Molecular Analysis of Stellar Clusters
Final Paper

James Wu

8/17/22

Abstract

WIth the diversity of research methods and measurement instruments found in astronomy, it is often difficult to draw direct comparisons between conclusions due to differences in both kind and quality of measurements. By using a machine learning algorithm we have been able to predict measurements that may not have been gathered, notably , the iron content of a star based upon temperature or gravity readings. The problem has not been answered before with this data set, and the scientific community does not have a large amount of research that has pertained to this specific APOGEE data release. By addressing this previously unaddressed problem, we can advance the scientific community with a new contribution that allows scientists of the future to gain additional data in conditions where such data may not be so readily available. We have approached the question by using various machine learning algorithms to discover correlations between the data points we were given, creating a predictive model to produce a numerical result. Gravity and temperature do not present a direct linear relationship to the iron content of a star. However, as their data points tend to cluster together, we can utilize the KNeighborsRegressor to find correlations in the way that these points have, and produce a numerical prediction. Throughout testing we've found that the KNeighborsRegressor does better on all accounts in comparison to a large variety of other regression models such as the GaussianProcessRegressor. Considering the results of the Mean Absolute Percentage Error function upon the results of both of these, we can note how the LinearRegressor, mathematically, does marginally better than the KNeighborsRegressor, at a value of 1.13% of average error in comparison to 2.53%.

1. Introduction

Consider the known relationship that can be found between a star's temperature and gravity, where a main-sequence star of higher temperature generally denotes an older, more massive, and higher gravity star (Futurism, 2014). However, we must also notice that a large portion of stars may not exactly be included in the main-sequence, and do not adhere strictly to the exact same correlation that we've found here. This presents that stars do not present a direct mathematical relationship between temperature, gravity, and their iron concentration, though, considering the graph in Figure 2, there still lies a relationship between these values. By using this assumption, we can conclude that a machine learning model can be used upon a set of data points detailing a star's effective temperature (TEFF) value and its gravitational pull (LOGG). We therefore train a machine learning model for predicting a star's iron content (FE_H) given this data. Due to this correlation in the data and the fact that the set is labeled, it is an ideal candidate for supervised learning. As we were also working only with numerical data, where the star's iron content can be expressed as a numeric quantity, this leaves only regression as our option. By providing another tool for scientists to use in the analysis of stars, a machine learning prediction will give scientists more of what to expect when they are given telescopic data.

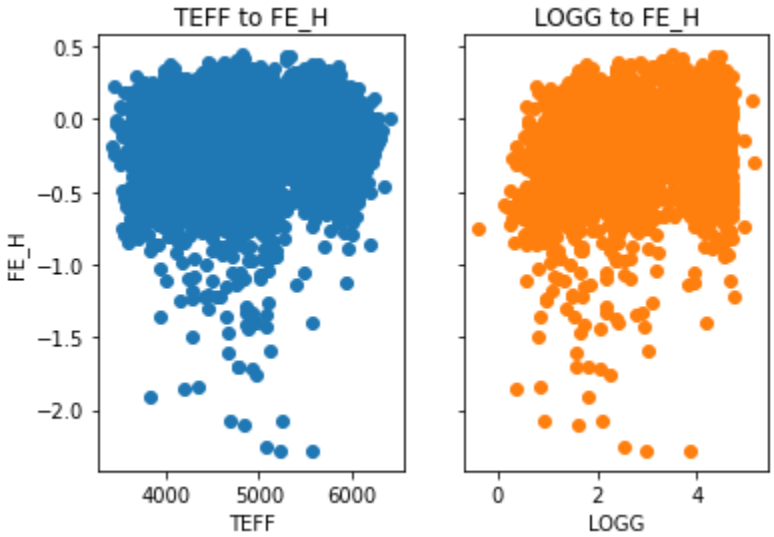
2. Background

Due to the recency of the APOGEE DR17 release (Majewski et al. 2017, AJ, 154, 94), few studies have been conducted on the data. We used the literature regarding previous versions of the dataset as a jumping off point for our analysis. A study conducted by Taylor Spoo et. al. highlights a star's chemical composition by studying the relationship between a star cluster's C/N value to determine its age, which are completely different metrics and goals from ours, though. Considering how many finish

