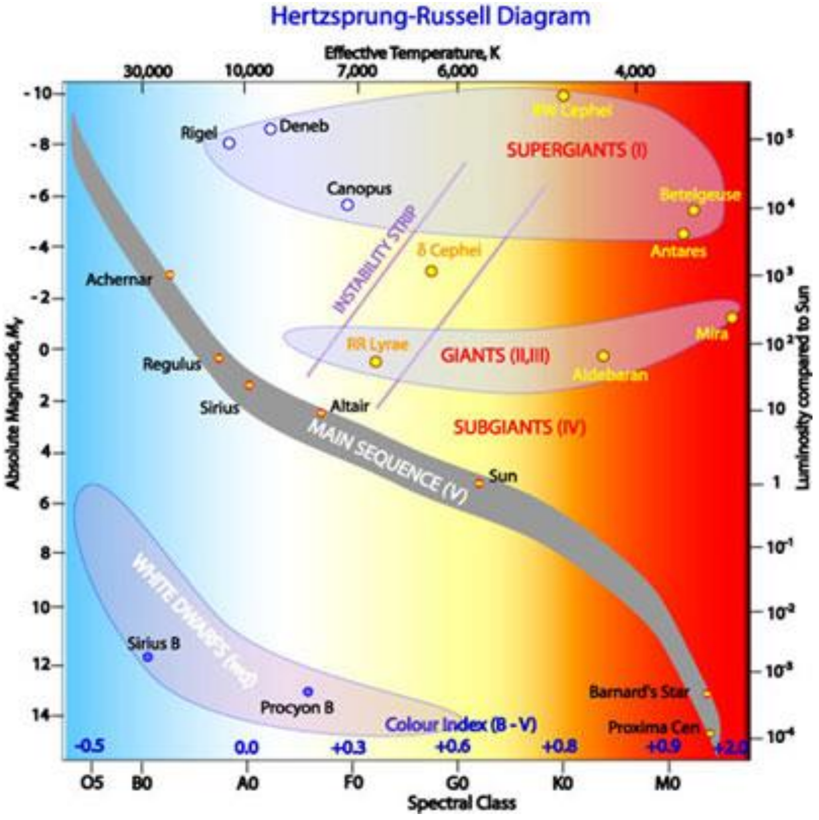
Star Classification

Crosby Pineda

Star Cluster

Stars in the universe are grouped into various clusters, each possessing unique characteristics. These stars are categorized based on specific classifications such as Brown Dwarfs, Red Dwarfs, White Dwarfs, Main Sequence stars, Supergiant, and Hypergiants.

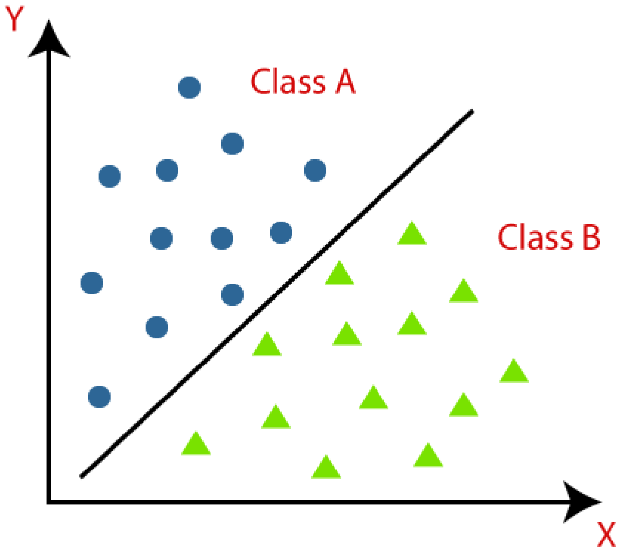


Star Classification

It is especially because there are so many stars and clusters of them that it would be helpful to identify the type of stars, therefore the power of prediction becomes all the more needed. It is like finding a needle in a haystack of the cosmos.

How?

Since this is a classification problem, different but similar to my original plan of identifying particles, I wanted to use a boosted decision tree. I also wanted to do regression, but with my data and the type of problem; it is not quite the same as a regular regression. For this reason, I decided to use the Ridge Classifier. In essence my 2 approaches are using machine learning to classify/ using the regression version of the combination of of the 2.

