IBM MACHINE LEARNING: CLASSIFICATION PROJECT

Nasa Star Type Classification using SVC, Random Forest and Decision Trees

1. Objective

Classification is a very useful task in machine learning and used in many fields. In this project we try to classify the star type given certain features using three different classifiers and compare the performance of the three models.

2. Data

The dataset of star type is given by NASA and contains a total of 240 instances with 6 features and 1 target. The dataset contains 4 continuous data features and 2 categorical data features. The target is going to be the star type where there are six-star types.

Following are the features in the dataset:

1. Temperature – K
2. Luminosity -- L/Lo
3. Radius-- R/Ro
4. AM Mass – Mv
5. Color -- General Color of Spectrum
6. Spectral Class -- O, B, A, F, G, K, M / SMASS

Following are the target details:

1. Red Dwarf - 0
2. Brown Dwarf - 1
3. White Dwarf - 2
4. Main Sequence - 3
5. Super Giants - 4
6. Hyper Giants – 5

3. Data Preprocessing/Cleaning

3.1 Data Exploration

The first step in any machine learning project is to explore the data and get to know the details like the type of feature variables, data distribution, check for missing values and other information.

```
1 df.dtypes # Describe Datatypes
```

```
Temperature          int64
L                  float64
R                  float64
A_M                float64
Color               object
Spectral_Class      object
Type                 int64
dtype: object
```

Figure 1. Feature variable type-Numeric or Categorical. Here there are 4 Numeric and 2 Categorical Feature.

	Temperature	L	R	A_M	Color	Spectral_Class	Type
0	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...
235	False	False	False	False	False	False	False
236	False	False	False	False	False	False	False
237	False	False	False	False	False	False	False
238	False	False	False	False	False	False	False
239	False	False	False	False	False	False	False

Figure 2. Check if there are null values. There are none.

```
1 df.nunique() # Unique Values in each column
```

```
Temperature        228
L                  208
R                  216
A_M                228
Color               17
Spectral_Class       7
Type                 6
dtype: int64
```

Figure 3. Checking for Unique Values present in each feature.

3.2 Data Visualisations:

A good way to understand our dataset is by visualizing our features through histograms, box plots and other visualization graphs.

