Since I was young, I have always been interested in space and the universe. I constantly want to understand the vastness of space and what exactly the universe is. I read books, watch documentaries, and even have taken a trip to the Kennedy Space Center in Florida. I found it difficult in the beginning of my case study to find a data set that interested me. Low and behold one night I found a data set that contained different types of stars and their various characteristics. We have one massive star in our solar system which we know of as the sun, but most of what we can learn about the space and the universe is based on observing distant stars. Stars emit photons, and humans have the means to study photons in detail. For hundreds of years humans have studied star life cycles, how they are distributed in space, how they group into clusters in galaxies, and their effects on the surrounding gas and dust. Fortunately, stars are profoundly important for our Universe, even though they do not dominate its total material. Not only do they light up the nights sky, but they produce the raw materials that make life as we know it possible, and if there are other forms of life out there, it is most likely orbiting a distant star on its own planet similar to our own home planet of earth. We compensate for the relatively little we can learn about a single, typical star by studying them in large numbers and trying to understand how their properties change in groups that may include a range of masses, ages, and so forth. In this way, we can piece together a surprisingly detailed picture of how they work.

There are billions, most likely even trillions of stars in the universe. As mentioned above the more we understand stars, the more we can understand the universe and space. My goal was to create a predictive supervised model to accurately classify star types based on characteristics given from the data set. My audience to promote my model would be towards people in the field of space such as astronomers. This algorithm if successful, could assist in their everyday jobs by automatically classifying stars with an acceptably high accuracy. By having this model automated it could save time, resources, and money.

The data set was downloaded from Kaggle. It contains seven columns, six of which are input/predictive variables, while one is my target or output column that I will be trying to classify.

Input/Predictive Variables:

* + Temperature
  + Relative Luminosity
  + Relative Radius
  + Absolute Magnitude
  + Color
  + Spectral Class

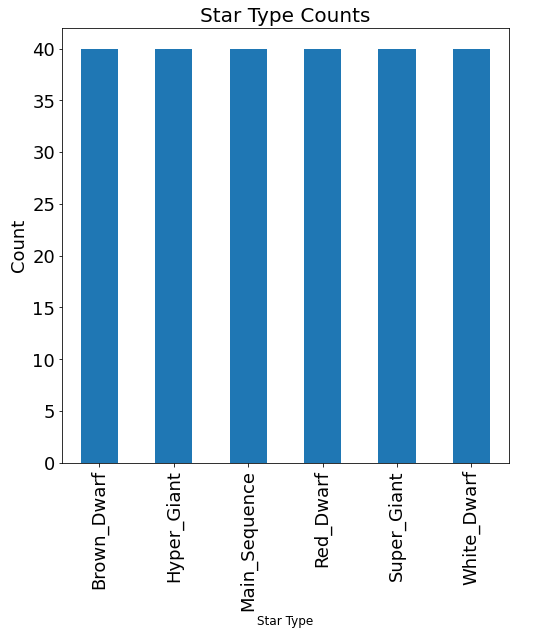
Output Variable

* + Type (Star Type)

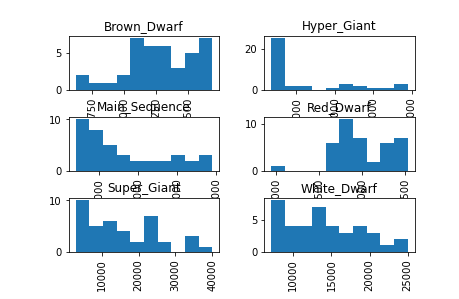
*Milestone 1*

Milestone 1 was the first step in my case study. The overall goal was to find a data set and then do some simple graphical analysis to understand the newly discovered data. Graphical analysis is defined as a graphical depiction of data using charts, figures, and graphs. Though this is one of the most basic types of analysis, it provides visualizations that are easy understand and interpret. Visualizations can aid in seeing how the data is distributed, if there are any trends in the data, or if there are any outliers.

