**Star Classifier**

**What is it?**

* It is an **UNIQUE** application of **Machine Learning(ML)** in **ASTRONOMY** which is used to classify stars **(whether it is a Red dwarf, White dwarf, Supergiant or Hypergiant).**

**# I’m doing a 4 class classification**

**What it does?**

* It uses a Learning algorithm (Deep Neural network) which learns data of stars from a **CSV file (an excel file format)**.
* **After Learning Data** , this algorithm is used to **test new data** and **predict star types.**

**Background Check of Star classes!**

* Stars in the universe can be classified into several groups . The groups I have used here are :
* **Red Dwarf**
* **White Dwarf**
* **Supergiant**
* **Hypergiant**
* These groups are classified based on several characteristics of stars such as :
* **Luminosity(L/L0) \*\***
* **Radius (R/R0) \*\***
* **Surface Temperature(T/T0) \*\***
* **Absolute Magnitude(Mv)**
* **Spectral Class ( O, B,A,F,G,K,M)**
* **Star Color**

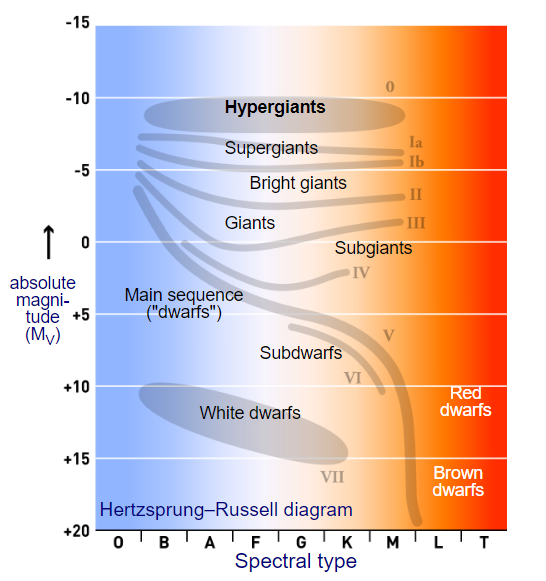
**\*\*** The **Luminosity** , **Radius** and **Surface Temperatures** are taken with respect to that of **Sun’s** . Subscript 0 denotes these values of Sun.

**L0 = 3.828 x 1026 Watts**

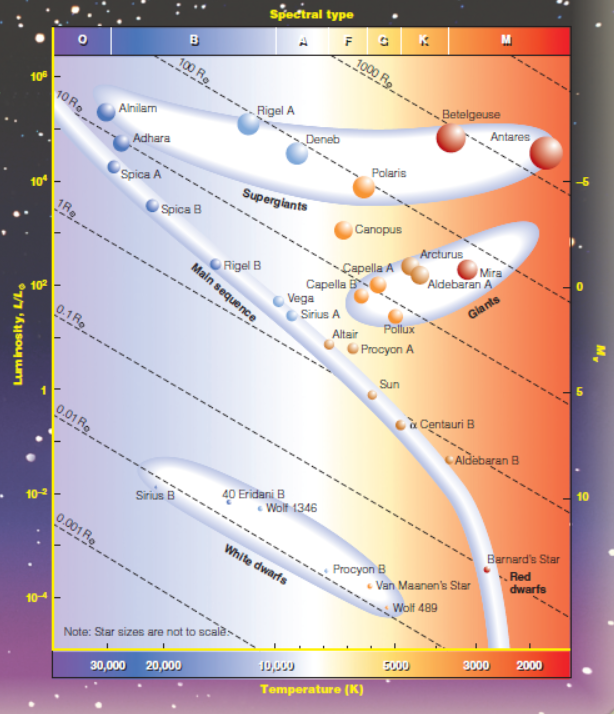
**R0 = 6.9551 x 108 m**

**T0 = 5778 K**

* The classification of stars is done using a H-R Diagram as shown below:



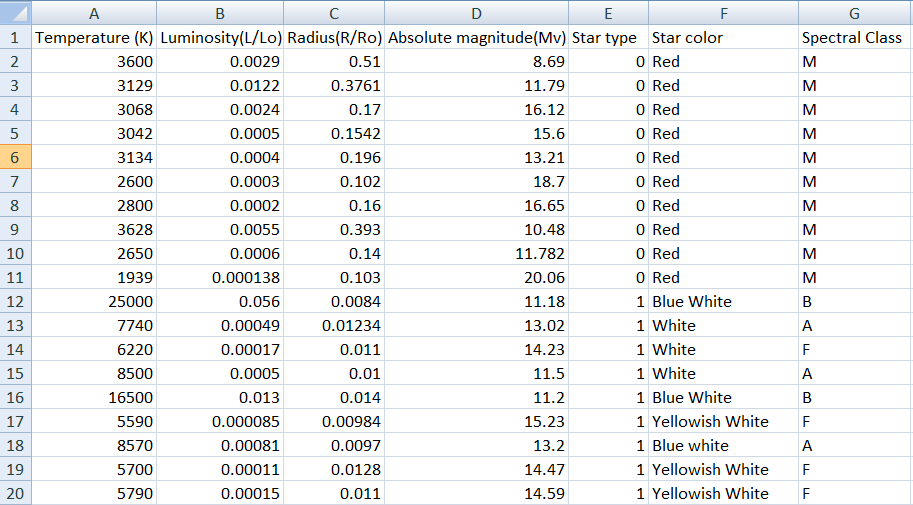
**Fig 1: H-R Diagram 1**

****

**Fig 2: H-R Diagram 2**

**How it works?**

* The dataset of stars are loaded into the program and it looks like given below:



**Fig 3: A sample Star dataset (CSV file)**

* Star Type denotes **:**

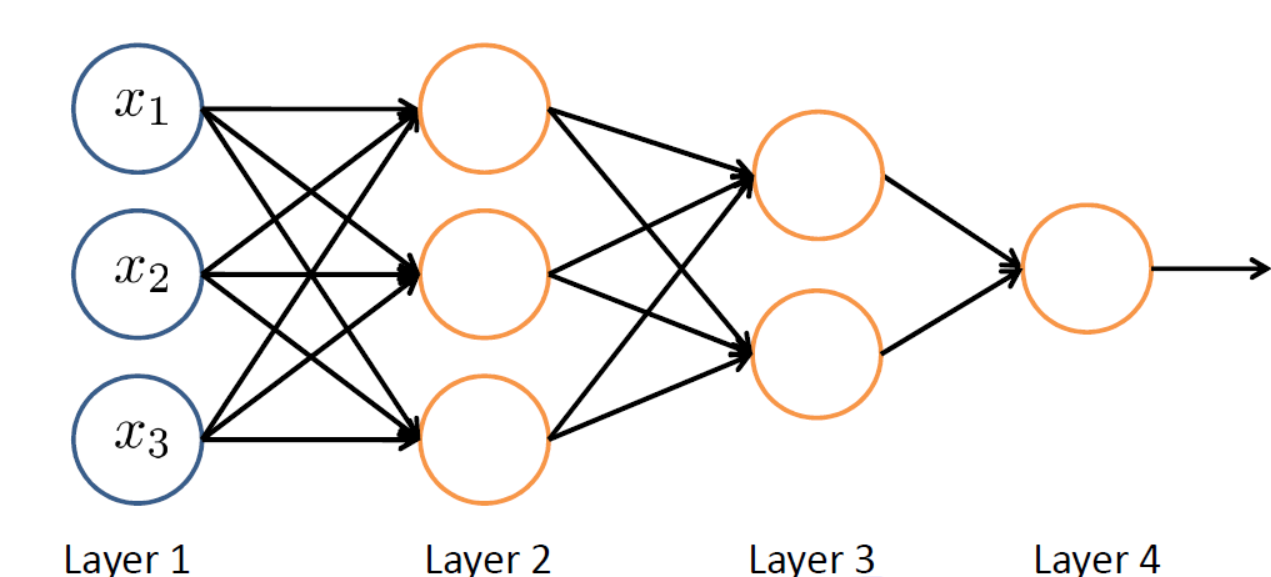
**0 – Red Dwarf**

**1 – White Dwarf**

**2 – Supergiant**

**3 – Hypergiant**

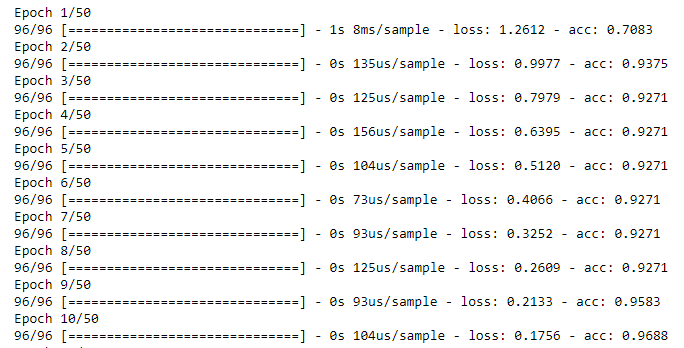
* This Data is fed into an algorithm called Deep Neural Network which looks like this:

****

**Input data Hidden layers Output**

**Fig 4: A Deep Neural Network**

* Algorithm learns 60% of the data (randomly distributed) . The learning takes place as follows:

****

**Fig 5: Process of Training Data**

* Here “Epoch” denotes the number of iterations completed.
* “loss” and “accuracy” are the factors which indicates how well the algorithm learned the data.