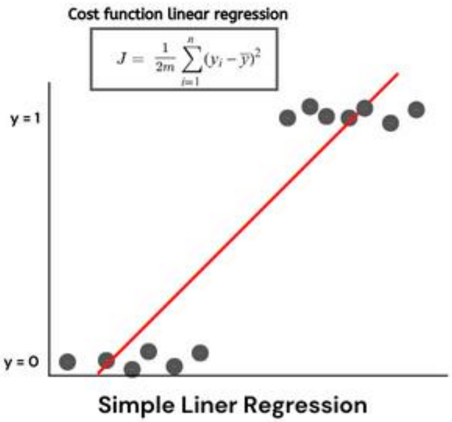


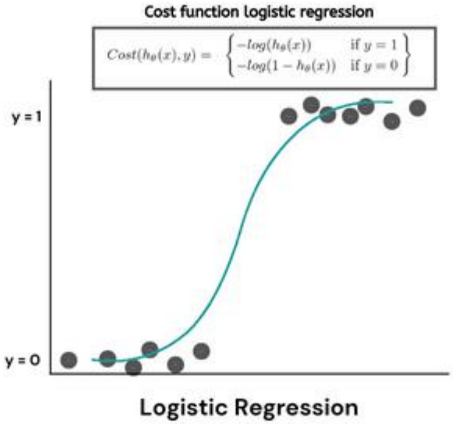
Ridge VS BDT

Ridge Classifier is a linear model, a logistic regression model with Ridge regularization. Hence, it is closer to regression in terms of its approach. Good for linear relationships between features and classes.

BDT is good for nonlinear relationships between features and classes.

By comparing the 2 we will see which one is more accurate.

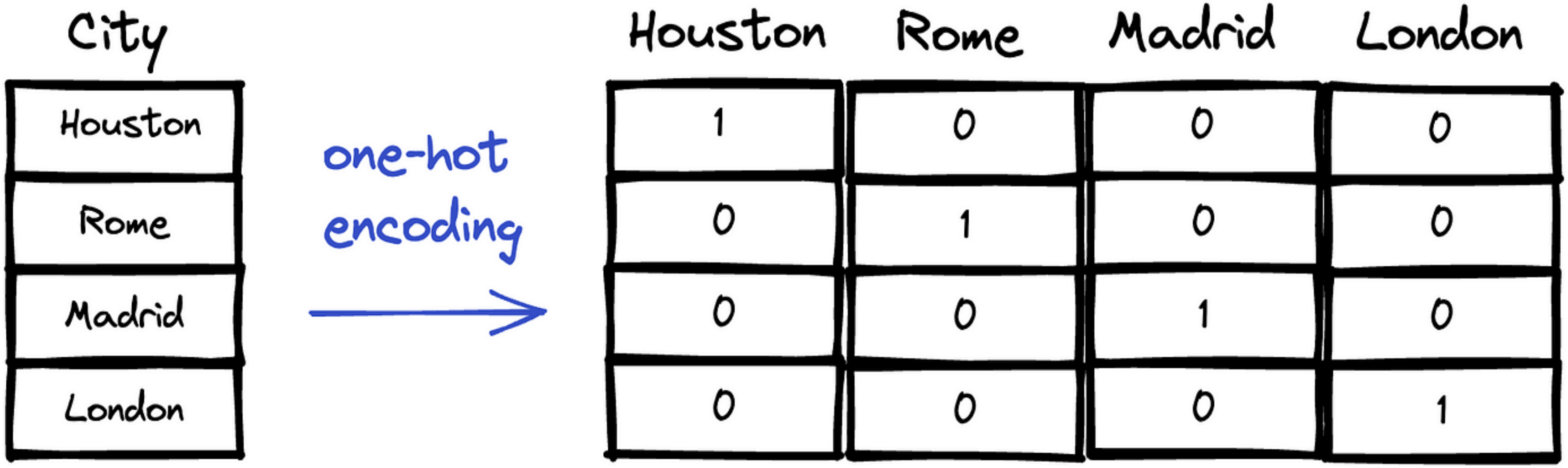




XGBoost libraries

Using XGBoost I can make a Boosted Decision Tree, where many trees are trained in succession.
Using, well you guessed it, RidgeClassifier for the Ridge Classifier, to compare BDT to.

Also, an important one to make this all work is OneHotEncoder for categorical data.

