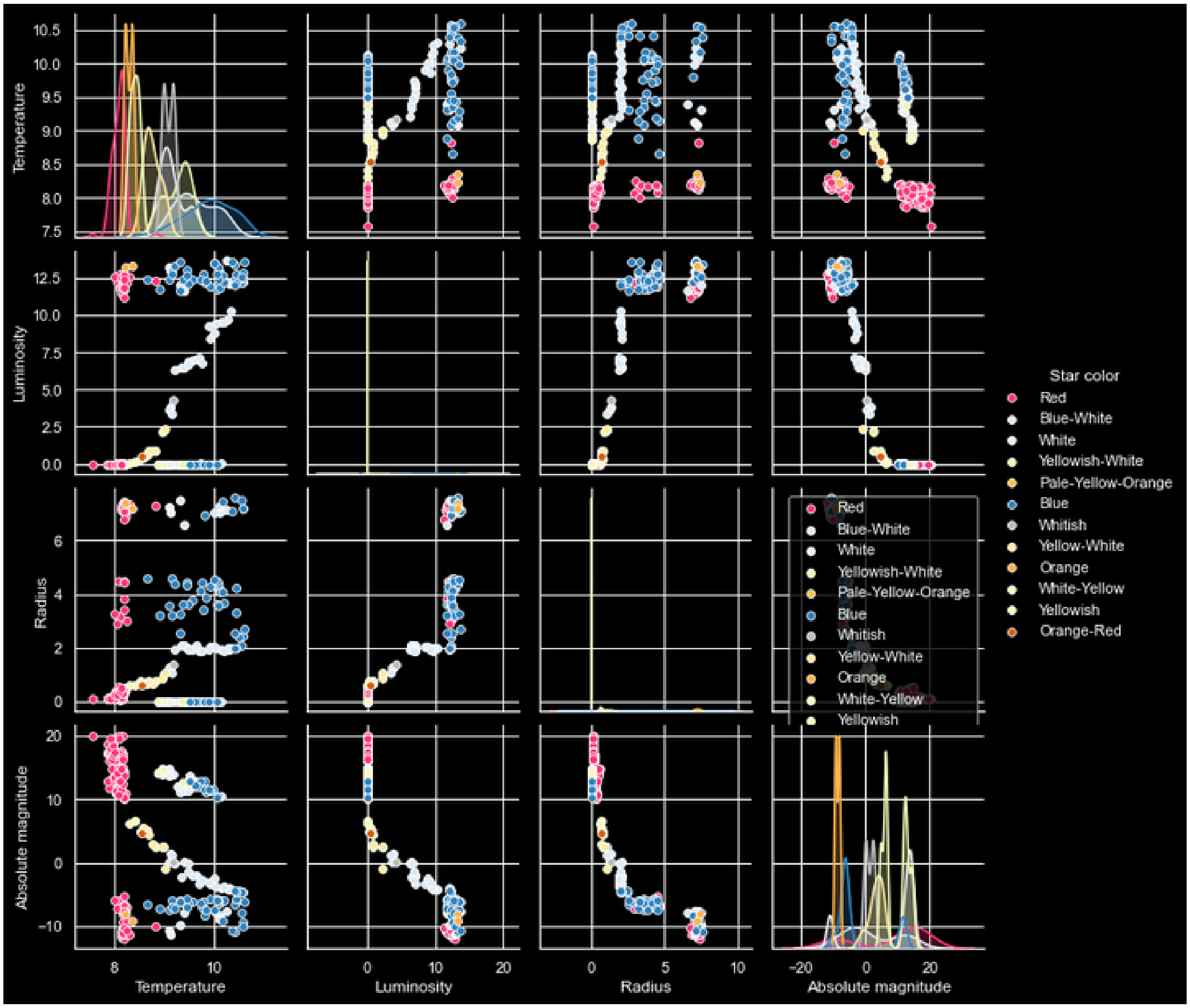
➔ We can already see groups of stars in this chart.

Below is displayed the relationship between the "Luminosity" and the "Temperature". Stars appear with their color and relative size in the scatter plot.