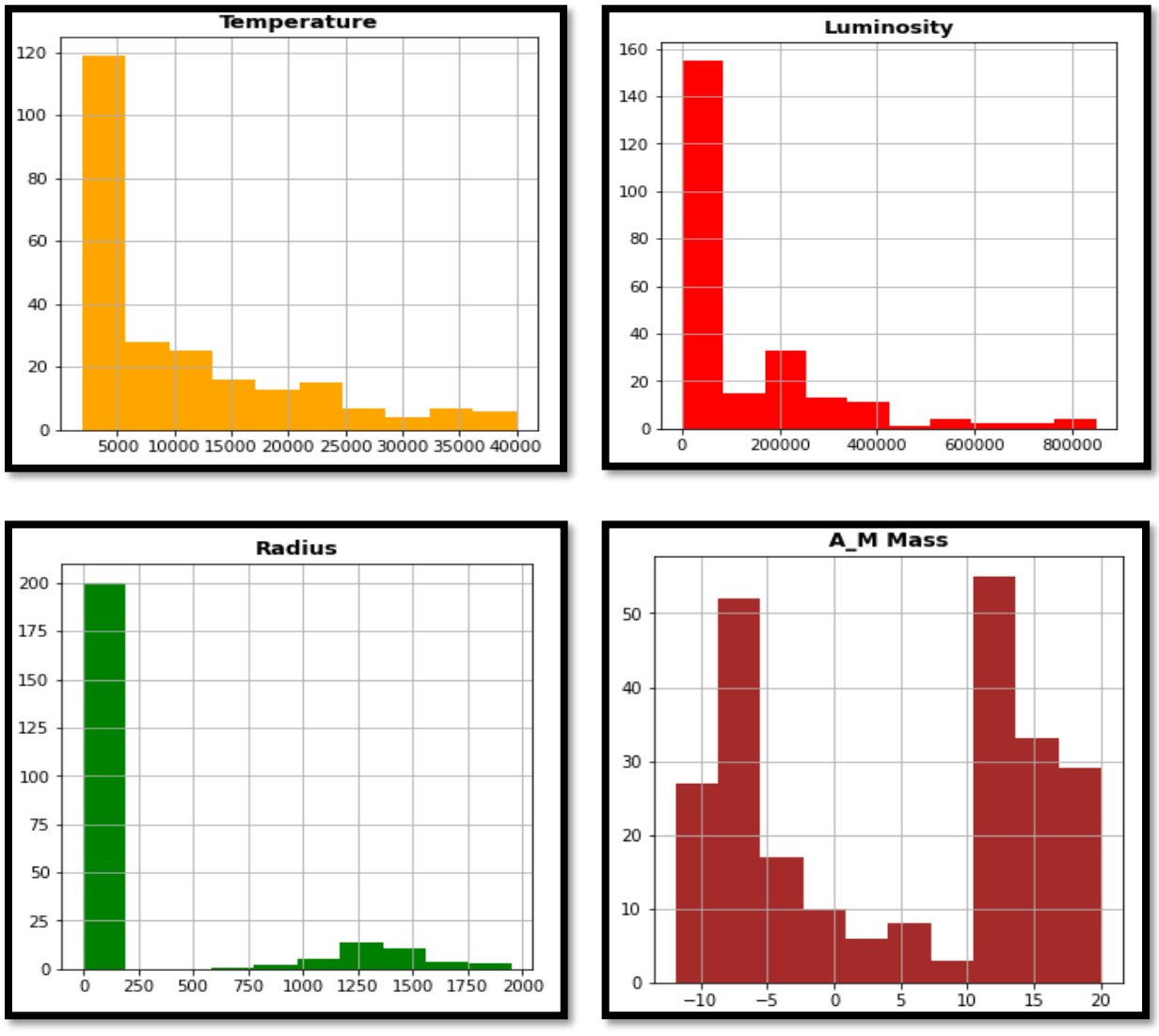

Figure 4. Histograms of Numerical Features.

