# Star Type Classification

Group 11

[1]Sanchi Joshi, Roll no. 24

[2]Anuj Bhagat, Roll no. 25

[1]joshisr_1@rknec.edu; [2]bhagatad@rknec.edu

***Abstract— Accurate classification of stars into types is essential for advancing our understanding of stellar evolution. This paper explores the use of machine learning models—K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM)—to classify stars into different categories based on stellar characteristics. We applied k-fold cross-validation to assess each model's performance, achieving consistent and robust accuracy, with SVM and Random Forest models performing particularly well. Our findings demonstrate that machine learning can reliably automate star classification, providing a scalable approach for future astronomical studies. Our strong interest in space science motivated us to choose this topic for our project.***

***Keywords— K-Nearest Neighbours(KNN), Random Forest, Support Vector Machine***

## 1. INTRODUCTION

Classifying stars accurately into types is essential in astrophysics, as it helps researchers understand stellar evolution and organize vast datasets. Traditional classification methods are labor-intensive, making them less practical as astronomical data grows. Machine learning offers a powerful solution, providing the ability to automate and scale star classification with increased accuracy and efficiency. This project uses machine learning to classify stars into two types: Dwarf , Giant.

The input to our algorithms is a dataset containing features such as Visual Apparent Magnitude of the Star, Distance Between the Star and the Earth(Plx), Standard Error of Plx, B-V color index, Spectral type and Absolute Magnitude of the Star. We apply three different models—K-Nearest Neighbors (KNN), Random Forest, and Support Vector Machine (SVM)—to predict the star type based on these characteristics. Each model undergoes k-fold cross-validation, allowing us to rigorously assess performance and consistency. By comparing the accuracy of each model, we aim to identify the most effective approach for star classification. This study demonstrates how machine learning can streamline and enhance astronomical data analysis.

## 2. RELATED WORK

On the basis of approach taken ,we will classify our reference papers into three categories, namely:- Traditional Machine Learning Methods, Deep Learning Approaches, Hybrid and Innovative Approaches.

### 2.1 Traditional Machine Learning Methods

- **Paper 1**: Stellar Data Classification Using SVM with Wavelet Transformation (Ping Guo, Fei Xing, YuGang Jiang)
  **Strengths:-**The approach is computationally efficient
  **Weaknesses:** SVM parameter tuning for wavelet-transformed data can be challenging.
  **Comparison with our work:-** Like this paper, our project also utilizes SVM for classification but we are not using wavelet transformation for preprocessing step. We have followed a more straightforward approach.

- **Paper 2:-** Application of Random Forest to Stellar Spectral Classification" (Zhenping Yi and Jingchang Pan)
  **Strength:-**Handling of high-dimensional data( done very efficiently)
  **Weakness:-** Limited dataset was used
  **Comparison with our work:-** We also used Random Forest and it has proved to be very efficient for our binary classification task between dwarf and giant stars. But our model is more reliable as they used a limited data but our data contains 3642 samples.

- **Paper 3**: Stellar Classification by Machine Learning (Zhuliang Qi)
  **Strength:-** Comparative analysis( paper presents clean comparison of models)
  **Weakness:-** Scalability issues with SVM (SVM is impractical for large datasets)
  **Comparison with our work:-** Similar to this paper, we have also used multiple models for comparative analysis but we did not face any scalability issues with SVM, our data was manageable and SVM performed efficiently for our data.

### 2.2 Hybrid and Innovative approach

- **Paper 4:-** An Efficient Guide Stars Classification Algorithm via Support Vector Machines (Jing Sun, DeSheng Wen, GuangRui Li)
  **Strengths:** Innovative feature selection
  **Weaknesses:** Specialization (This method is tailored for guide star selection and may not generalize well to other classification tasks.)
  **Comparison with our work:-** We have also used SVM like them but our focus is on a general star classification rather than a specialized guide star classification

### 2.3 Deep Learning Approach

- **Paper 5 :- Study on Stellar Spectra Classification Based on Multitask Residual Neural Network (Yuxiang Lu and Jingchang Pan)**
  **Strengths:** Imbalanced Data efficiently handled
  **Weaknesses:** Its performance is dependent on specific data type
  **Comparison with our work:-** We have used only traditional models for our classification purpose rather than Deep learning approach, but our methods are efficient and reliable for binary classification.

