Data Mining: Project 2

Akash Shanmugam

9/29/24

Dataset link:

https://www.kaggle.com/datasets/waqi786/stars-dataset

Data Introduction:

The dataset for this project is called the “Star Dataset” which in total contains a comprehensive .csv file that contains a list of known stars with the key features: distance, luminosity, radius, temperature, and spectral class categorization. The dataset’s data is all used from public astronomical databases and cross-referenced. Stars have relatively clear categories in terms of their data because of their known lifecycle; this will likely make any sort of model prediction or data analysis pretty straightforward. My goal for this specific data project is to be able to use the data here to create a Spectral Class classification model that can accurately predict the spectral class based on the given parameters of the other columns (excluding Star ID of course).

Pre-Processing the Data:

Given that the data in the csv file provided was hand-crafted from multiple sources and had clear categorical data that wasn't survey based or taken from a study, just like my previous project the pre-processing is simple because the data is all laid out in the table. Learning from my previous projects' pre-processing, I wanted to first check for any null values at all. Since I had found nothing and there were no specific categories in the column that needed to be excluded unlike last time, my pre-processing step was essentially done. The final important addition I made was to sort the whole dataset alphabetically based on the spectral class column because the spectral classes have important distinctions based on the leading letter (It is worth noting that "B" is considered a higher spectral class that "A" as an exception). Below is an image to show what exactly the stellar classes are.

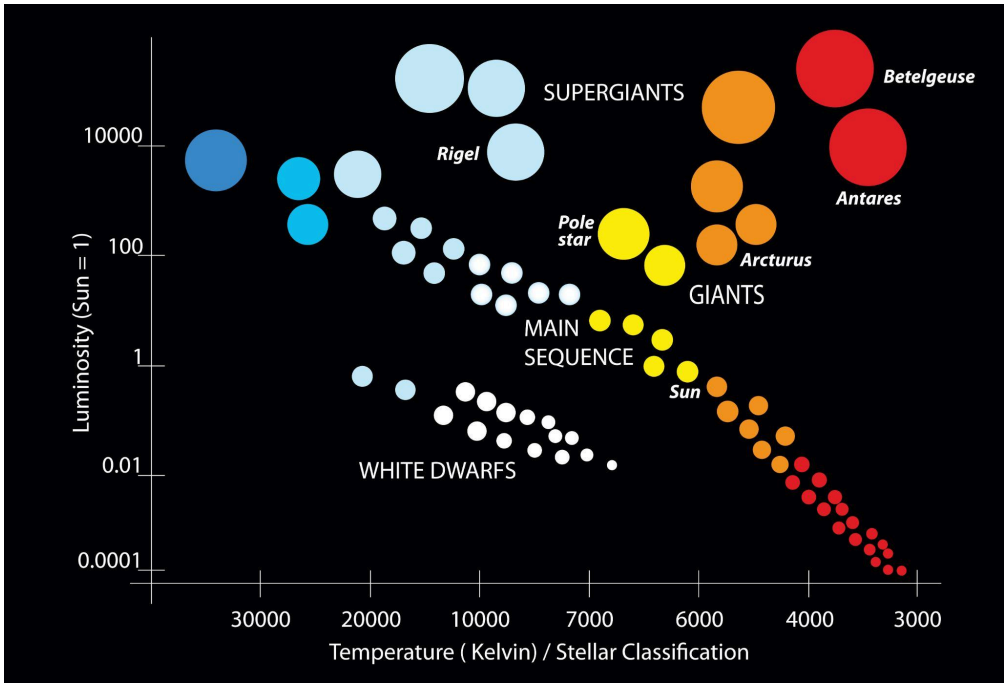
```
df.isna().sum()
df = df.sort_values(by='Spectral Class')
```

Class	Temperature	Conventional color	Apparent color	Mass (solar masses)	Radius (solar radii)	Luminosity (solar luminosity)	Hydrogen lines	% of all Main Sequence Stars
O	30,000–60,000 K	blue	blue	60	15	1,400,000	Weak	~ 0.00003%
B	10,000–30,000 K	blue white	blue white to white	18	7	20,000	Medium	0.13%
A	7,500–10,000 K	white	white	3.1	2.1	80	Strong	0.6%
F	6,000–7,500 K	yellowish white	white	1.7	1.3	6	Medium	3%
G	5,000–6,000 K	yellow	yellow	1.1	1.1	1.2	Weak	8%
K	3,500–5,000 K	orange	yellow orange	0.8	0.9	0.4	Very weak	13%
M	2,000–3,500 K	red	orange red	0.3	0.4	0.04	Very weak	>78%