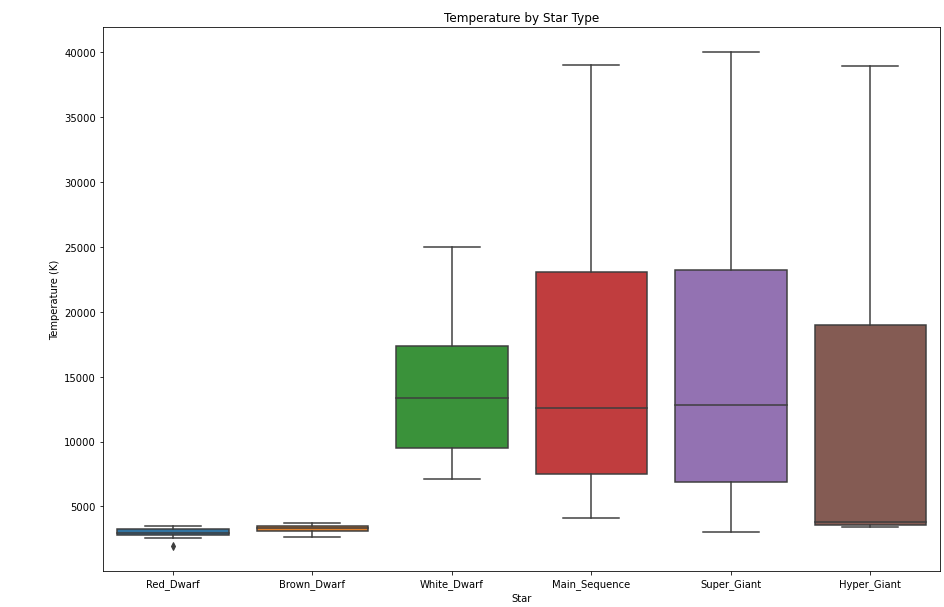
The main goal of my graphical analysis was to understand the state of the data I have. Is it messy? Are there outliers? Do I need to aggregate variables or even remove variables? These are all important questions that can be answered with graphical analysis. My first chart that I created was to see the counts of each type of star in the dataset. Fortunately, every star type had the same number of observations in the dataset (40). This was great for modeling because now I knew that the models will not weigh certain variables differently in its algorithm due to too many or too less observations of certain output variables.



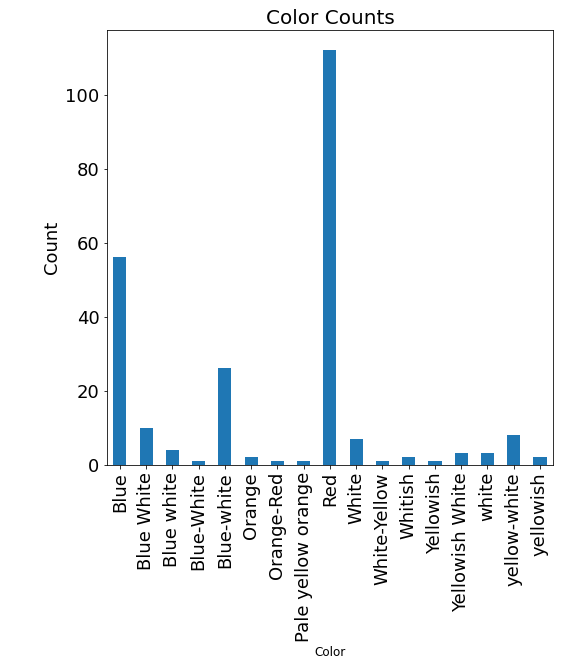
For the second graph(s) in my graphical analysis I wanted to see the distribution of each feature variable by each star type to check for any significant outliers. To get insight on the distribution of a variable it is best to use a histogram. A histogram is used to measure variables whose values are numerical and measured on an interval scale. It is a great tool to investigate distributions and outliers. I created four histogram plots of each numerical input variable by star type. The color and spectral class input variables are categorical; hence we cannot use a histogram to analyze. Below is the plot of Temperature by each star type. I do apologize for the poor resolution of the graphs. I had issues dealing with formatting as well as moving some graphs from Jupyter Notebook to a word document. For temperature we can see that each star tends to have a different range and center of temperature.



