I analyzed, drew insights and created a classifier model to determine a celestial star’s type based 6 attributes. My hypothesis is that I can create a model with 95% accuracy. There are 6 Star Types from 0 to 5 (integers).

***Data Insights***

* Temperature, Luminosity and Absolute Magnitude all have a right skewed distribution and are not normal; Radius’s distribution is also not normal; There to the statistical test needed to use on this data set will be non-parametric tests because I cannot assume the data set has a normal distribution.
* There is an equal distribution of star types in the data set, so when I train a model, I can easily avoid bias or will not need to bootstrap a particular star type.
* The Star Color distribution and the Spectral Class distribution seem to have a similar shape. This could mean there is a strong association between the color and spectral class.
* The index of the outliers of each feature (except Spectral Class and Star Color) actually were all Star Type 5, so I kept them as part of the analysis
* There seems to be no signs homoscedasticity or too much correlation after seeing the correlation values of the compared features.
* Double-checked with the Spearman Correlation test because the data set does not have a normal distribution; The test confirms there is a significant correlation between the features.
* Performed the Kruskal-Wallis Significance to test the significance difference between the feature data amongst the population of the Star Types. So, this way I know if a Star Type has too similar data values as another Star Type, which would lower the accuracy of the model; It was confirmed that the Star Types have different enough values to differentiate between them.
* I evaluated two different model types to see which one performed better using K-Fold Cross-Validation; The Support Vector Machine model type under-performed the Random Forest model type so I chose to use the Random Forest model type.
* The sampling method I used was Stratified Sampling method, to make sure the model trains with low bias and high variance.
* To tune the Random forest model, I discovered 30 Decision Trees and 5 K-Folds had a high chance of producing a model with the highest accuracy.
* I used the Chi-Square test for significance between the categorical features; This way I will know if I should include or not include one of the categorical features in the model training; Confirmed they are not too similar, there is a significant association between them.
* I saved the model to be able to put into a mini-application with having to re-train each time it tries to run a prediction.
* Using max and min values, I discovered each Star Type’s ranges for each feature or which categorical values fall into the respective types, so this way I can create simulated data in the proper value ranges to test the model after its trained, to make sure it is tested against data it has not seen before.
* Using Numpy I created sets of simulated data to test the trained model.
* I used Python’s Tkinter module to create a mini-application with sliders, so a user can create a simulated set of data for the model to predict with; The model has a 90% accuracy.
* Star Type 5 is very similar to Star Type 0, so the model does have a hard time classifying if a model is Star Type 5 because the great similarity between them. Part of this issue also comes from Star Type 5 having a wide variety of feature combinations. Star Type 5 can have the same feature value ranges as other star types as well.
* This model and analysis I did can be easily applied to business data and other types of data sets.