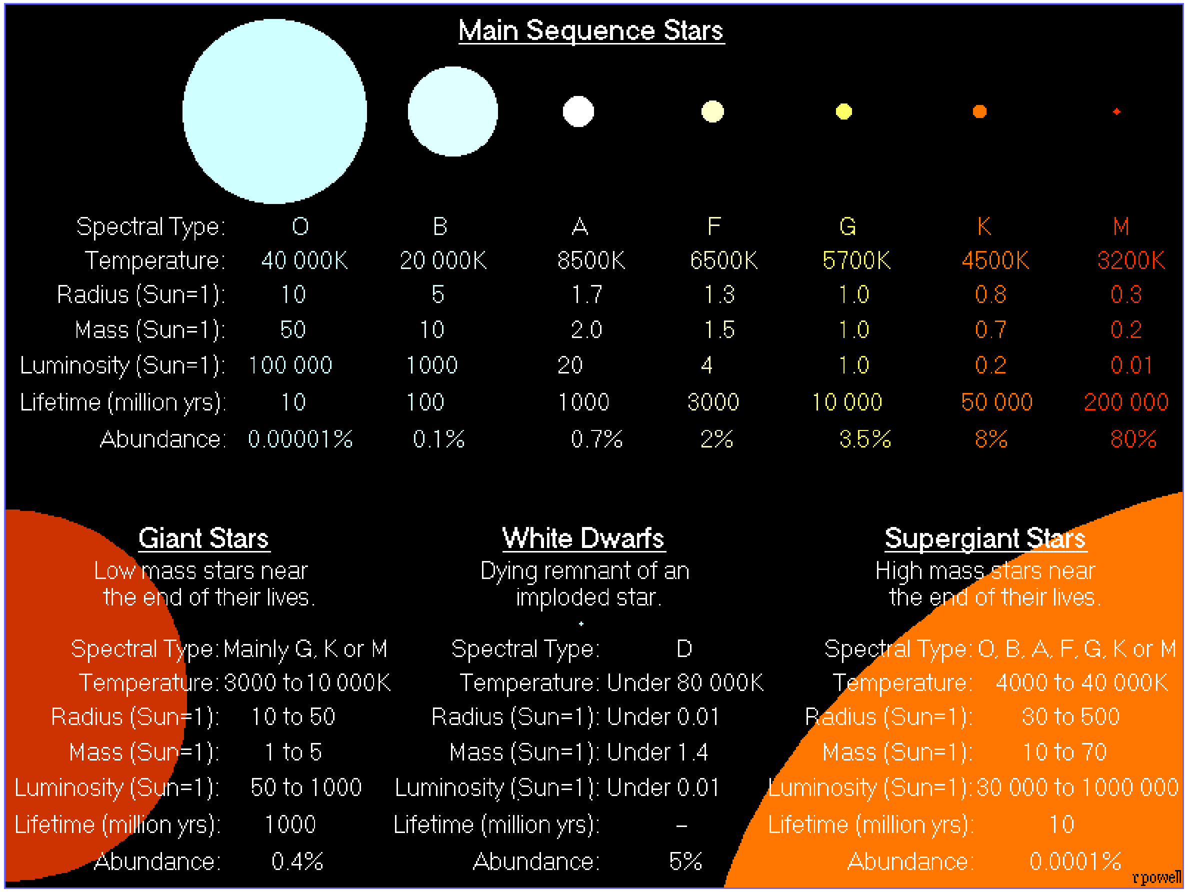


City
Houston
Rome
Madrid
London

Houston	Rome	Madrid	London
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Features

Deciding on the features was not too difficult. The features should just be the variables that make up different star types:
Temperature, Luminosity, Radius (size), Absolute magnitude, Star color, and Spectral Class. A total of 6 variables that I will use to try and make predictions.



Main Sequence Stars

Spectral Type:	O	B	A	F	G	K	M
Temperature:	40 000K	20 000K	8500K	6500K	5700K	4500K	3200K
Radius (Sun=1):	10	5	1.7	1.3	1.0	0.8	0.3
Mass (Sun=1):	50	10	2.0	1.5	1.0	0.7	0.2
Luminosity (Sun=1):	100 000	1000	20	4	1.0	0.2	0.01
Lifetime (million yrs):	10	100	1000	3000	10 000	50 000	200 000
Abundance:	0.00001%	0.1%	0.7%	2%	3.5%	8%	80%

Giant Stars

Low mass stars near the end of their lives.