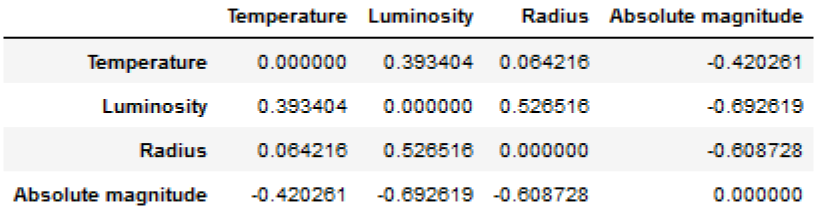


➔ We can already see groups of stars in this chart.

All relationships between numerical features are displayed below:

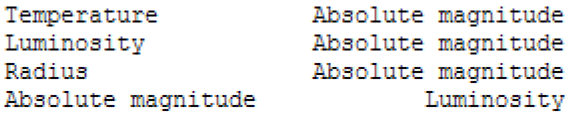


The correlation matrix for numerical data is the following:



	Temperature	Luminosity	Radius	Absolute magnitude
Temperature	0.000000	0.393404	0.064216	-0.420261
Luminosity	0.393404	0.000000	0.526516	-0.692619
Radius	0.064216	0.526516	0.000000	-0.608728
Absolute magnitude	-0.420261	-0.692619	-0.608728	0.000000

The pairwise correlation for numerical data is the following:

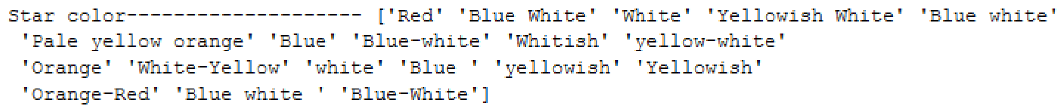


```
Temperature           Absolute magnitude
Luminosity            Absolute magnitude
Radius                Absolute magnitude
Absolute magnitude            Luminosity
```

➔ Absolute magnitude is the most correlated feature

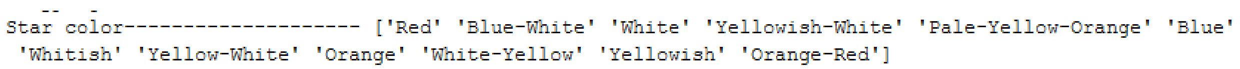
FEATURE ENGINEERING

On the dimension “Star color”, there are a lot of similar color name:



```
Star color-------------------- ['Red' 'Blue White' 'White' 'Yellowish White' 'Blue white'
 'Pale yellow orange' 'Blue' 'Blue-white' 'Whitish' 'yellow-white'
 'Orange' 'White-Yellow' 'white' 'Blue ' 'yellowish' 'Yellowish'
 'Orange-Red' 'Blue white ' 'Blue-White']
```

I reduced the number of categories by grouping similar color: ‘Blue White’, ‘Blue-white, ‘Blue white’ and ‘Blue white ‘ will become 1 category ‘Blue-White’

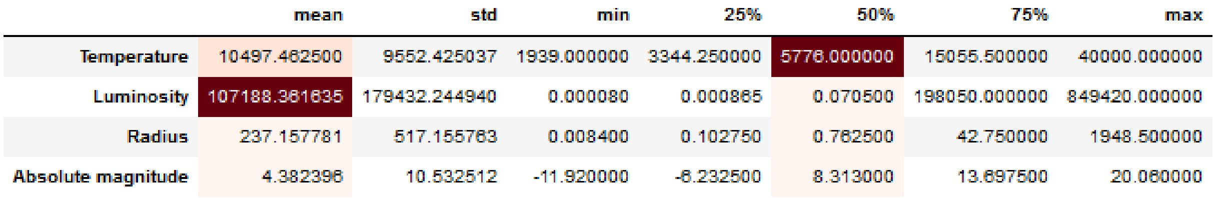


```
   --   -
Star color-------------------- ['Red' 'Blue-White' 'White' 'Yellowish-White' 'Pale-Yellow-Orange' 'Blue'
 'Whitish' 'Yellow-White' 'Orange' 'White-Yellow' 'Yellowish' 'Orange-Red']
```

SCALING

Usually, numerical data of a dataset have different scales.