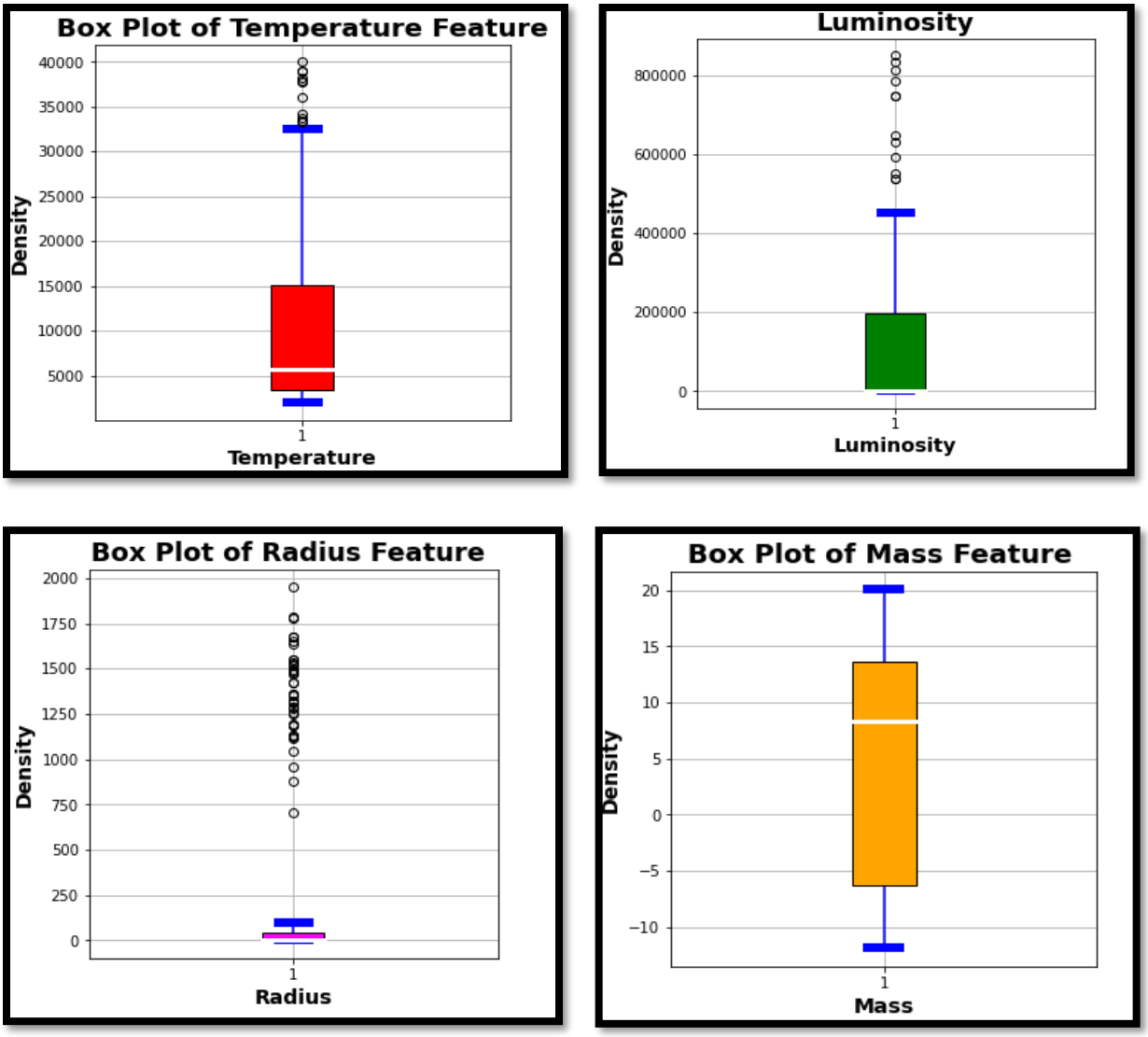

Figure 5. Boxplots of Numerical Features.

Histograms generally provide us the distribution of data and the boxplots tells us the statistical information of the features like median, percentiles and outliers.

3.3 Data Preparation

Before training the data has to be prepared. The categorical features need to be one hot encoded and then trained. The data then has to be split into training and testing dataset with 80% as training and 20% test split. Then the training and testing features need to be normalized.

Shuffle Dataset

```
1 df = shuffle(df) # Shuffle Data
```

Make two dataframes of labels and features

```
1 df_features = df.drop(labels=['Type'], axis = 1) # Drop the Type column
2 df_labels = df[['Type']] # Labels
```

One hot encode " Color " Column

```
1 df_colors = pd.get_dummies(df.Color) # Colors one hot encoding
```

One hot encode " Spectral Class" Column

```
1 df_spectral = pd.get_dummies(df.Spectral_Class) # Spectral Class One Hot Encode
```

Figure 6. Shuffle and One Hot Encode.

Normalize

```
1 scaler = StandardScaler() # Standard Scaler
2 X_train = scaler.fit_transform(X_train) # X-train
3 X_test = scaler.transform(X_test)# X-test
```

Figure 7. Standard Normalization.

4. Methodology

In order to conduct classification, we use three different type of classifier on our dataset. The three different classifiers are:

1. Support Vector Machines
2. Random Forests
3. Decision Trees

1. Support Vector Machines:

We use support vector machines to classify our dataset. A linear kernel with regularization parameter of 2.0 is used in the current model. The training set is fitted to the model and using the test set we get the predictions. Using the testing data, we plot the confusion matrix and calculate the accuracy of our model to get the performance of SVM

2. Random Forests:

Random forests are popular classifiers used in classifications. Here we initialise the random forest model with estimators' number of 100. The training set is fitted to the model and using the test set we get the predictions. Using the testing data, we plot the confusion matrix and calculate the accuracy of our model to get the performance of Random Forest.