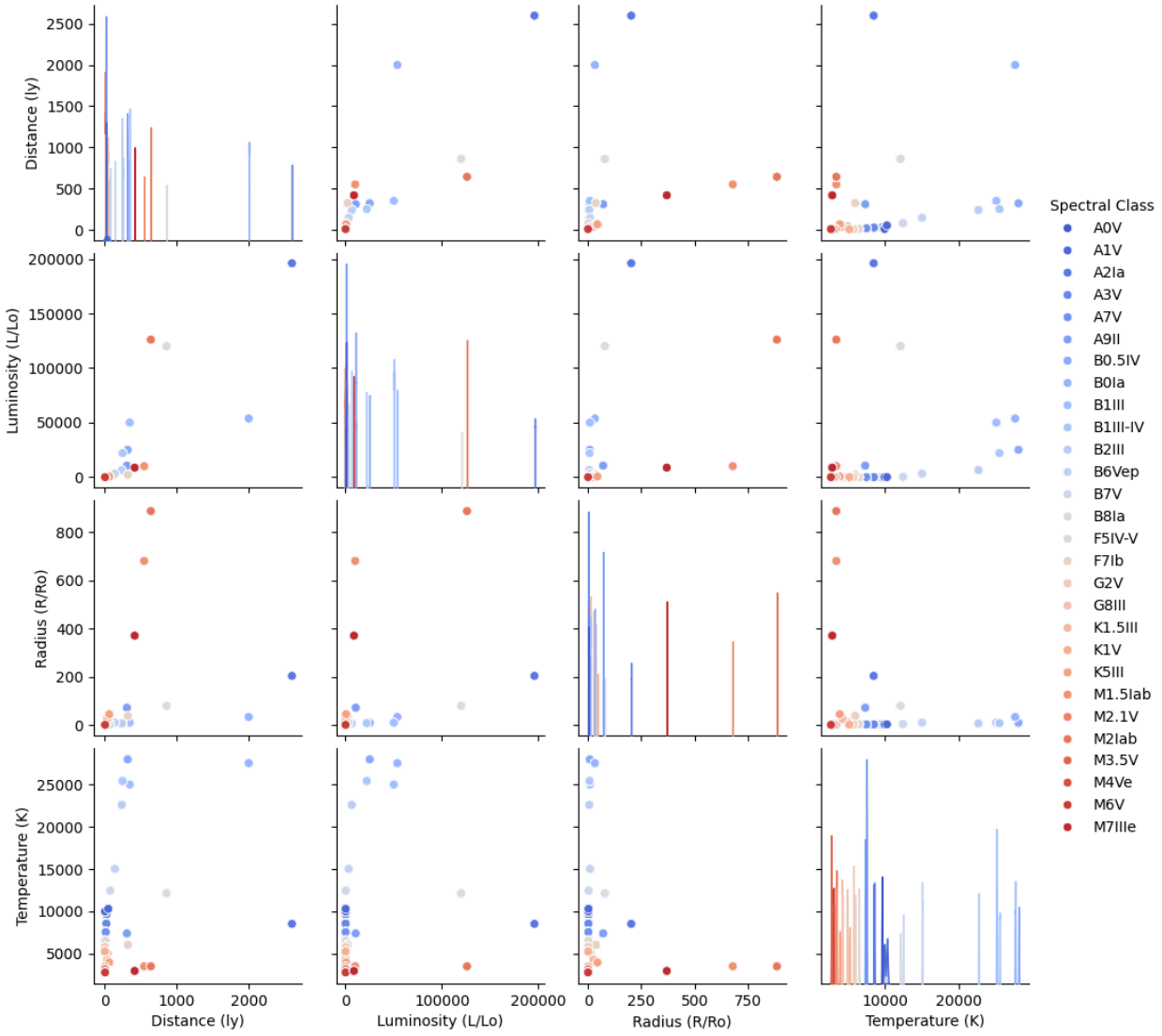
IMG 1

Vis 1: Pairplot of the overall relationship of features

For this dataset, the first thing I wanted to see was what specific relationships all the features had with each other based on how the Stellar classes were spread across the plot. The pairplot is simple and uses the whole database to show how the stellar classes relate to the features such as how many of the data points for distance, luminosity, and radius show that they tend to be closer to the lower left end of the graph due to big outliers in the stellar class data points. Other than that, some of the observations that can be made are that the "higher spectral class" stars trended lower right on the temperature column plots yet in terms of the radius column of plots, the opposite is true (but much less so in terms of volume of data). An image representing the relationship of stellar classes has also been added to show how to visualize the classes themselves.