This is the case here where features **Temperature** and **Luminosity** have a bigger scales than other features:



	mean	std	min	25%	50%	75%	max
Temperature	10497.462500	9552.425037	1939.000000	3344.250000	5776.000000	15055.500000	40000.000000
Luminosity	107188.361635	179432.244940	0.000080	0.000865	0.070500	198050.000000	849420.000000
Radius	237.157781	517.155763	0.008400	0.102750	0.762500	42.750000	1948.500000
Absolute magnitude	4.382396	10.532512	-11.920000	-6.232500	8.313000	13.697500	20.060000

Later, I will evaluate clustering algorithms that require scaled features. So, I will need to proceed to numerical data scaling.

TRAINING UNSUPERVISED MODELS

➔ Summary of training at least three variations of the unsupervised model you selected. For example, you can use different clustering techniques or different hyperparameters.

As a reminder, in this project we will try different clustering techniques to identify stars groups and compare them to the current "Star type" field in the dataset.

For this purpose, 4 clustering algorithms are evaluated:

- K-means
- Hierarchical clustering (WARD)
- DBSCAN
- MeanShift

As discussed previously, numerical data are were scaled.

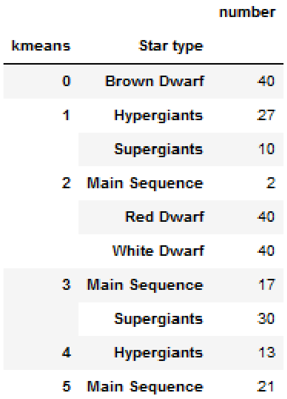
For this project, I will not take into account the following categorical data:

- Color
- Spectral class

K-MEANS

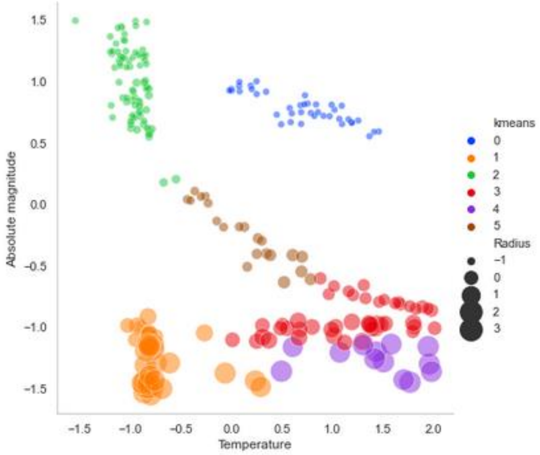
K-means is run with the parameter 6 clusters.

Here is the result with the 6 clusters of K-means compare to the Star type column:



kmeans	Star type	number
0	Brown Dwarf	40
1	Hypergiants	27
	Supergiants	10
2	Main Sequence	2
	Red Dwarf	40
	White Dwarf	40
3	Main Sequence	17
	Supergiants	30
4	Hypergiants	13
5	Main Sequence	21

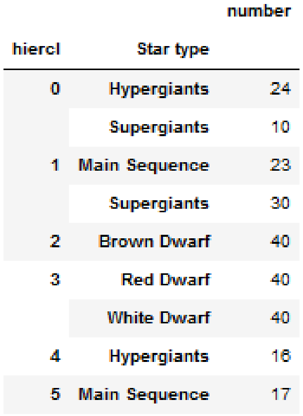
Now, these 6 K-means clusters on the Absolute magnitude distribution by temperature:



HIERARCHICAL CLUSTERING (WARD)

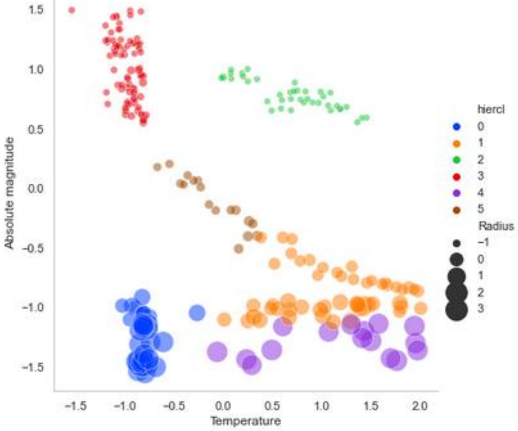
Hierarchical clustering is run with the parameter 6 clusters and the WARD distance.