3. Dataset

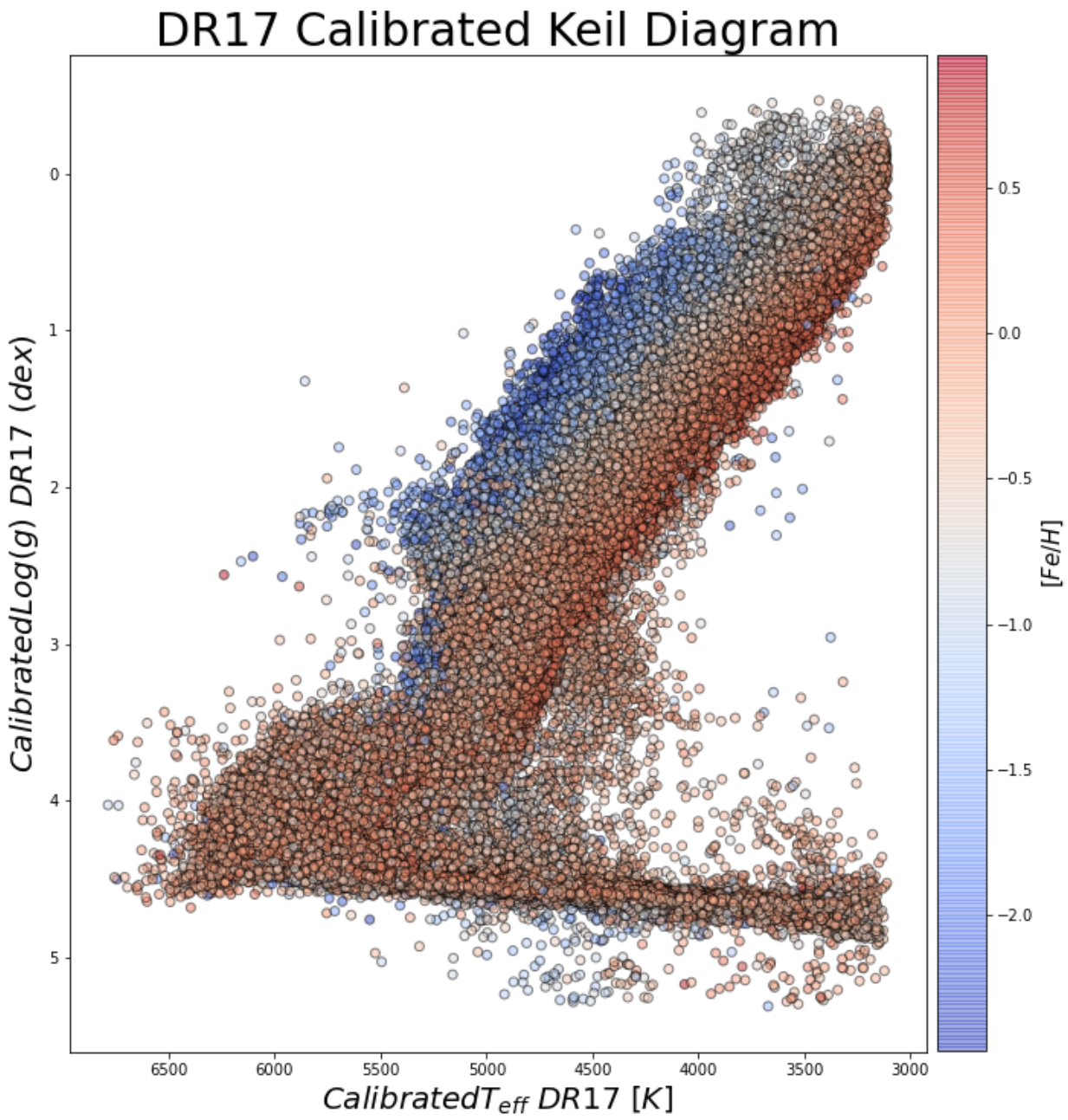
For this project, we've used the latest APOGEE DR17 as our data, which gives us access to a mass of numerical data of which we've narrowed down to only using the LOGG, TEFF, and FE_H columns, rather than dealing with the whole dataset. By using a data preprocessing method that filters out the bad data flagged in the dataset by its APSCAPFLAG bit 23, we keep only the "good" data, that has all of its measurements that are least likely to be affected by external factors that impede the accuracy of the data that we want. In the interest of time, we cut down the number of samples from the dataset down to 100000, to allow iteration on the model in a shorter period. In the visualization of the dataset, we compared its data points to one another, as evident in Figure 1:

(*Figure 1*)



From this, we notice that the iron concentration of a star generally has a similar relationship in comparison with TEFF and LOGG variables, being a large cluster of points, and a trail of outlying data. Considering the distribution of stars in the following Keil diagram given by an official release of APOGEE data, we consider the following graph, and notice its interesting distribution pattern, which can be attributed to the correlation between temperature, gravity, and a star's iron content.

(*Figure 2*)

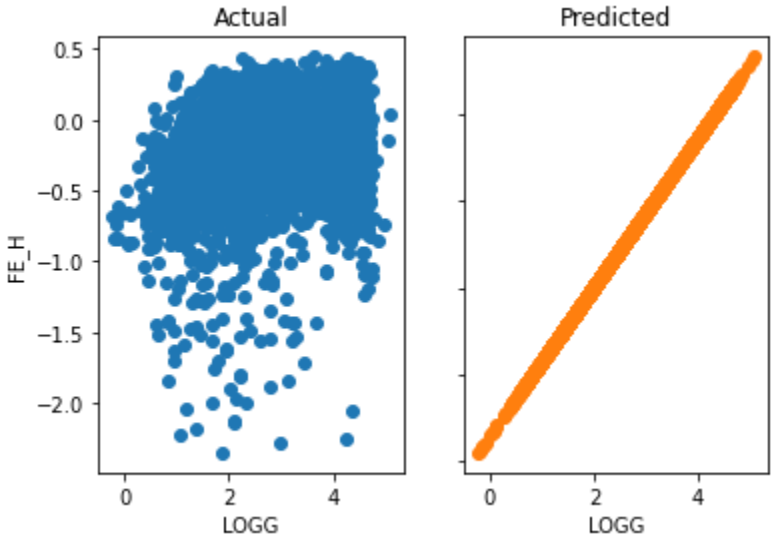


From this graph, we can see that there is a clear gradient distribution in the points, each of which are relatively clustered together, without a clear mathematical correlation. This explains why the GaussianProcessRegressor, according to the Mean Absolute Percentage Error (MAPE) function's calculations, would be poorly equipped to handle the clusters of data points displayed in Figure 1, due to its tendency to look for wave-based distributions of data. Therefore, a model more specialized to find more complex nonlinear relations would be better suited for our problem, such as a KNeighborsRegressor or a neural network.

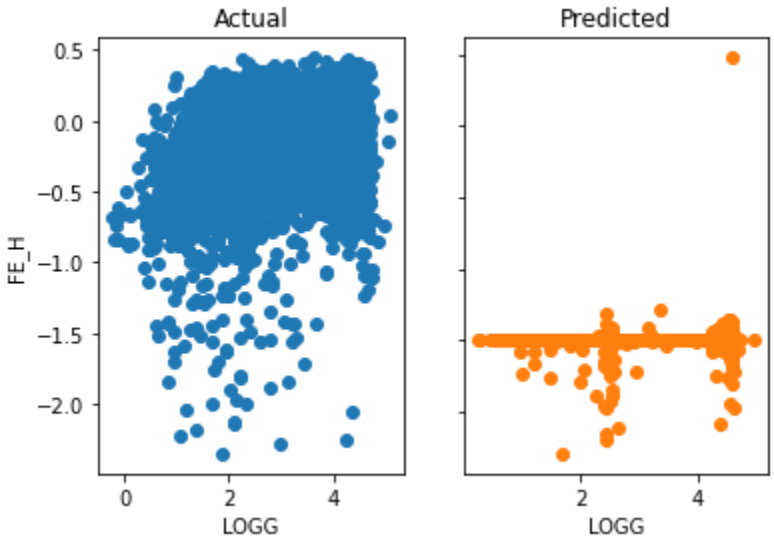
4. Methodology / Models

To solve this problem, we approached it by testing the results of a multitude of regression models upon the data set, to try and find the best model that could be fit to our specific circumstance in the data we were given. Regression models, such as linear regression (Figure 3) and Gaussian Process regression (Figure 4) both were clearly not in line with our expectations due to their incompatibility with our data set. Their MAPE values read along similar lines to what we'd expect of such differences in their graphs, with the Gaussian Process Regressor scoring the worst at 85.7e15% and the Linear Regression algorithm interestingly scoring at 1.13% despite its failure as displayed in the graph below.

