

Part 1 - Why is the classification of stars important?

A fundamental part of modern astrophysics is studying stars and how they evolve. It is a scientific investigation that has many scientists and researchers all over the world. Once a star is born and as it evolves slowly, it can be grouped into several categories. The grouping would depend on scientific measurements and observations that describe the stars or attributes of the stars. So, instead of randomly grouping stars, scientists have an organized method of grouping stars based on information like temperature, brightness, mass, and size. Every star has its unique physical traits, but they are not much different from each other. For instance, scientists have not been able to observe any green-colored stars. Most stars are red, orange, yellow, green, white, and blue [1]. So, based on these attributes, it is possible to group stars into categories. However, what are the benefits of such classification? There are many reasons; one of them is that it helps us grasp the bigger picture of the universe and its intricate workings. As we explore how stars are classified and how they change over time, we uncover more mysteries of the universe, both in the past and what lies ahead. For example, by studying red giants and supernovae, researchers can find hints about the final phases of a star's life and the processes that create elements heavier than iron, distributing them across the cosmos [3]. This understanding is essential for figuring out the chemical composition of galaxies, including our own Milky Way Galaxy. Ultimately, it sheds light on the origins of life here on Earth. Moreover, exploring stars and how they are categorized impacts exoplanet research. Understanding the traits of various star types can help us make more accurate predictions about the conditions on planets that orbit those stars, like whether they could potentially harbour life. Another example of why the classification of stars is important is because it helps us understand how stars evolve. The behaviour of supernovae can guide our plans for long-term space exploration and even help us assess whether other planets could be habitable in the

future [4]. It is about uncovering cosmic mysteries. In summary, understanding and using star characteristics to classify different star groups is crucial for us.

Part 2 - Characteristics of Dataset and Stars

In this assignment, our main objective is to classify stars into various categories based on observational characteristics. In order to classify stars, we must use a dataset with essential attributes that set each star apart from the others. The dataset we chose for this assignment can be found in Kaggle [2]. Using this dataset and the information provided in the dataset, we have decided to take a structured approach to classify the dataset using data mining topics learned in CPS844. The dataset contains exactly seven attributes: Temperature (K), Luminosity(L/Lo), Radius(R/Ro), Absolute Magnitude (Mv), Star Color, and Star Type. In total, it contains 240 instances. The dataset also did not need any pre-processing as it did not contain any missing values, noisy data, or other inconsistencies.

Temperature plays a vital role in classifying stars. It's measured in Kelvin. Hotter stars typically fall into classes O or B because they emit blue or ultraviolet light. Cooler stars, on the other hand, belong to classes like K or M because of the cooler red light emissions [5]. This attribute encompasses a broad range of temperatures, covering everything from the hottest to the coolest stars found throughout the universe.

Another crucial attribute in our dataset is the luminosity, expressed as a ratio relative to the Sun's luminosity or L/Lo. Luminosity indicates how bright a star is compared to the Sun. It gives us insights into the star's energy output and size, closely linked to its classification. This attribute helps distinguish between stars such as dwarfs, giants, and supergiants.

The radius, also compared to the Sun's radius or R/R_o , is another essential feature in this dataset. It provides us with information about the star's size. The radius of a star plays a vital role in determining star type because stars of each type show similar radii, which helps classify them.

Another vital attribute in this dataset is the absolute magnitude of a star. It measures the star's intrinsic brightness. It is the true brightness of a star as if they were all placed at a uniform distance. This attribute can also be helpful in grouping or distinguishing stars from each other based on true brightness.

The star color is also another important attribute in the dataset. It is mainly an observational characteristic. The color of the star depends primarily on the temperature of the star. Hotter stars usually appear bluer, and cooler stars appear redder. This is a valuable characteristic of a star in classification.

Finally, the last important attribute for classifying stars is the spectral class. It distinguishes stars based on their spectral characteristics. The spectral characteristics of a star are mainly related to the star's temperature. O is the spectral class for the hottest and bluest stars, and M is for the coolest and red dignitaries [5]. The range from O-M can help classify stars.

The dataset also includes the Star type column, not an attribute used to classify stars. Instead, it is the target class or the class we will predict using different classification models and methods. Combining all these attributes and characteristics will help us classify stars into different star types.