Here is the result with the 6 clusters of Hierarchical clustering (WARD) compare to the Star type column:



hiercl	Star type	number
0	Hypergiants	24
	Supergiants	10
1	Main Sequence	23
	Supergiants	30
2	Brown Dwarf	40
3	Red Dwarf	40
	White Dwarf	40
4	Hypergiants	16
5	Main Sequence	17

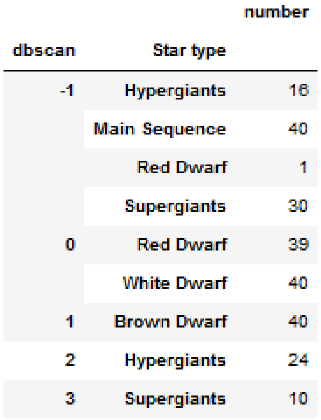
Now, these 6 Hierarchical clustering (WARD) clusters on the Absolute magnitude distribution by temperature:



DBSCAN

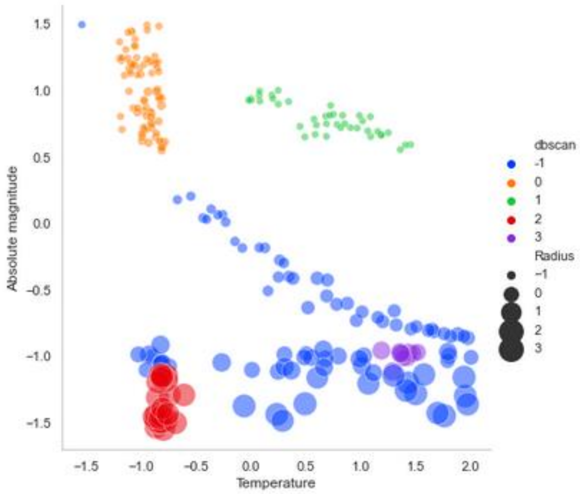
DBSCAN is run. There is no cluster parameter for this algorithm.

DBSCAN found 5 clusters.
Here is the result and the comparison to the Star type column:



		number
dbscan	Star type	
-1	Hypergiants	16
	Main Sequence	40
	Red Dwarf	1
	Supergiants	30
0	Red Dwarf	39
	White Dwarf	40
1	Brown Dwarf	40
2	Hypergiants	24
3	Supergiants	10

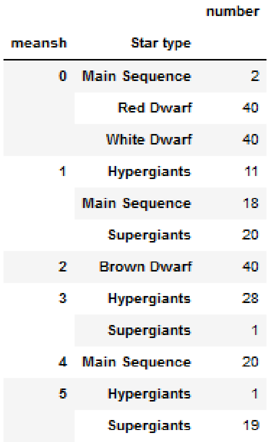
Now, these 5 DBSCAN clusters on the Absolute magnitude distribution by temperature:



MEANSHIFT

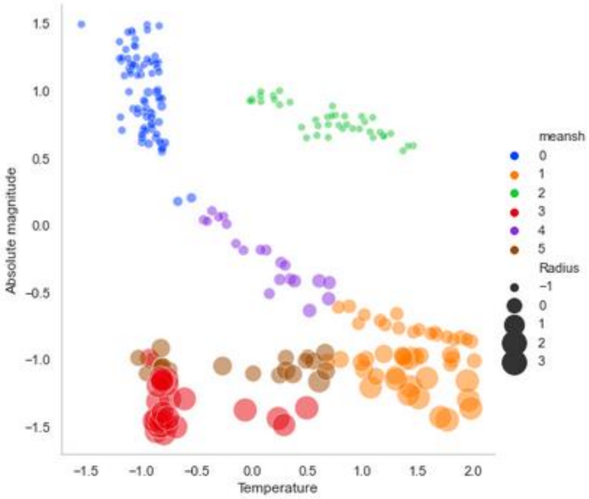
MeanShift is run. There is no cluster parameter for this algorithm.