**How it Predicts Star Classes?**

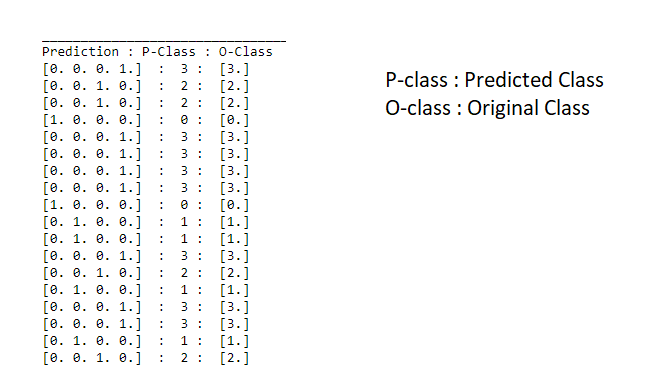
* From the data it learned, new data are tested and the algorithm predicts star types in the form of an array:

[1,0,0,0] denotes **class 0** : Red dwarf

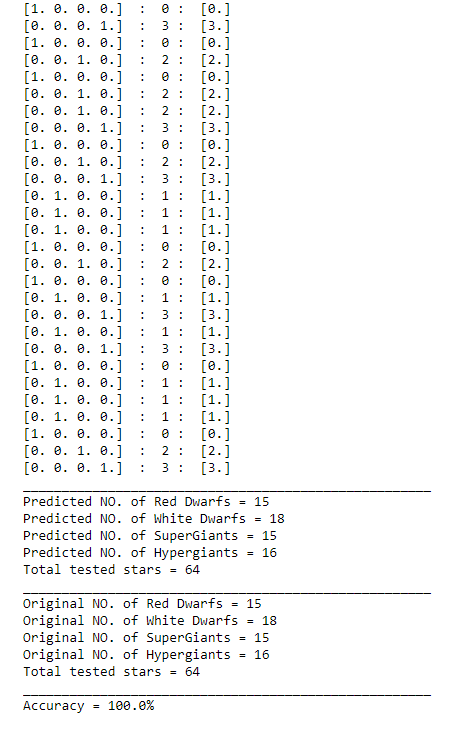
[0,1,0,0] denotes **class 1**: White dwarf

[0,0,1,0] denotes **class 2**: Supergiant

[0,0,0,1] denotes **class 3**: Hypergiant



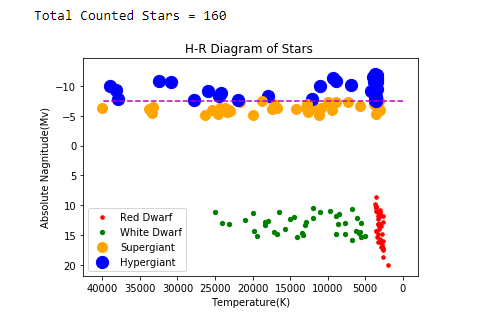
**Fig 6: Predicting Star types of tested data**

****

**Fig 7 : Predicting Star Types with an accuracy**

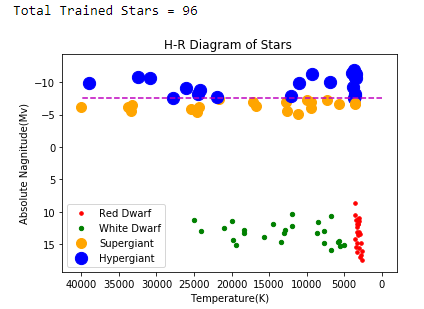
**How to Visualize the data?**

* Data visualization is a key aspect of ML .
* First we need to know that most of the stars in the universe can be classified based on the H-R Diagram of stars shown previously.
* Let’s check out the H-R Diagram for all the stars in our dataset.

****

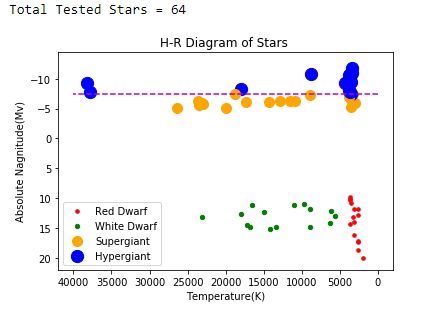
**Fig 8: H-R diagram of whole Dataset**

* From the 160 stars, we trained 60 % (i.e. 96) stars randomly . The H-R Diagram for trained stars looks like:

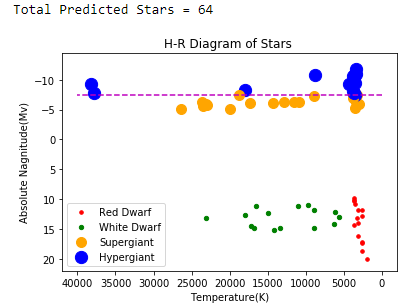


**Fig 9: H-R Diagram of Trained stars**

* After analysing the original tested dataset, if the predicted dataset is almost similar (at least 98% accurate) to that of the tested dataset then we can say that the classifier learned efficiently (In our case we got 100% accuracy).



**Fig 10: H-R Diagram of Original tested stars**

****

**Fig 11: H-R Diagram of Predicted stars**

**Clearly, we can see that both the above H-R diagrams are equivalent and hence our classifier worked efficiently and thus can be applied to predict future datasets of stars.**

**N.B. :- More the data u train, better it will predict**