3. Decision Trees:

The third model which we use is decision trees. The decision trees are initialized with maximum depth of 5 and with entropy criterion. The training set is fitted to the model and using the test set we get the predictions. Using the testing data, we plot the confusion matrix and calculate the accuracy of our model to get the performance of our Decision Tree Model. We also plot the Decision Tree Pathway.

5. Results

In order to verify the performance of the three models, we plot the confusion matrix to see how the model predicted true positives. We also check the performance of the model by seeing the accuracy

5.1 Support Vector Machines

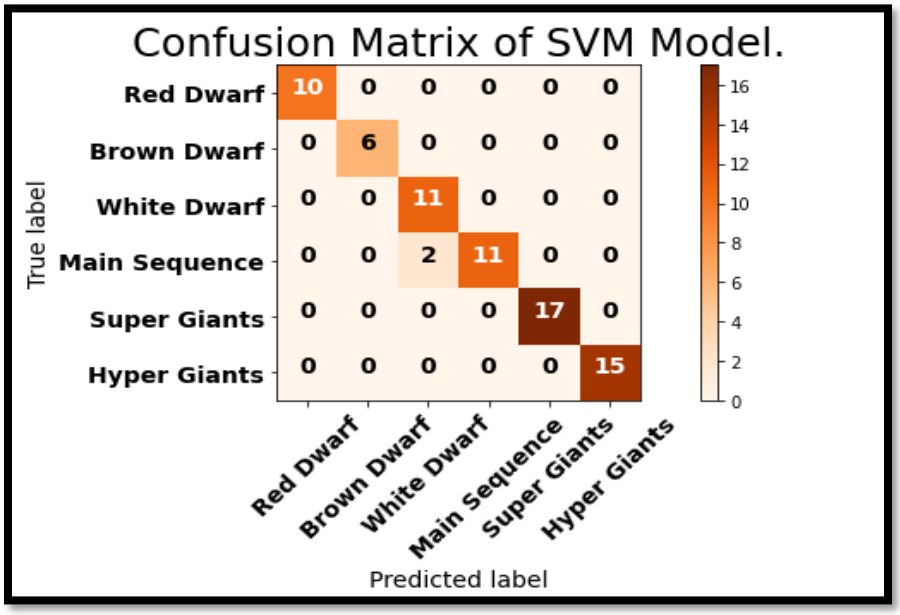

Figure 8. Confusion Matrix of SVM Model.

Except the class of Main Sequence all other classes were correctly classified and 2 labels in Main Sequence were falsely classified as white dwarf.

Classification Report Support Vector Machines:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	6
2	0.85	1.00	0.92	11
3	1.00	0.85	0.92	13
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	15
accuracy			0.97	72
macro avg	0.97	0.97	0.97	72
weighted avg	0.98	0.97	0.97	72

Figure 9. Classification Report of SVM Model Prediction.

The accuracy of SVM was obtained to be 0.97.