MeanShift found 6 clusters.
Here is the result and the comparison to the Star type column:



		number
meansh	Star type	
0	Main Sequence	2
	Red Dwarf	40
	White Dwarf	40
1	Hypergiants	11
	Main Sequence	18
	Supergiants	20
2	Brown Dwarf	40
3	Hypergiants	28
	Supergiants	1
4	Main Sequence	20
5	Hypergiants	1
	Supergiants	19

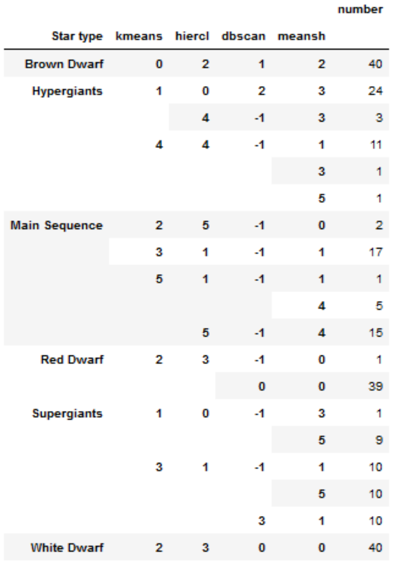
Now, this 6 MeanShift clusters on the Absolute magnitude distribution by temperature:


Absolute magnitude
1.5
1.0
0.5
0.0
-0.5
-1.0
-1.5
Temperature
-1.5
-1.0
-0.5
0.0
0.5
1.0
1.5
2.0
meansh
0
1
2
3
4
5
Radius
-1
0
1
2
3

UNSUPERVISED MODEL RECOMMENDATION

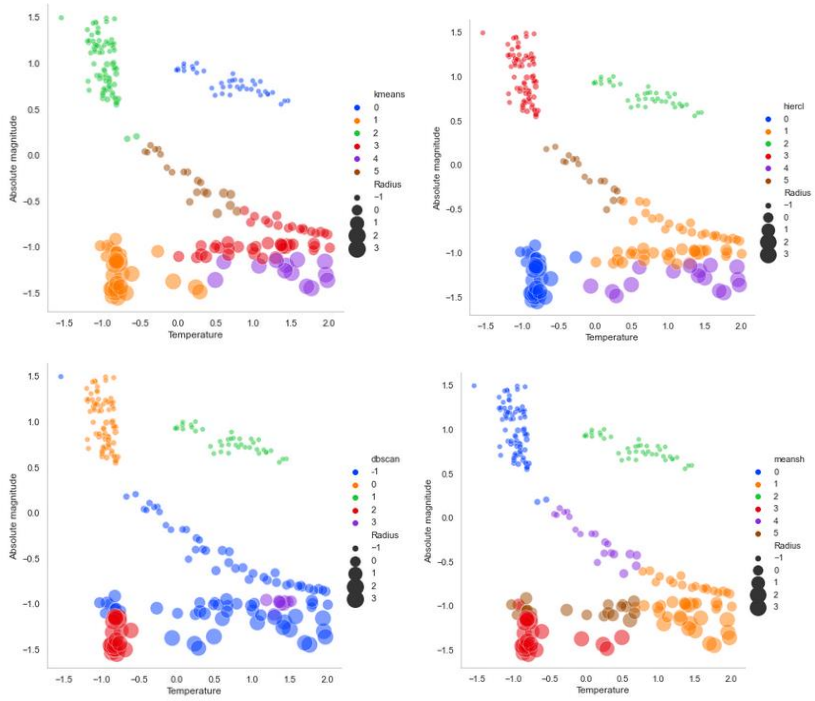
- ➔ A paragraph explaining which of your Unsupervised Learning models you recommend as a final model that best fits your needs in terms.