Spectral Type:	Mainly G, K or M
Temperature:	3000 to 10 000K
Radius (Sun=1):	10 to 50
Mass (Sun=1):	1 to 5
Luminosity (Sun=1):	50 to 1000
Lifetime (million yrs):	1000
Abundance:	0.4%

White Dwarfs

Dying remnant of an imploded star.

Spectral Type:	D
Temperature:	Under 80 000K
Radius (Sun=1):	Under 0.01
Mass (Sun=1):	Under 1.4
Luminosity (Sun=1):	Under 0.01
Lifetime (million yrs):	–
Abundance:	5%

Supergiant Stars

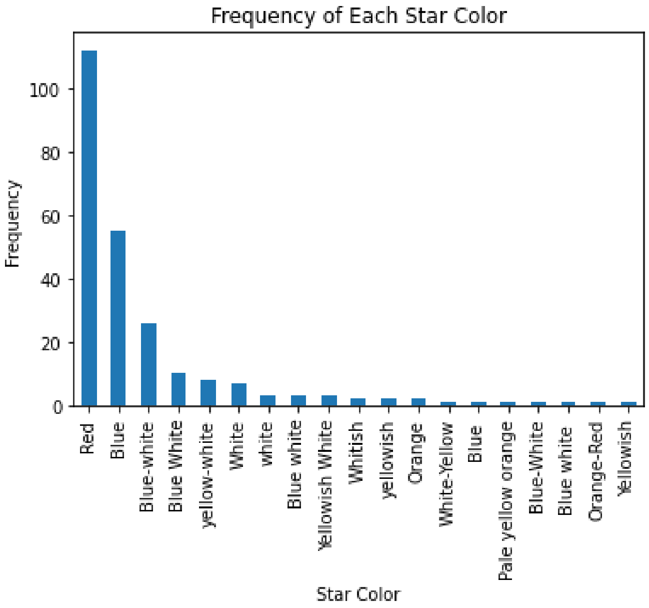
High mass stars near the end of their lives.

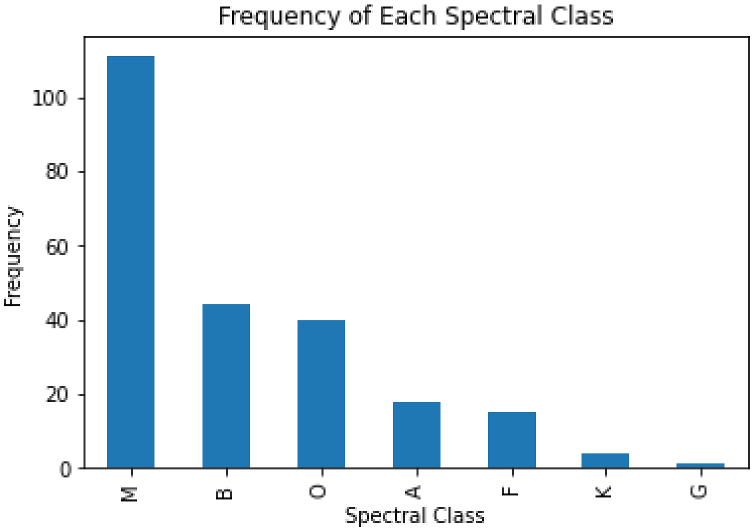
Spectral Type:	O, B, A, F, G, K or M
Temperature:	4000 to 40 000K
Radius (Sun=1):	30 to 500
Mass (Sun=1):	10 to 70
Luminosity (Sun=1):	30 000 to 1000 000
Lifetime (million yrs):	10
Abundance:	0.0001%

rpowell

Issues

The distribution of the data is not symmetrical or even. This means there are prone to be more biases than I would like. Such as the amount of Star color descriptions or the different spectral classes.





Issues

Specifically, star color descriptions seemed a bit more subjective than say the size of a star. Each one had different and varying amounts of descriptions. Not to mention it was also categorical. Such as Blue white, white, and whitish.