Part 3 - Approaches Used and Important Attributes

In order to classify stars into different star types, we used five different classification methods that we have learned in CPS844. The classification methods that we used are the following: Decision Tree Classifier, K-nearest neighbor, Random Forest Classifier, Bagging Classifier, and Naive Bayes Classifier; each of these classifiers has its way of classifying the stars

into different star types using the attributes given in the dataset. Before we began our classification implementations, we had to do some data pre-processing. The attributes star color and spectral class contain data in string type. However, that format has a problem as the classifiers cannot use strings to compare the different instances. We decided to convert the string data into numerical ones so that the different classifiers could use them for the calculations. We also allocated 30% of the dataset for the test set and the remaining 70% for the training set. All of these are done randomly. We also used a random state variable 42 to ensure that random splitting is reproducible.

The first classification method we used is the 'Decision Tree classifier.' The model is a decision tree where each node represents a feature, and each branch represents a decision rule. In implementing the decision tree, we used the Scikit learn library, which is commonly used in this class and labs. We have decided to use 'Entropy' as our criterion for splitting because our goal is to maximize information gain. We also set our tree depth to a maximum depth of 3. There was a significant jump in training accuracy score after changing the tree depth to 3 from 2. Also, having a maximum depth of 3 gives us an accuracy score of 100%. Thus, we can conclude that our model is not overfitted or underfitted. We also used the `feature_importances_` function to print out the importance of each attribute for this mode. The results showed that the key features were absolute magnitude and radius.

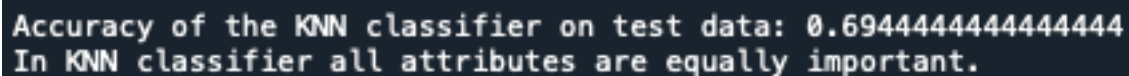
Following is a screenshot of the output for the Decision Tree Classifier:

```
Accuracy of the Tree classifier on testData: 1.0
Temperature (K): 0.0000
Luminosity(L/Lo): 0.0000
Radius(R/Ro): 0.1821
Absolute magnitude(Mv): 0.8179
star_color_encoded: 0.0000
spectral_class_encoded: 0.0000
```

The second classification method we used is 'K-nearest neighbor' or 'KNN' as abbreviated. This is a type of classification that uses the majority vote of the nearest neighbors to classify

instances. In our implementation, we set the value of the nearest neighbor or k to 3. We tried k values lower than three and got a drastically lower accuracy score, meaning the model was underfitted. We also tried k values greater than 3, but the model became too complex because there was very little increase in the accuracy score, which would be overfitting. The accuracy score that KNN yielded was 69.4%. It is also important to note that KNN classifiers have no ranking for which attributes are more important. In KNN classification, all attributes are given equal importance.

Following is a screenshot of the output for the KNN Classifier:

A screenshot of a terminal window with a dark background and light green text. The text displays the accuracy of the KNN classifier on test data as 0.6944444444444444 and states that in KNN classification, all attributes are equally important.

```
Accuracy of the KNN classifier on test data: 0.6944444444444444  
In KNN classifier all attributes are equally important.
```

The third classification method we used in our assignment is the 'Random Forest Classifier.' This method uses multiple decision trees to make its predictions. We have chosen the number of trees or, in other words, `n_estimators` 2. There are several reasons why we chose `n_estimators` as 2. We got quite a low accuracy score when we tried inputting values lower than 2 for `n_estimators`. As we kept adding more trees after 2, we kept getting higher scores, but the increase in accuracy score was very low. At one point, around ten estimators, we got an accuracy score of 100%. However, we kept the value to 2 because we think that was a cause of overfitting since the increase in accuracy score was very low. Thus, the accuracy score of the Random forest classifier for our given dataset was 95.83%. Also, using the `feature_importances_` method, we could identify the attributes used by the classifier. Almost all of the attributes were used except for the Spectral class.