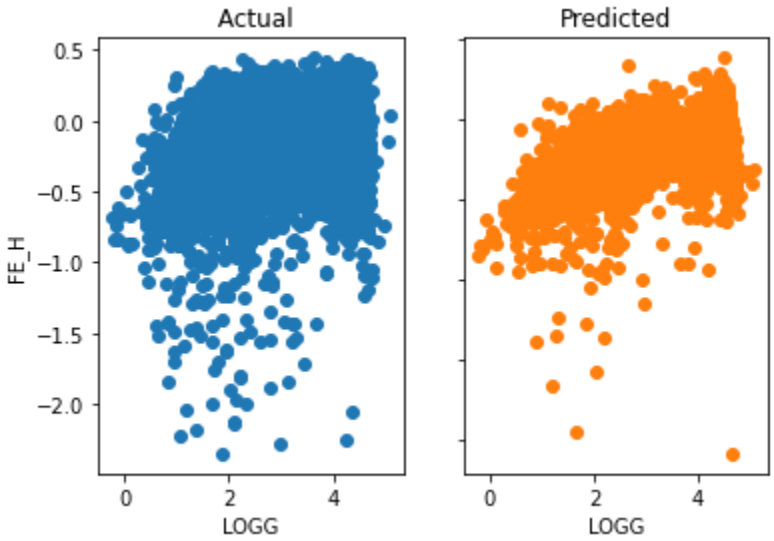
Linear Regression (Figure 3)



Gaussian Process Regression (Figure 4)



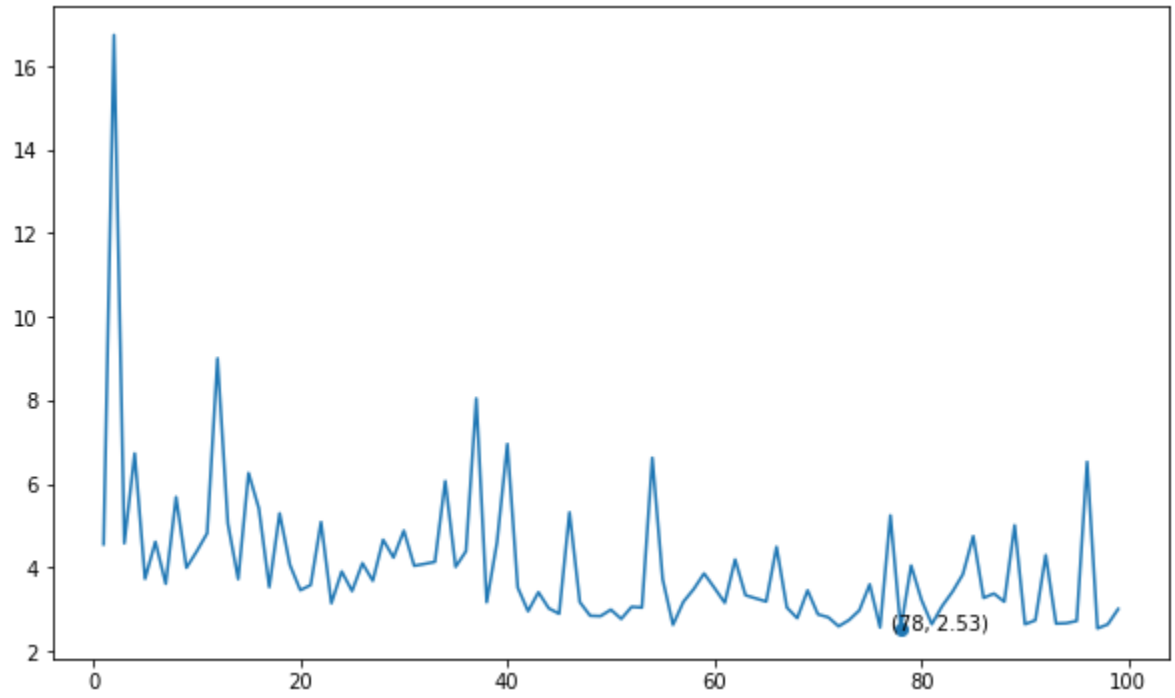
By graphically visualizing our data, we have managed to display that mathematical error values do not encompass the actual difference between the predicted data and the actual data in these cases. Moving forward, since they were more focused upon finding a mathematical relationship between the points, where it isn't actually present in this situation. Following that discovery, we decided to use a machine learning algorithm more optimized to work with data with clustered data relationships, settling on the KNeighborsRegressor. As the KNeighborsRegressor depends on predicting a data point based upon the "nearest" points that it can find around an inputted X-value, we believed that it would be the best option. As it works by attempting to find a sensible point based on its surroundings, the KNeighborsRegressor should be able to find the values we want, based on our data's distribution. With a regression algorithm that focuses on finding nearby values, it visibly exceeds the other two tested models by a significant margin, as displayed in the following graph, alongside presenting a much improved MAPE value of 2.53%.