Blue White
White
White
White
Blue White
Yellowish White
Blue white
Yellowish White
Yellowish White
Pale yellow orange
Blue
Blue-white
Blue-white
Whitish
yellow-white
Whitish
yellow-white
yellow-white
yellow-white
yellow-white

Data Prep

In order to handle this I had to ensure that for star color, the descriptions appeared at least twice.
Using One-Hot Encoding to convert categorical data to have unique values to be used for machine learning to handle star color and spectral class.

using aspects of linear algebra in encoding categorical features and transforming them to be put back into the feature set to be able to be used for machine learning.

Train Test Split

To make the Train Split I made sure to use a random seed to not only replicate it, but also to randomize the distribution. While at the same time, I had to ensure that given the disruption of the data, the split had a bit of each for star color description. If I did not do this then the accuracy would be lowered if the training set did not have one of each.

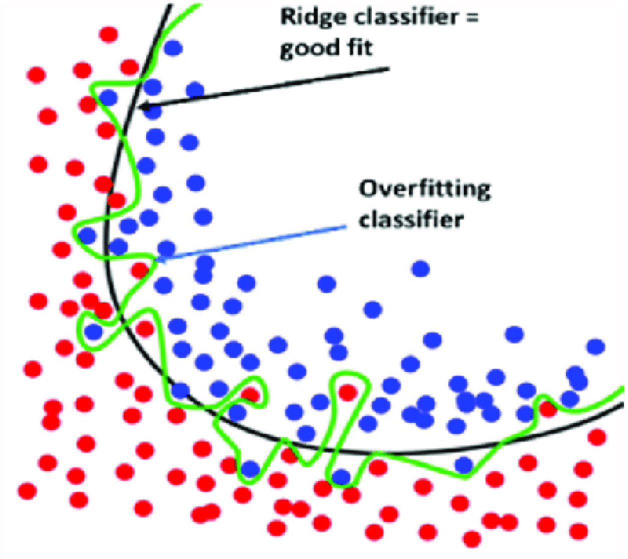
Train Test Split

I used about 80% of the data to train and 20% of the rest to test. This falls within the usual ratio. Was curious to see other ratios, and it does make both classifiers less accurate.

Dataset
Training Set
Test Set

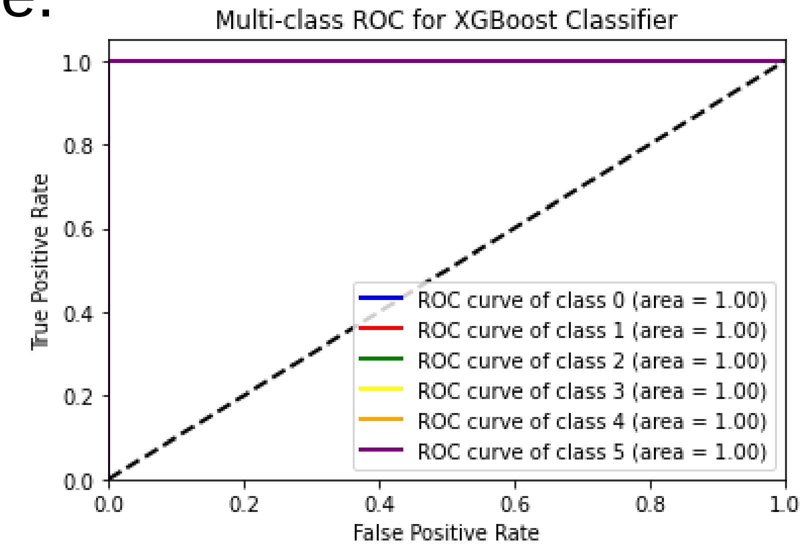
BTD and RC

After doing all of that I then used Ridgeclassifier and xgboost to do the machine leaning aspect. Most of the hard part was getting the data prep and integrating the categorical data.



ROC Curve

As I stated in my original plan, I did want to do a ROC curve to see the accuracy. I did this for each star type to see the accuracy of each one. Which each one was 100% accurate.