Here are the results for all of the 4 algorithms:

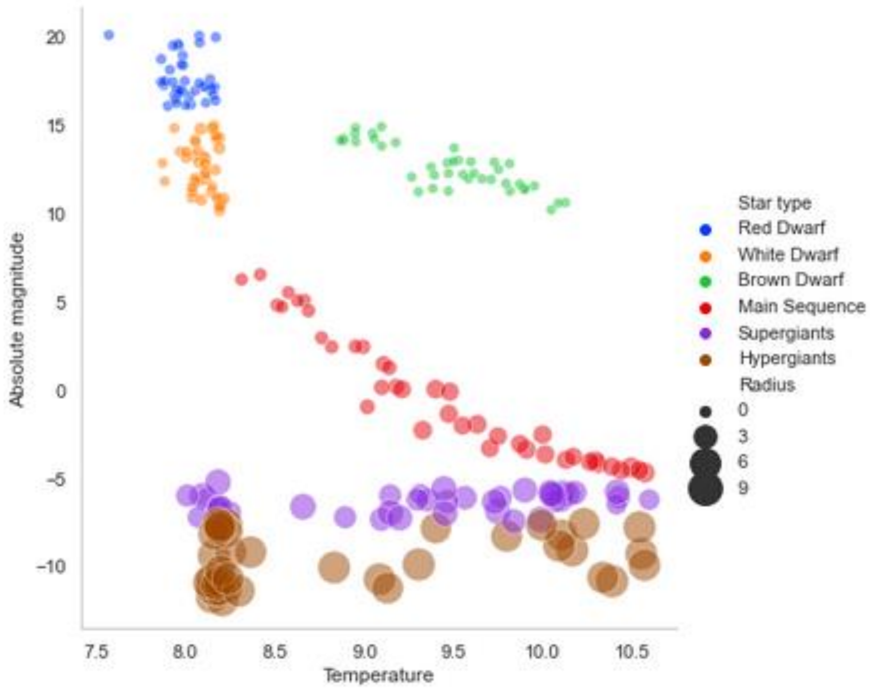


					number
Star type	kmeans	hiercl	dbscan	meansh	
Brown Dwarf	0	2	1	2	40
Hypergiants	1	0	2	3	24
		4	-1	3	3
	4	4	-1	1	11
				3	1
				5	1
Main Sequence	2	5	-1	0	2
	3	1	-1	1	17
	5	1	-1	1	1
				4	5
		5	-1	4	15
Red Dwarf	2	3	-1	0	1
			0	0	39
Supergiants	1	0	-1	3	1
				5	9
	3	1	-1	1	10
				5	10
			3	1	10
White Dwarf	2	3	0	0	40

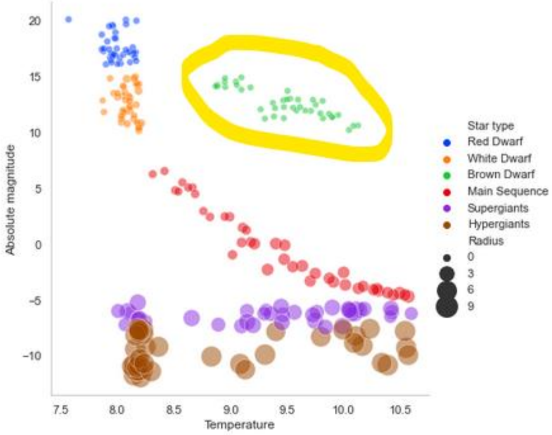
Here are the Absolute magnitude by temperature with clusters in color for each algorithm…



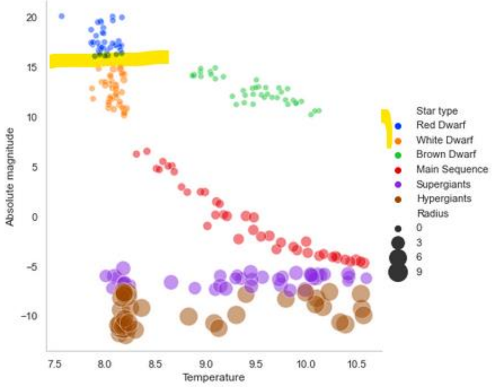
Vs the same information with the star type in color:



All algorithms created a single cluster for “Brown dwarf”, on the top right side of the chart (in green in the previous chart).