5.2 Random Forest

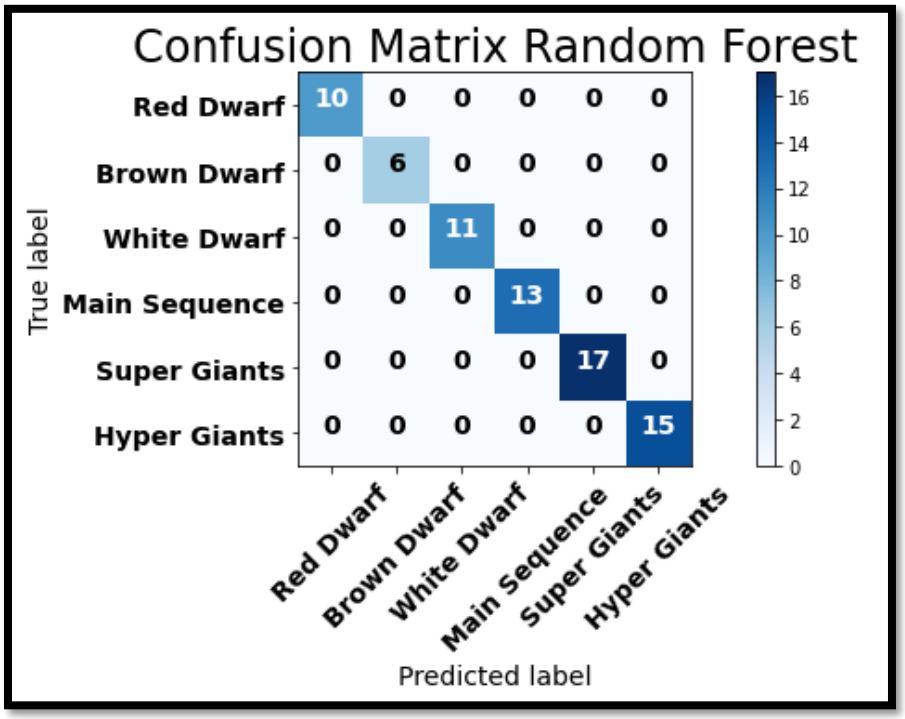

Figure 10. Confusion Matrix of Random Forest.

The Random forest predicted all the labels correctly.

Classification Report Random Forest:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	6
2	1.00	1.00	1.00	11
3	1.00	1.00	1.00	13
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	15
accuracy			1.00	72
macro avg	1.00	1.00	1.00	72
weighted avg	1.00	1.00	1.00	72

Figure 11. Classification Report of Random Forest prediction.

The random forest got an accuracy of 1.0

5.3 Decision Trees

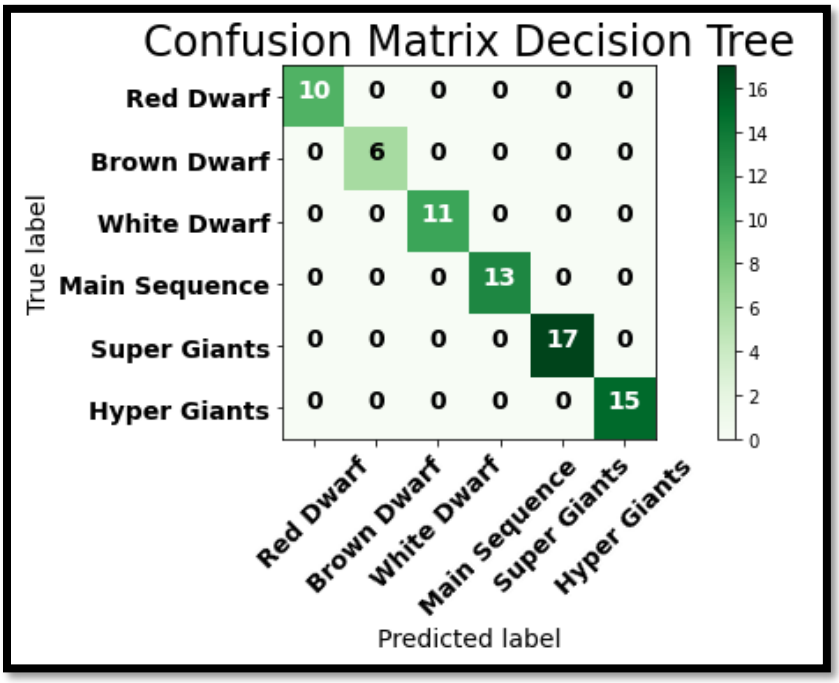

Figure 12. Confusion Matrix of Decision Tree Classifier.

The Decision Tree predicted all the labels correctly.