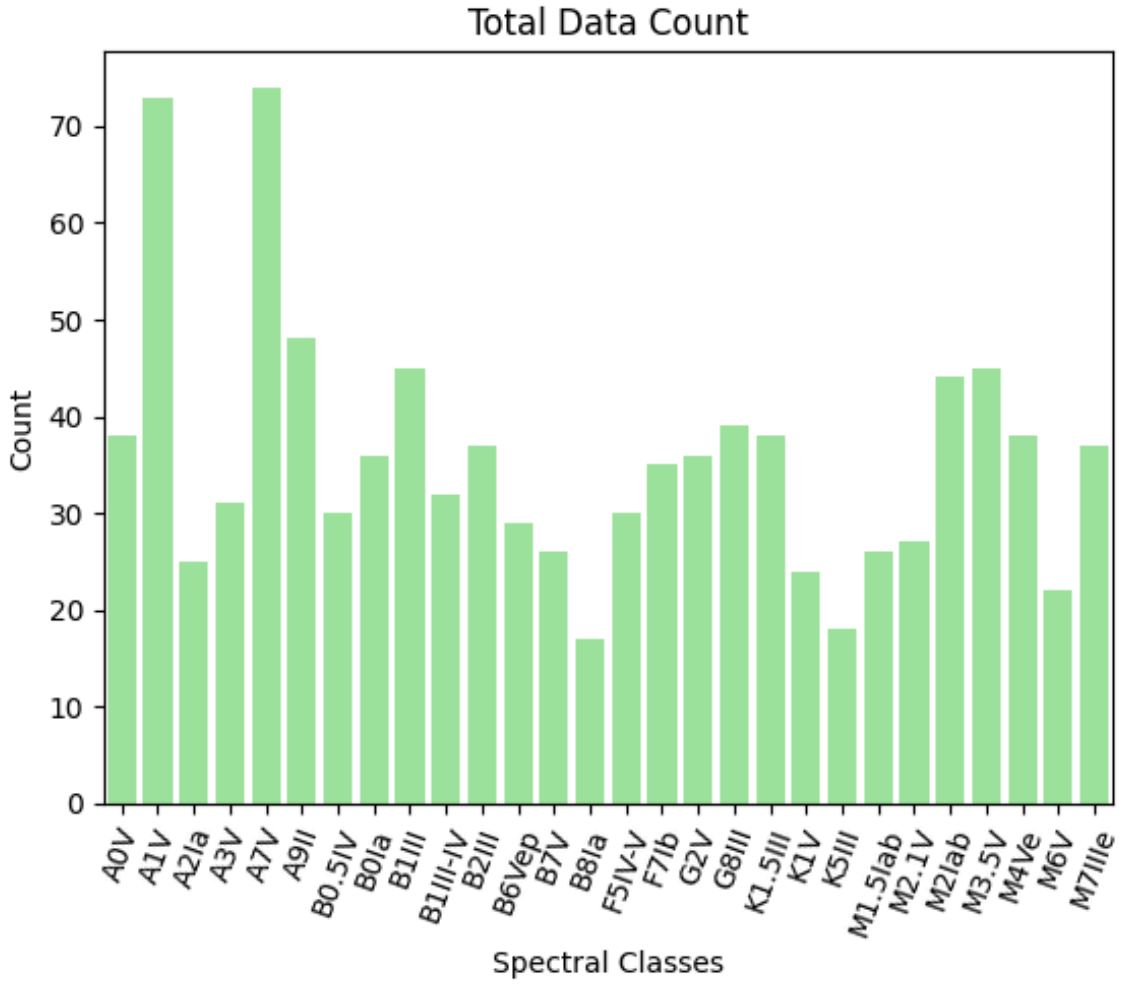


IMG 2


Distance (ly)
Luminosity (L/Lo)
Radius (R/Ro)
Temperature (K)
Spectral Class
A0V
A1V
A2Ia
A3V
A7V
A9II
B0.5IV
B0Ia
B1III
B1III-IV
B2III
B6Vep
B7V
B8Ia
F5IV-V
F7Ib
G2V
G8III
K1.5III
K1V
K5III
M1.5Iab
M2.1V
M2Iab
M3.5V
M4Ve
M6V
M7IIIe

Vis 2: Amount of data in each Stellar Class

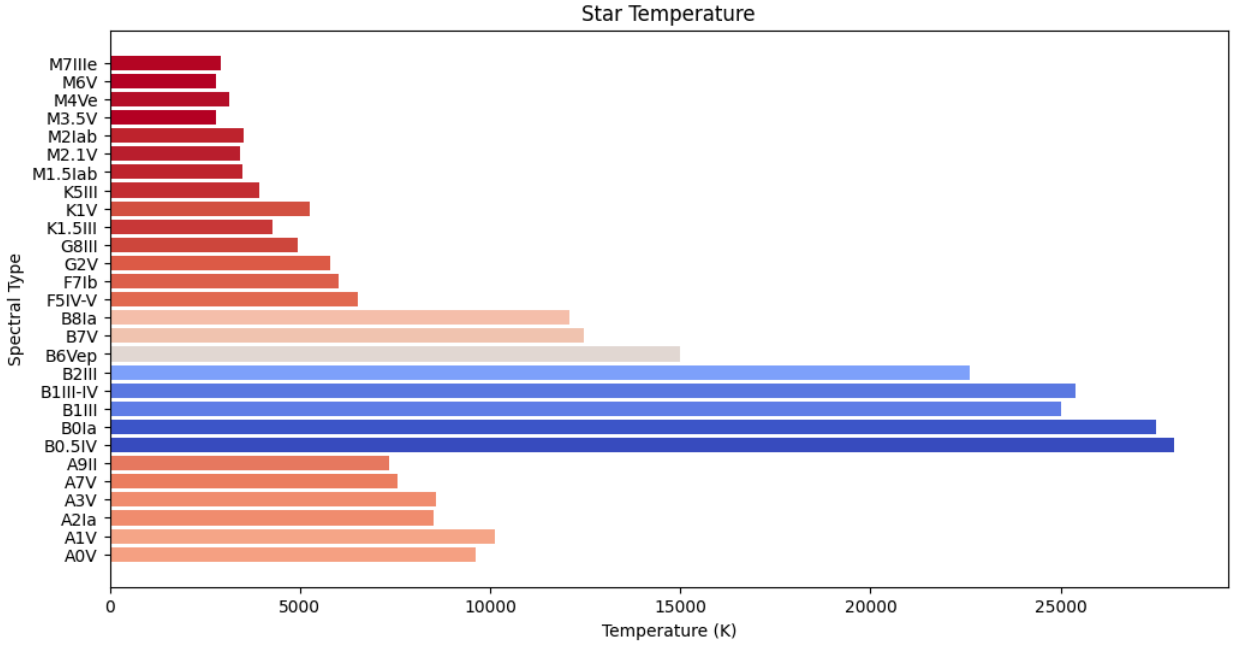
The next thing I needed to visualize for this dataset was how much of the entire star list fit into which stellar classes. Using a simple countplot, the main thing that is noticeable is that the A1V and A7V categories have nearly twice the count of the averages of the other dataset. The stellar bodies in the "A" category are "dwarf stars" and tend to be smaller and hotter. Many of the stars in this category are commonly visible ones like Altai, Vega, and Sirius for example. As a reference the sun is in the "G" category. Other than that the distribution of the data seems to be reasonably varied.