5. Results and Discussion

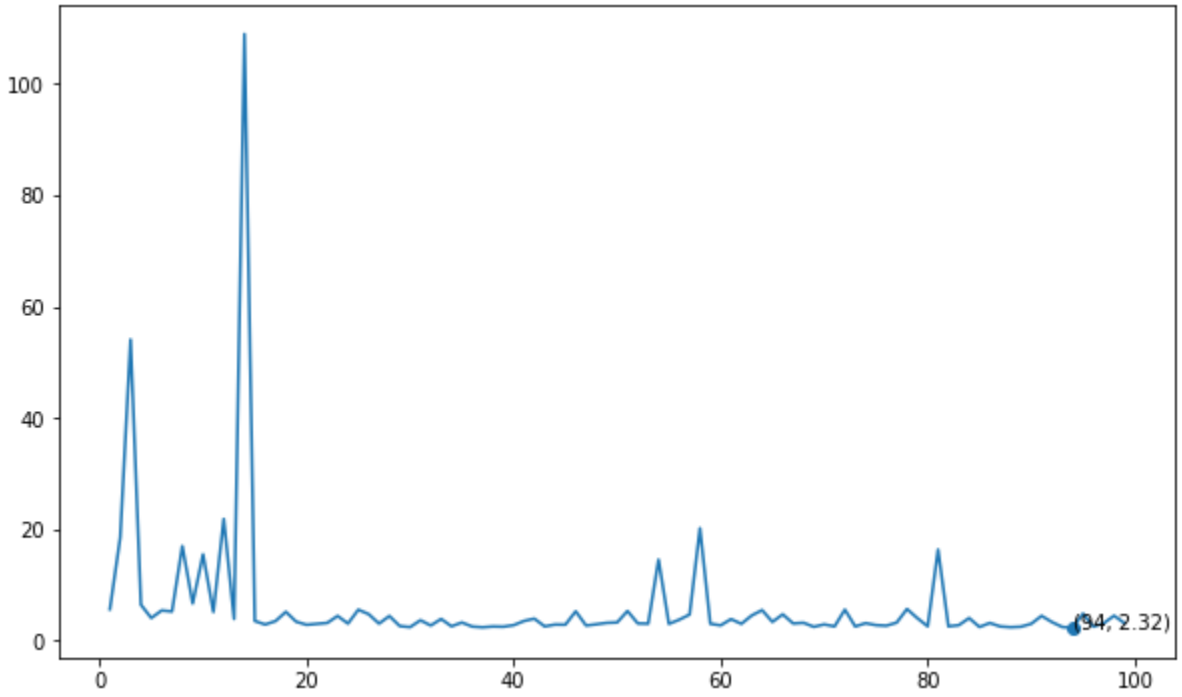
With KNeighborRegressor as our best performing algorithm we began testing various hyperparameters included with the algorithm, to attempt to minimize the margin of error and give us a product better able to predict the inputs we give it. This drove us to produce a way to find the best possible hyperparameters for the function, which gave us the following graph of error values, giving us that the 94 neighbors are the optimal amount we should use on a set comparing LOGG to FE_H.