Classification Report Decision Tree:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	10
1	1.00	1.00	1.00	6
2	1.00	1.00	1.00	11
3	1.00	1.00	1.00	13
4	1.00	1.00	1.00	17
5	1.00	1.00	1.00	15
accuracy			1.00	72
macro avg	1.00	1.00	1.00	72
weighted avg	1.00	1.00	1.00	72

Figure 13. Classification report of Decision Tree.

The Decision Tree got an accuracy of 1.0

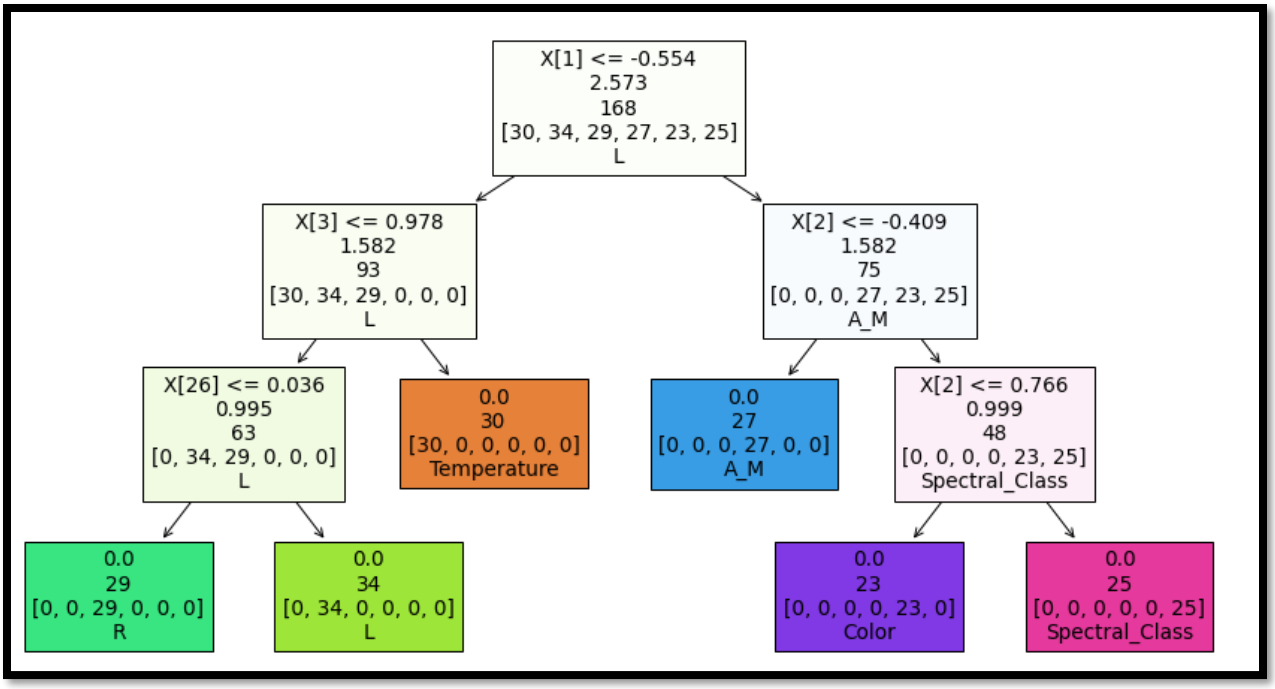

Figure 14. Decision Tree Path way.

6. Conclusion

Finally, we can conclude that both random forest and decision tree classifier predicted correctly with an accuracy of 1.0 and support vector machines got an accuracy of 0.97. We plotted the confusion matrix of all the classifier and saw how many labels were correctly classified.

In order to further improve our accuracy, we can use cross validation method to make sure that our prediction did get lucky. A lot of hyper parameters of the models can be tweaked and the performance can be verified to ensure that our model is not overfitting or under fitting.

7. References

1. Data: https://www.kaggle.com/brsdincer/star-type-classification