Notice in the histogram plots that the x-axis range was different for each star type. This made it difficult to compare the ranges against each star type. Another great graph for looking at the distribution of a variable is a box and whisker plot. This type of plot is a method for graphically depicting groups of numerical data through their quartiles. Box and whisker plots have lines that extend from the boxes indicating variability outside the upper and lower quartiles. The line running through the rectangular box is the median. Like that of the histogram plots, I also plotted box and whisker plots for each of the four numerical variables. To show you a comparison, the plot shown below is of Temperature; the same variable that was used to show the histogram plots above. Notice how it is easier to compare the numerical variables across each star type when they are side-by-side. An interesting observation in this plot is that the Red and Brown Dwarf stars have very low temperature and temperature variation compared to the other four star types.



The one variable that has not had much focus in the previous graphs is the color variable. I wanted to investigate how many different color types there were in the data set. To my surprise there were 16 different colors listed in the column, most of which were similar.



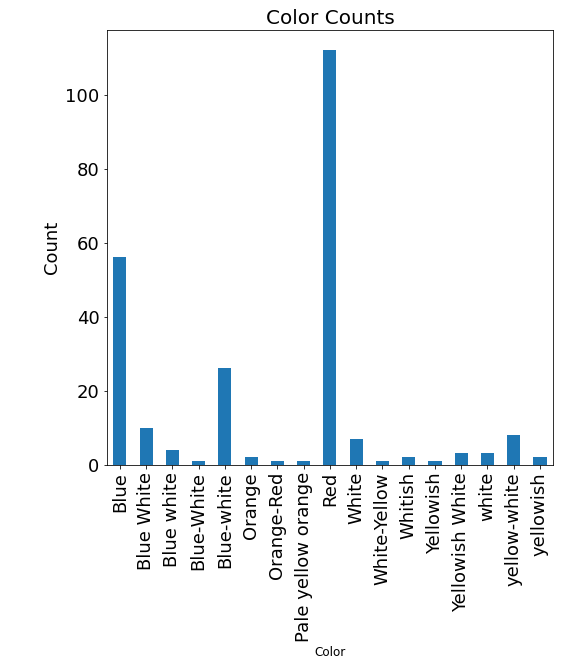
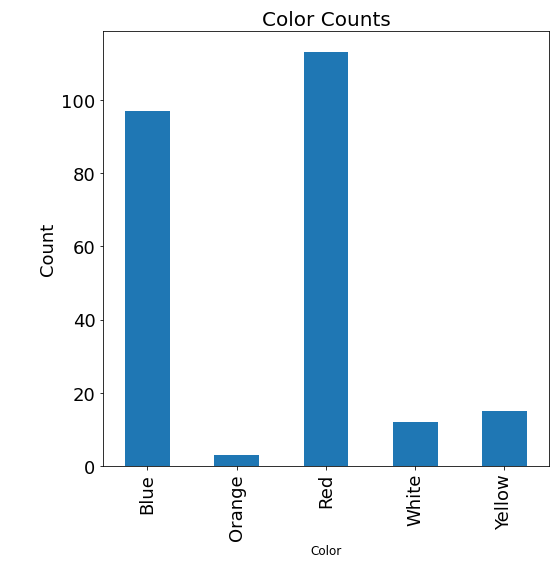
This was a great find because it is easy to see that this variable has numerous formatting issues within, and some colors can be condensed or reformatted to be grouped into the same category. I was able to adjust the color variable in Milestone 2 code section.

*Milestone 2*

Moving into Milestone 2 my goal was to make any adjustments to the dataset and begin preparations for model building. From Milestone 1 my main concern was with the color variable. The column had many colors that could be grouped together to minimize the many color options.



As a data scientist I tried my best to reduce the color groups into the most similar as seen above. There are some colors that I was unsure of which group to put them in such as “Pale Yellow Orange” and “Orange – Red”. Below are how the combined groups compare to the original graphical analysis. The color column went from 16 to 5 categories.