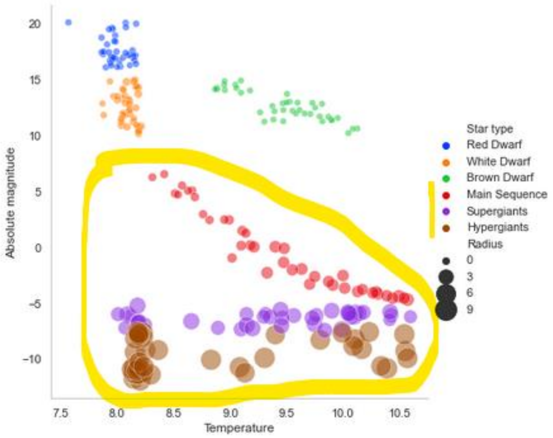
They all failed at differentiating Red dwarf and White dwarf (respectively blue and orange in the previous chart).



DBSCAN failed as it found only 5 clusters instead of 6. Actually, DBSCAN found almost the same number of clusters. But, this algorithm needs to be fine-tuned to get the most of it. DBSCAN is eliminated.

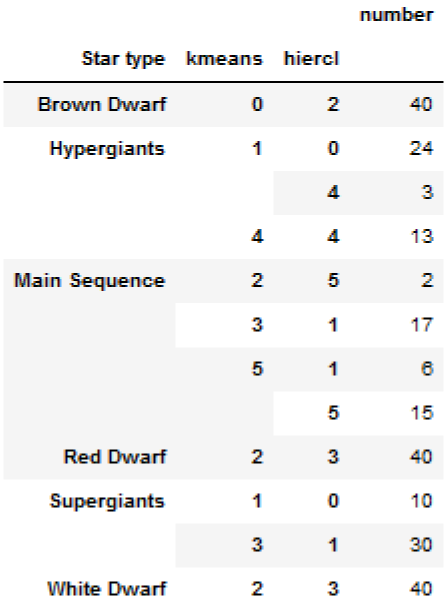
So, the winner will be the one to find the best cluster for:

- Main sequence
- Supergiants
- Hypergiants

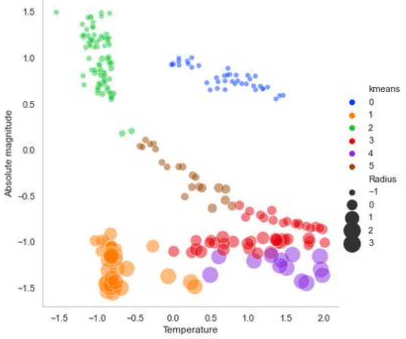


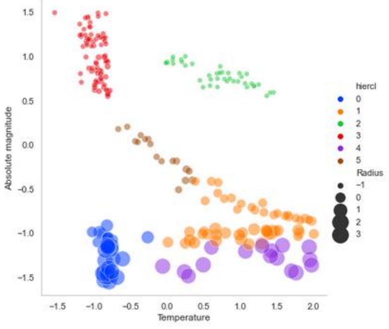
MeanShift mixed the 3 groups and is eliminated.

So, there is now left K-means and Hierarchical clustering (WARD):



			number
Star type	kmeans	hiercl	
Brown Dwarf	0	2	40
Hypergiants	1	0	24
		4	3
	4	4	13
Main Sequence	2	5	2
	3	1	17
	5	1	6
		5	15
Red Dwarf	2	3	40
Supergiants	1	0	10
	3	1	30
White Dwarf	2	3	40





- ➔ Hierachical clustering (WARD) was a bit better on Hypergiants and is then the best model

KEY FINDINGS AND INSIGHTS

- ➔ Summary Key Findings and Insights, which walks your reader through the main findings of your modeling exercise.

Clustering methods allow grouping similar occurrences in cluster. They did a pretty well job on star clustering.

Here is the number of cluster was known (6), K-means and Hierarchical clustering (WARD) were found better but I do think that DBSCAN and MeanShift could be fine-tuned to get better results.

SUGGESTIONS FOR NEXT STEPS IN ANALYZING THIS DATA

- ➔ Suggestions for next steps in analyzing this data, which may include suggesting revisiting this model or adding specific data features to achieve a better model.

The categorical data “Star color” and “Spectral class” were not used for this clustering analysis. They could be added with specific management for this categorical data in a clustering model.

It could be good to get additional data as the dataset is quite small, only 240 stars.

APPENDIX: CODE

The code for this project is available on GITHUB:

https://github.com/Olivier-FONTAINE/IBM-Machine-Learning-professional-certificate/blob/main/04-UNS-Stars%20clustering.ipynb