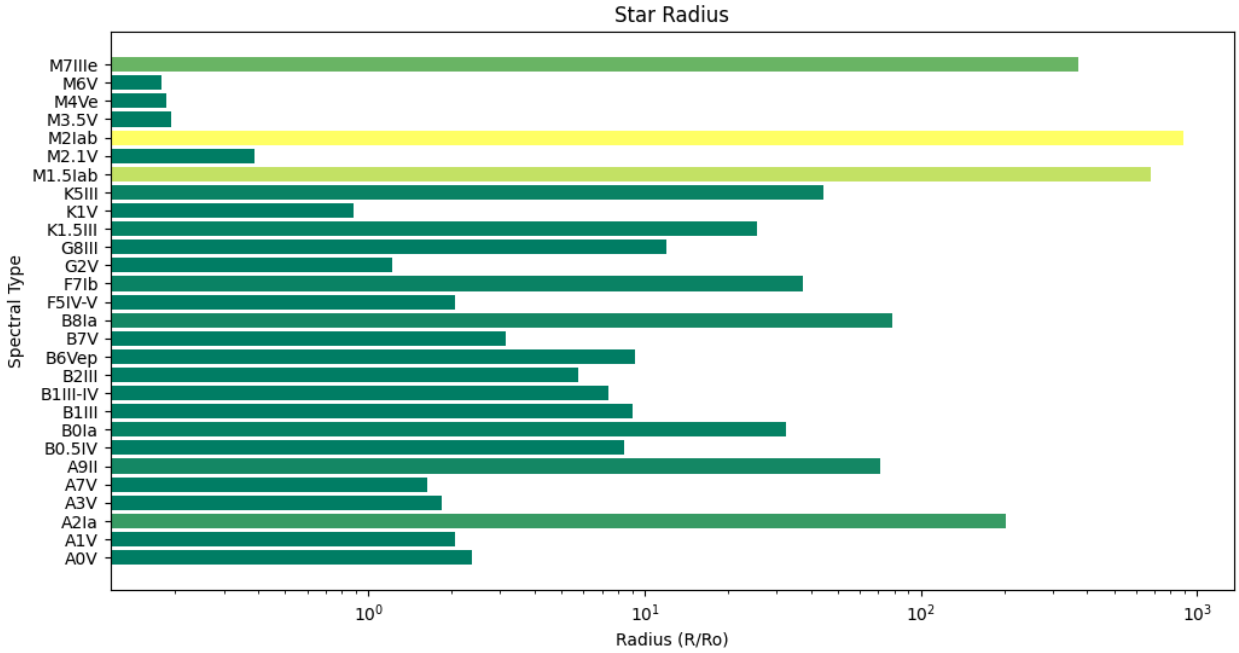


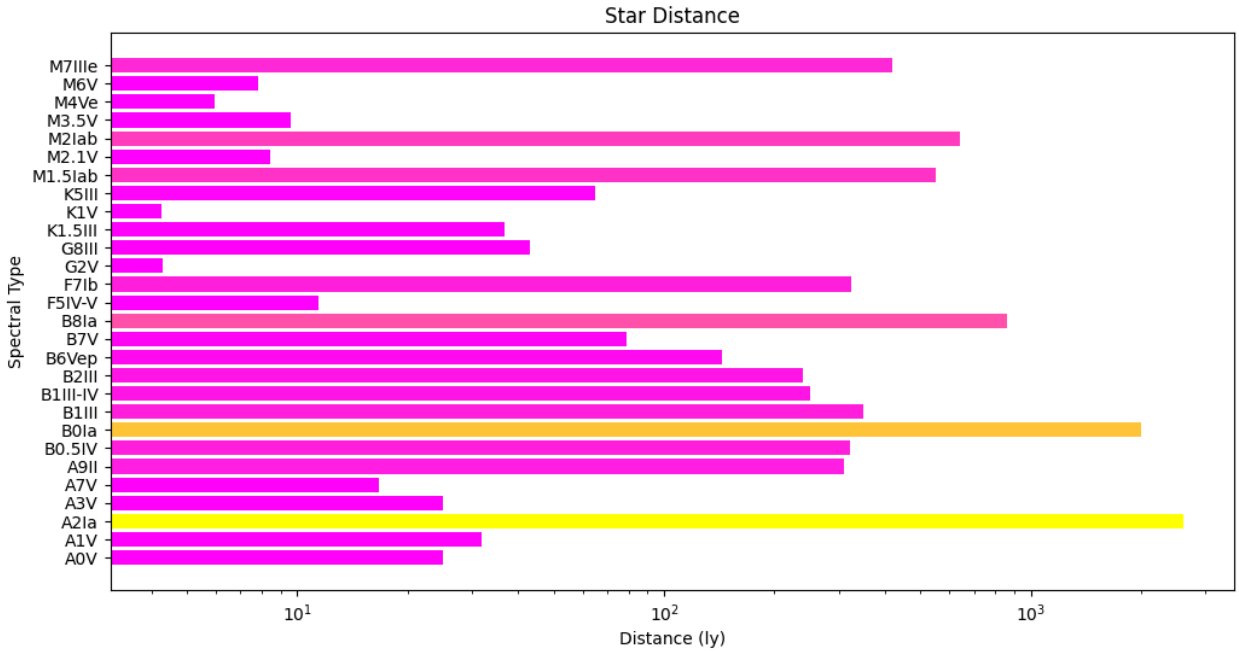
Vis 3: General Move Power by Special and Physical