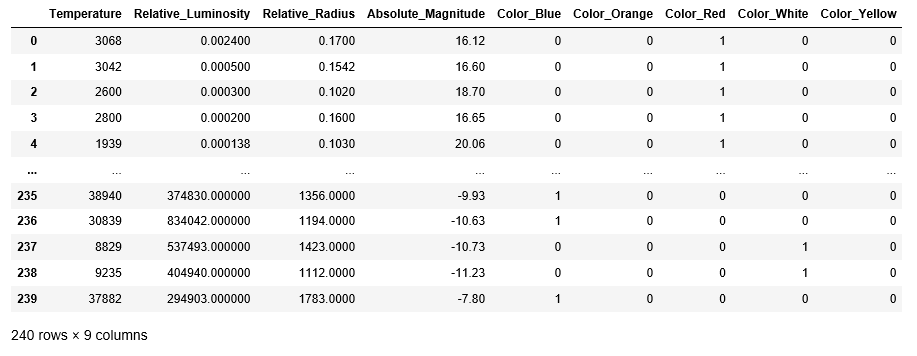
 

Moving on to other adjustments to the data, I removed the column “Spectral Class” because I could not get a clear concise definition of what that variable was representing. I attempted numerous searches and some reading but failed to gain an understanding of this variable. I try my best when doing analyses to understand what the variables are in the data set. By not understanding certain variables it can be hard to explain or correlate the results to the real world.

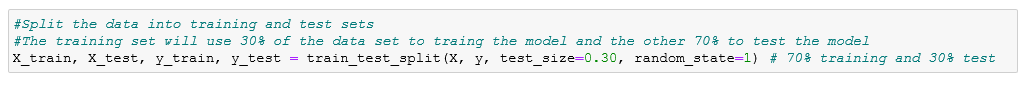
*Milestone 3*

For my first model I attempted a decision tree. A decision tree is a supervised learning technique that is a flow-chart like structure in which each internal node represents a test on an attribute. Each branch then represents the outcome of that test. I chose a decision tree for my first model because it is simple to understand, easy to interpret, and a great way to visualize how the algorithm is making its decision to classify each star. Decision trees also implicitly perform variable screening or feature selection.

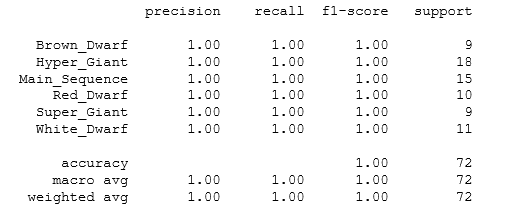
I built the model using the variables Temperature, Relative Luminosity, Relative Radius, Absolute Magnitude, and Color. Since the Color variable is categorical, I had to apply one-hot encoding to the dataset. One hot encoding is a method of converting data to prepare it for an algorithm to get a better prediction. It works by converting each categorical value (5) into a new categorical column and assign a binary value of 1 or 0 to those columns. This was my last step before finally attempting to create the model.



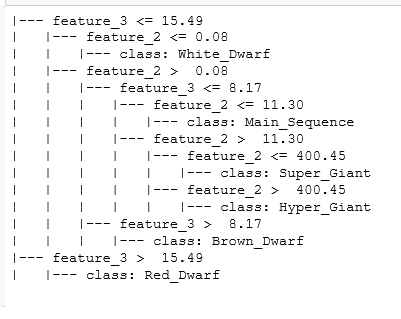
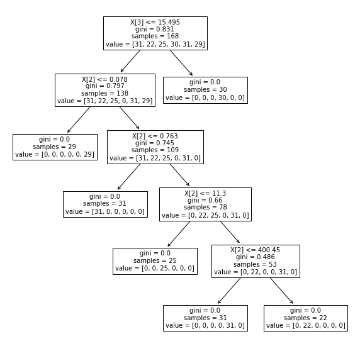
To create the model I used the DecisionTreeClassifier in sklearn.tree. I began by separating my dataset into feature and target variables. *X* was given the feature columns and *y* was given my target variables. Next, I split the data set into training and test sets. The training set will be used to train the algorithm and the test set will be used to test the model’s accuracy. 30% of the data will be in the training set, while the test set will contain the other 70% of the dataset.



Now that I had my training and test sets created it was time to build the decision tree model using DecisionTreeClassifier(). I fit the model using the training data sets of *X* and *y* (X\_train, y\_train), then used predict() to see how well my created model does against the test data sets of *X* and *y* (X\_test, y\_test). To show the predictive accuracy of the model I used a confusion matrix to visualize the results. A confusion matrix is a table that can visualize the performance of an algorithm.



From the confusion matrix we can see that our model was able to correctly predict the star type with 100% accuracy. One thing nice about creating decision tree models is that it is possible to see and interpret what the model is doing during its decision-making process. Below are two outputs of my decision tree algorithm.