Overall Accuracy

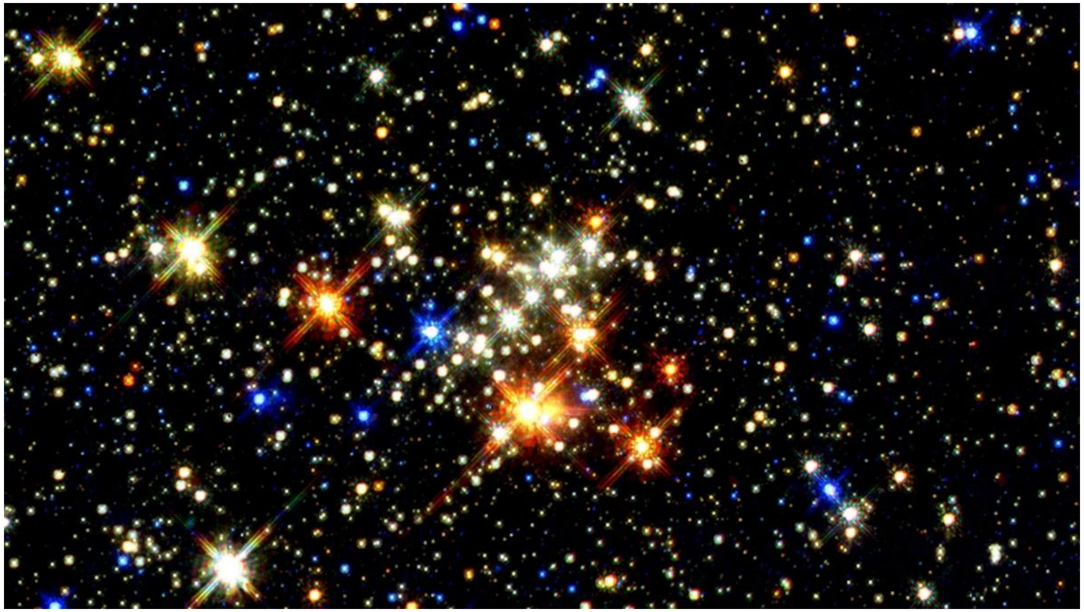
The over all accuracy for my BDT was 100% as the accuracy of each class was 100%. My Ridge Classifier on the other hand was about 91%. This means in the specific circumstance that the BDT was better at predictions.

Ridge Classifier Accuracy: 0.9148936170212766

XGBoost Accuracy: 1.0

Thoughts

While I do not think a BDT should be 100% accurate, I think it largely has to do with my sample size being rather small. It was about 200 +. Which is rather small compared to the billions of stars in the universe.



Simplifications

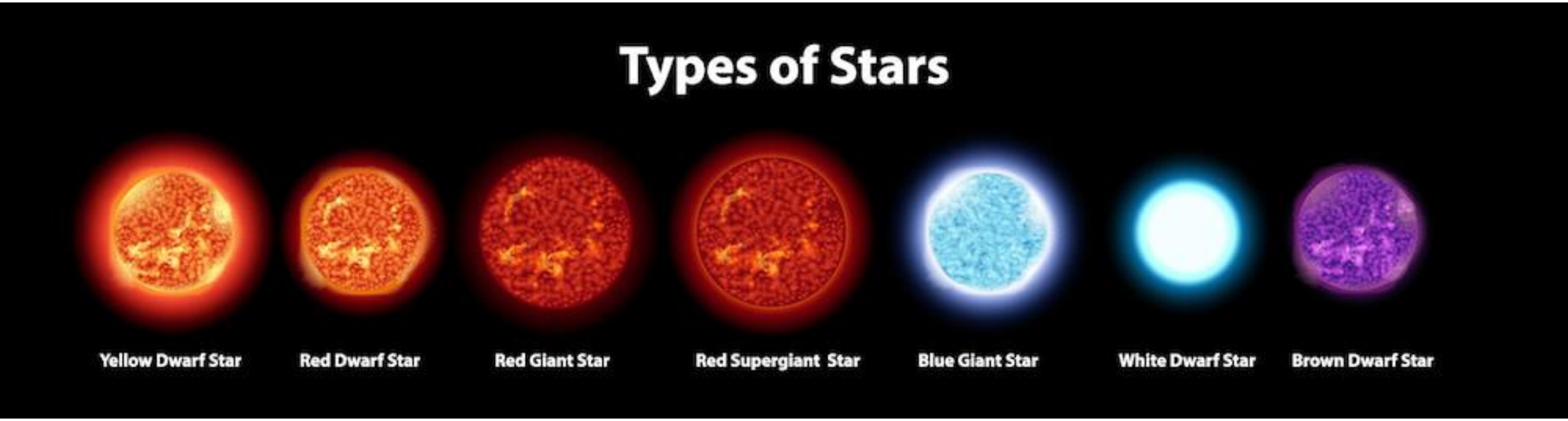
My simplifications interfered with the accuracy as I eliminated star colors that do not repeat and also ensured that they were distributed between the split. This could have also been the reason for it.