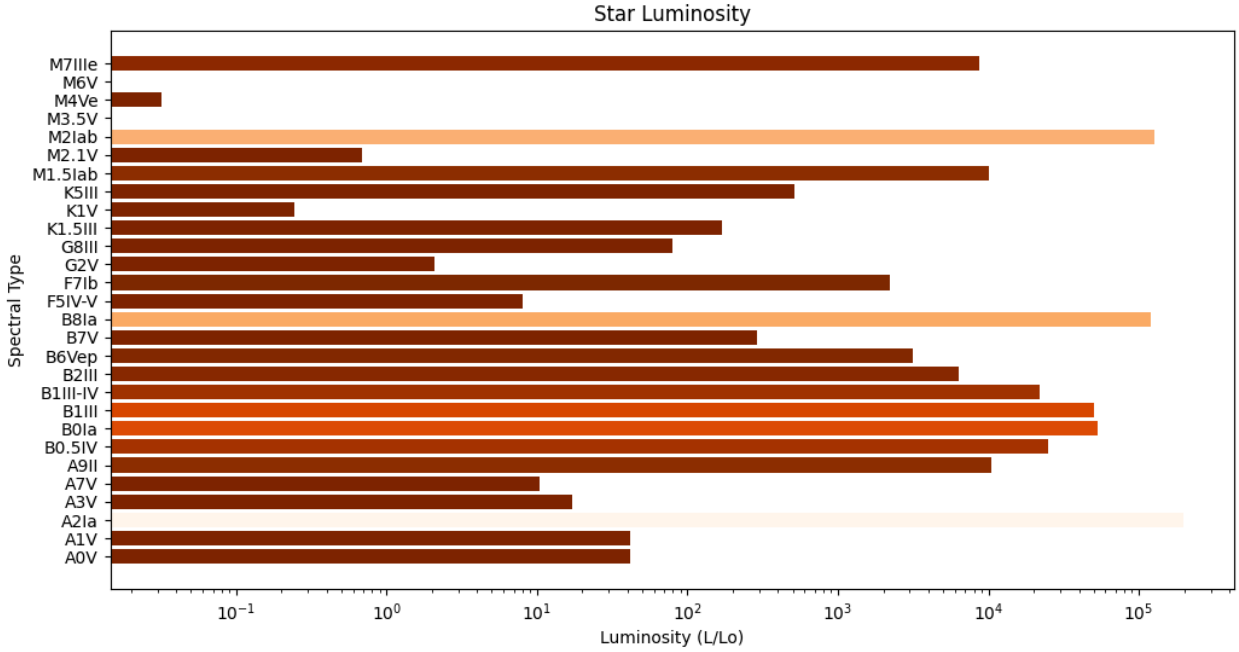
The next trend in data I wanted to show is how stellar classes relationship affected specific features individually, essentially this will be a much clearer representation of the data shown in the pairplot. For my first plot, I wanted to show how the temperature of the stars was directly correlated to the stellar class; something that was the most visible relation in the pairplot already. In this plot you can see that there is a much bigger temperature average in the stars that fit within the “B” category which is the highest stellar category contained in the dataset. After this massive leap in the data, starting from the A class, they all decrease slowly after 10,000K. The next category I wanted to test was the star radius. Looking at the pairplot a good amount of the features other than temperature are very difficult to actually make out at all because all of the data is crammed on the left bar a few outliers. This is a big indicator of these features being logarithmic in nature and those features actually needing to be viewed on that kind of scale. For radius, that was the first thing that needed to be done for the plot. Immediately looking at the plot, the biggest radius categories by a large margin are the ones that end in “la” or “lab”. These refer to “supergiant” class stars and generally the biggest stars around going around up to 10^3 R/Ro, key examples being: Deneb or Rigel. Now that these two categories have clear trends with their stellar class average stats, the next is Distance. The distance plot was somewhat similar to the radius plot with supergiants being generally the closest star category but the data collected could be affected only by what is visible which means that distance data can only be reliable up to certain point as a metric to meaningfully predict a stellar class which leads us on to the final feature: Luminosity. The results for this plot ended up being noticeably similar to the radius plot with the main difference being a big increases in the higher part of the B category. Luminosity here seems to be affected both by the previous three features together with distance and radius