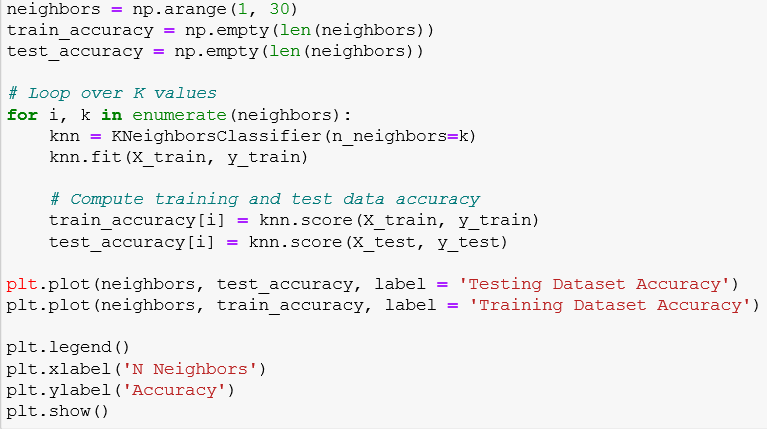
 

The model only needed the variables Relative\_Radius (X[2]) and Absolute\_Magnitude (X[3]) to be able to distinguish between star types with perfect accuracy! The most important attribute is placed at the root node which in my case was Absolute\_Magnitude. For the evaluation of the model we would start at the root node and work our way down the tree by following the corresponding node that meets the decision. This process continues until a leaf node is reached giving us our prediction on star type.