(Figure 5)



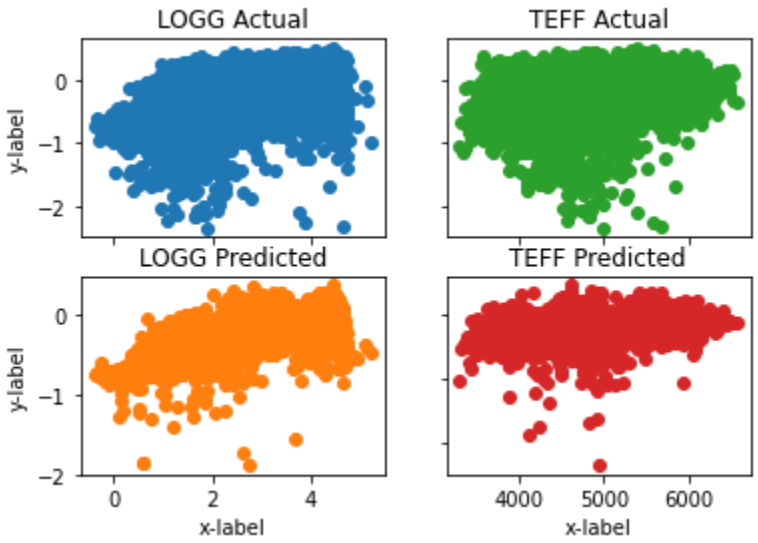
Now that we selected our optimal methods, we moved on to applying a similar process to the same dataset, but using TEFF as our X-axis. Running the same hyperparameter analysis as we did on the LOGG data, we get the following as our graph of error:

(Figure 6)



Following both of these least possible error values gives us the following graphs of predicted vs actual value:

(Figure 7)



The above graphs demonstrate a clear improvement in comparison to those predicted by linear regression and Gaussian Process Regression. Following the creation of these models, we successfully achieved our goal, having created a satisfactory tool to predict the iron concentration of a star based on its gravitational pull or its effective temperature.

6. Conclusions

By testing a multitude of models from the python package scikit-learn upon the APOGEE dataset, we've managed to produce a resultant product that can predict a star's iron concentration depending upon its gravitational pull (LOGG) value or effective temperature (TEFF) value. Using the KNeighborsRegressor from scikit-learn, we have created a reliable algorithm that can help in filling in missing data in surveys. I believe that the model hasn't managed to perform up to its best due to the abstract relationship between the data points we were observing.

This research can be furthered by adapting the model to additional relationships between APOGEE data points, and allowing the machine learning algorithm to predict other data points included in similar data releases. By fitting the algorithm to more data points, then additional research can be conducted. As the model we chose, a KNeighborsRegressor that took in a large amount of nearby points, is better suited for dealing with clusters of data, we were unable to touch upon data points that were outliers from the rest of the data set. That causes us to believe that a model more suited for dealing with more abstract relationships, such as a neural network, may be able to outperform the current model that we used in this research.

Acknowledgments

I give my sincerest thanks to the SDSS foundation and their APOGEE offshoot project, for providing the data that fueled this project. I also give my thanks to Aidan Donaghey and the InspiritAI organization for giving me the opportunity to learn about and work with these data sets to form the research that I've done.

References

Futurism, "What happens when stars produce iron?," Futurism, 14-Jul-2014. [Online]. Available: https://futurism.com/what-happens-when-stars-produce-iron. [Accessed: 17-Aug-2022].

Blanton et al., "Data release 17," SDSS, Jul-2017. [Online]. Available: https://www.sdss.org/dr17/. [Accessed: 17-Aug-2022].

Taylor Spoo et al, "The open cluster chemical abundances and mapping survey. vii. apogee ...," The Open Cluster Chemical Abundances and Mapping Survey. VII. APOGEE DR17 [C/N]–Age Calibration, 22-Apr-2022. [Online]. Available: https://iopscience.iop.org/article/10.3847/1538-3881/ac5d53. [Accessed: 18-Aug-2022].