seemingly having a much bigger effect on the result with “A2la” being the best of all features here.






Star Distance
M7IIIe
M6V
M4Ve
M3.5V
M2Iab
M2.1V
M1.5Iab
K5III
K1V
K1.5III
G8III
G2V
F7Ib
F5IV-V
B8Ia
B7V
B6Vep
B2III
B1III-IV
B1III
B0Ia
B0.5IV
A9II
A7V
A3V
A2Ia
A1V
A0V
Spectral Type
10^1
10^2
10^3
Distance (ly)


Star Luminosity
M7IIIe
M6V
M4Ve
M3.5V
M2Iab
M2.1V
M1.5Iab
K5III
K1V
K1.5III
G8III
G2V
F7Ib
F5IV-V
B8Ia
B7V
B6Vep
B2III
B1III-IV
B1III
B0Ia
B0.5IV
A9II
A7V
A3V
A2Ia
A1V
A0V
Spectral Type
10^{-1}
10^0
10^1
10^2
10^3
10^4
10^5
Luminosity (L/Lo)

Model: KNN Stellar Class Prediction

Here the first thing that needed to be determined was which type of classification model to use. Between options like SVM, Random Forest or KNN, I wanted to chose a classification model with lower complexity due to the simple amound of features in the dataset as well as how clearly it is structured and laid out. In the end, I chose a KNN model because of how quickly the result was able to be created and the accuracy in the end. The model ended up being in a range of 97-98% with enough tuning of nearest neighbors and test size while also trying to prevent over/underfitting. I also implements a way to see test points and predictions to better assess how the model was interpreting the data.

```
df = df.drop('Name', axis=1)

X = df.drop('Spectral Class', axis=1)
y = df['Spectral Class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=68)

knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
y_pred = knn.predict(X_test)
#print(f"Test points: {X_test}")
#print(f"Predictions: {y_pred}")

accuracy = accuracy_score(y_test, y_pred)
print(f'KNN Accuracy: {accuracy*100:.2f}%')
```

```
KNN Accuracy: 97.67%
```

Impact:

The project was able to use the dataset taken, understand what features likely matter and which don’t, and then create a classification model that can predict the stellar class of stars using the

raw statistics while seeing how the previous predictions lined up with the actual result. The impact of this project is that stellar objects can be classified relatively accurately even with simple models given how comprehensive yet simple the data for this field is. This can help easily identify newly discovered stars and help people understand why a star is in a certain category and how they fit in the context of where they are in space or how they can help shed light on a previously unknown part of the universe.

References:

https://www.kaggle.com/datasets/waqi786/stars-dataset

IMG 1: https://cosmosfrontier.com/astronomy/stars/stellar-classification/

IMG 2:

https://assets.iflscience.com/assets/articleNo/68575/iImg/67409/shutterstock_1648865182.jpg