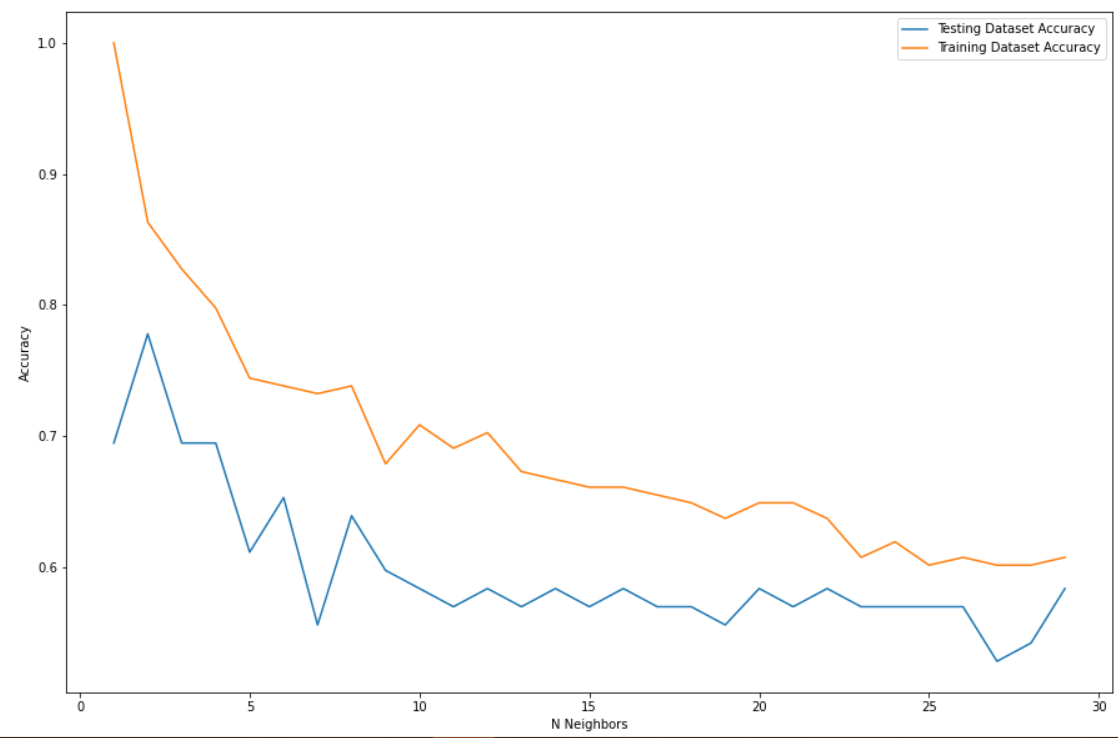
*Milestone 4*

After a very successful first try at modeling, I decided to try the K-Nearest Neighbor (KNN) Algorithm to see if I can predict star type with similar accuracy as the decision tree model. The KNN is also a supervised learning algorithm that is used for classification and regression. KNN algorithms use data to classify new data points based on similar measures. The classification is done by distance form neighboring points. The data is then assigned to the class which has the nearest neighbors. As you increase/decrease the number of nearest neighbors or *k* value, the accuracy can be affected. Hence why trying out multiple *k* values is important to see how the different values can improve/deteriorate accuracy. The only change I am going to make is to remove the color variable form the dataset because I felt uncomfortable using this variable since I made such a dramatic change to the number of colors in the variable.

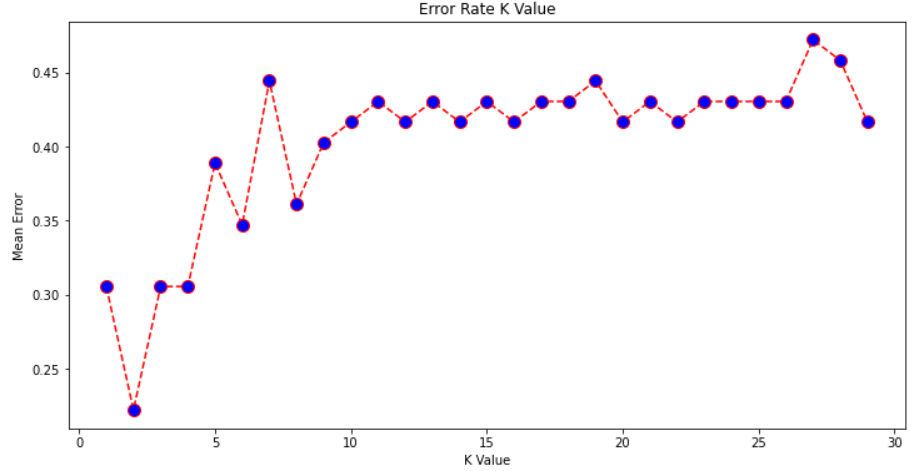
Similar to the Decision Tree Model I split my dataset into a training and testing set using 30% of the dataset for training and 70% for testing. To create the model I used KNeighborClassifier(). I used the function in a for loop to run through the algorithm using different values of k from 1 to 30.



The code above will output a graph that shows a comparison between the accuracy between the testing and training datasets. The best accuracy of the KNN model was when the value of *k* is 2



Playing around with some other visualizations with the model, I also created a graph to show the mean error of the models with the various values of *k.* Once again when *k* equals 2 we get the lowest error in the model.



Overall, the KNN model performed decent when trying to classify star types with the best accuracy of 77% at k equals 2. Out of both models I am obviously going to use the decision tree model that had 100% accuracy on the test data set.

The goal of my case study was to attempt to build a model to classify star types using various features or input variables. The first attempt at model building started with a decision tree algorithm and produced a model that could classify star types with 100% accuracy. When trying to attempt the same thing with a K Nearest Neighbor algorithm the results produced were less than desirable only being able to predict with ~78% accuracy using the best *k*. This case study taught me that we need to try different models when attempting machine learning. Yes, I did get lucky with my first attempt, but that will not always be the case. It was shocking to see such a dramatic difference in accuracy between the two types of models that both used the same dataset.

Looking back at my case study, there were a few challenges that caused some frustration. The first and obvious one is trying to find a data set that interested myself and create a project out of that data. The other difficulty was trying to find a modeling technique to accomplish my goal. There are so many different types and techniques when it comes to modeling, it really was a struggle to narrow down which specific ones to use. Out of this project, some additional opportunities could be to explore using the decision tree algorithm to differentiate between more categories of star types. The algorithm could go beyond stars and maybe predict types of planets or other objects in space.

To anyone looking to attempt machine learning I would recommend really doing some research and understanding how different models work with different types of projects. By having a good understanding of how different models work and understanding the pros and cons of a modeling technique, it can better the chances of being successful. There is no one size fits all solution for every dataset when it comes to machine learning, but with a better understanding and determination it is possible to be successful in machine learning.