🚫 Ridge Classifier

✓ Boosted Decision Tree

What Can we Say

We can say that perhaps from out sample size that the relationship between the features and class are not exactly linear. Physically this means that: Temperature, Luminosity, Radius (size), Absolute magnitude, Star color, and Spectral Class may not have a strictly linear relationship with the star type. After all the Ridge Classifier is effective at linear relationships, so if it was about 91% then its not quite there.

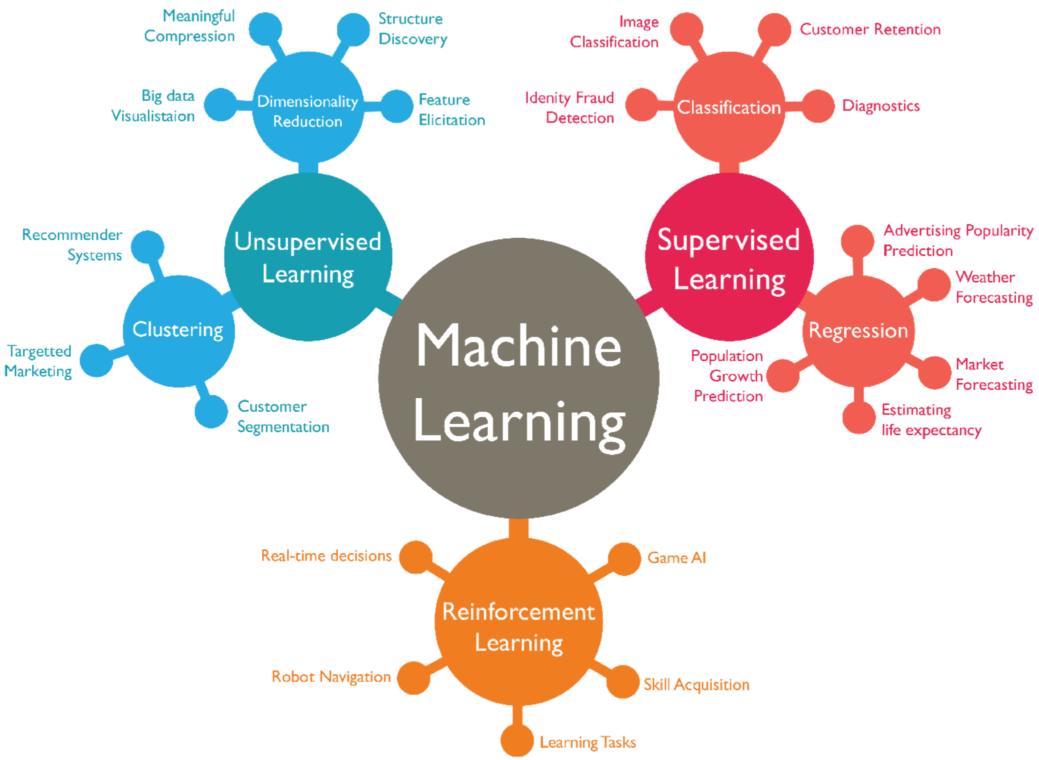


Relationships

Rather the relationship tends to be more complex and multifaceted than just a 1 to 1. This makes sense because it required 6 variables to make a single prediction. This is why machine learning should be used for complicated relationships.

Conclusion

This was a fun project, that made me go to areas I am not quite familiar with. We see the importance of using the appropriate type of machine learning for the problems at hand, it is not just machine learning solves all. I got my feet wet on the concepts on machine learning. There is far more on the horizon.


Machine Learning
Unsupervised Learning
Dimensionality Reduction
Meaningful Compression
Structure Discovery
Big data Visualistaion
Feature Elicitation
Clustering
Recommender Systems
Targetted Marketing
Customer Segmentation
Supervised Learning
Classification
Image Classification
Customer Retention
Idenity Fraud Detection
Diagnostics
Regression
Advertising Popularity Prediction
Weather Forecasting
Population Growth Prediction
Market Forecasting
Estimating life expectancy
Reinforcement Learning
Real-time decisions
Game AI
Robot Navigation
Skill Acquisition
Learning Tasks