Following is a screenshot of the output for Random Forest Classifier:

```
Accuracy of the Random Forest classifier on testData: 0.9583333333333334
Temperature (K): 0.2288
Luminosity(L/Lo): 0.0108
Radius(R/Ro): 0.4284
Absolute magnitude(Mv): 0.2746
star_color_encoded: 0.0574
spectral_class_encoded: 0.0000
```

The fourth classifier that we used in our assignment was the ‘Bagging Classifier.’ This type of classifier classifies data using multiple decision trees through bootstrap aggregating, a.k.a. bagging. For this method, we used a decision tree as a base tree with a maximum depth of 2. Even using a decision tree with less depth than the decision tree classifier, the bagging classifier got a perfect accuracy score of 100%. Again, we used the `feature_importances_` method to get all the essential attributes used by this classifier. The bagging classifier used all attributes except star color to predict the target class.

Following is a screenshot of the output for the Bagging Classifier:

```
Accuracy of the bagging classifier on test data: 1.0
Temperature (K): 0.0451
Luminosity(L/Lo): 0.1011
Radius(R/Ro): 0.5127
Absolute magnitude(Mv): 0.2911
star_color_encoded: 0.0000
spectral_class_encoded: 0.0500
```

The fifth and final classifier we used in our assignment was ‘Naive Bayes classifier’. It is a probabilistic classifier that assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature, also known as the naive assumption. The model was able to get an accuracy score of 86.1%. We could not use any built-in methods to find the essential features for this particular classifier. So, we used the variance and mean of each feature because in Naive Bayes, each feature’s importance is typically inferred from the mean and variance of the feature’s

distribution within each class. We found that luminosity had the highest variance, which suggests that it was the critical factor in the Naive Bayes classifier.

Following is a screenshot of the output for Naive Bayes Classifier:

```
Accuracy of the Naive Bayes classifier on test data: 0.8611111111111112
Features sorted by importance (based on variance):
spectral_class_encoded: Variance = 4.5079
star_color_encoded: Variance = 24.7182
Absolute magnitude(Mv): Variance = 106.9741
Radius(R/Ro): Variance = 243316.0675
Temperature (K): Variance = 86759788.1497
Luminosity(L/Lo): Variance = 28077193323.1237
```

Part 4 - Explanation of results, comparison, and conclusion

In our assignment, we have used the following classifiers to predict the star types of the instances in the dataset: Decision tree classifier, K-nearest neighbor classifier, random forest classifier, bagging classifier, and Naive Bayes classifier. Each method has its way of classification using different attributes. However, based on the accuracy scores, we can rank them from best to worst.

In our code output, we saw that the KNN classifier had an accuracy score of 69.4%, which is significantly low compared to the accuracy score of other classifiers. Even after playing around with the k value for nearest neighbor this is the highest accuracy score we could produce. Also, this method puts equal importance on every attribute that might not be good. The next classifier that got a higher score than KNN is the Naive Bayes classifier. Using a probabilistic approach, it got an accuracy score of 86.1%. That score is still lower compared to the other classifiers. The next classifier with a higher score is the Random Forest classifier. It was able to get a score of 95.83%. However, with a higher number of estimators, it is possible to get a score of 100%. According to the code output accuracy score, there is a tie for the highest score. The decision tree and bagging classifiers could accurately predict all the test data. However, it is essential to note

that the decision tree classifier only used two attributes, unlike Random Forest and Bagging, to correctly predict the star type, which could be a downside. The fewer attributes used, the more simple a model can indeed be. On the other hand, it is also possible that additional attributes could provide meaningful information about classifying without overfitting. Thus, based on the number of attributes used, accuracy score, and simplicity, the Decision-Tree classifier is the best in this assignment.

References:

- [1] Forde, T. C. (2023, October 21). *What Color are Stars? The Astronomer's Guide to a Stellar Rainbow*. Love the Night Sky. <https://lovethenightsky.com/what-color-are-stars/#:~:text=Summary>
- [2] *Star dataset to predict star types*. (n.d.).
Www.kaggle.com. <https://www.kaggle.com/datasets/deepul109/star-dataset>
- [3] Miller, C. (n.d.). *Stellar Life Cycle / Earth Science*.
Courses.lumenlearning.com. <https://courses.lumenlearning.com/suny-earthscience/chapter/stellar-life-cycle/>
- [4] NASA. (2022, April 13). *Stars / What is an Exoplanet?* Exoplanet Exploration: Planets beyond Our Solar System. <https://exoplanets.nasa.gov/what-is-an-exoplanet/stars/>