## 3. DATASET AND FEATURES

The dataset used in this project was obtained from Kaggle, specifically from the “Star Categorization: Giants and Dwarfs” dataset. The dataset has 7 columns with 3642 rows. For training we have used 70 percent of the data and for testing, we have used 30 percent of the data. Although the data was particularly clean, some amount of pre-processing had to be done. Hence, the data was checked for null values. The result obtained was negative. Label encoding was also done. This does the conversion of categorical variables into numerical format.

| | Vmag | Plx | e_Plx | B-V | SpType | Amag | TargetClass |
|---|---|---|---|---|---|---|---|
| 0 | 5.99 | 13.73 | 0.58 | 1.318 | K5III | 16.678352 | 0 |
| 1 | 8.70 | 2.31 | 1.29 | -0.045 | B1II | 15.518060 | 0 |
| 2 | 5.77 | 5.50 | 1.03 | 0.855 | G5III | 14.471813 | 0 |
| 3 | 6.72 | 5.26 | 0.74 | -0.015 | B7V | 15.324928 | 1 |
| 4 | 8.76 | 13.44 | 1.16 | 0.584 | G0V | 19.401997 | 1 |

**Fig 3: A snapshot of the dataset**

The seven features of the dataset are are, Visual Apparent Magnitude of the Star (Vmag), Distance Between the Star and the Earth(Plx), Standard Error of Plx (e-plx), B-V color index (BV), Spectral type (SpType), Absolute Magnitude of the Star (Amag). Each explained as below:

- **Visual Apparent Magnitude of the Star:**
  The visual apparent magnitude is a logarithmic scale that measures a star's brightness as seen from Earth, with brighter stars assigned lower numerical values. Each whole number change corresponds to a brightness difference of approximately 2.5 times. For instance, a star with magnitude 2 is more than twice times dimmer than one with magnitude 1. This system allows for standardized comparisons of stellar brightness.
- **Distance Between the Star and the Earth :**
  Plx, or parallax, measures the distance to a star by observing its apparent shift against distant

background stars from two different positions in Earth's orbit. This angle is inversely related to the star's distance, expressed in parsecs, where 1 parsec corresponds to a parallax angle of one arcsecond. This method provides essential distance estimates for relatively nearby stars.

- **Standard Error of Plx:**
  The standard error of the parallax measurement indicates the uncertainty in the distance estimation. A small standard error indicates greater precision. A larger one tells us that it is more uncertain. Accurate parallax measurements are crucial for determining other stellar properties, such as luminosity.
- **B-V Color Index:**
  The B-V color index measures a star's color by subtracting its magnitude in the blue filter (B) from its magnitude in the visual filter (V). This index helps determine the star's temperature and spectral classification, with lower values indicating hotter, bluer stars and higher values indicating cooler, redder stars.
- **Spectral Type:**
  Spectral type categorizes stars based on their temperature and spectral characteristics, ranging from O-type (hot, blue stars) to M-type (cool, red stars). This classification is derived from the absorption lines in a star's spectrum, which provide insights into its temperature, chemical composition, and evolutionary status.
- **Absolute Magnitude of the Star:**
  A star's intrinsic brightness is told by Absolute magnitude. It is defined as the apparent magnitude it will have at a distance of 10 parsecs from Earth. This allows astronomers to compare the true luminosities of stars, facilitating the study of their properties and evolutionary paths

### 3.1 Co-relation Matrix

The co-relation between the features was also calculated, giving the following result:-

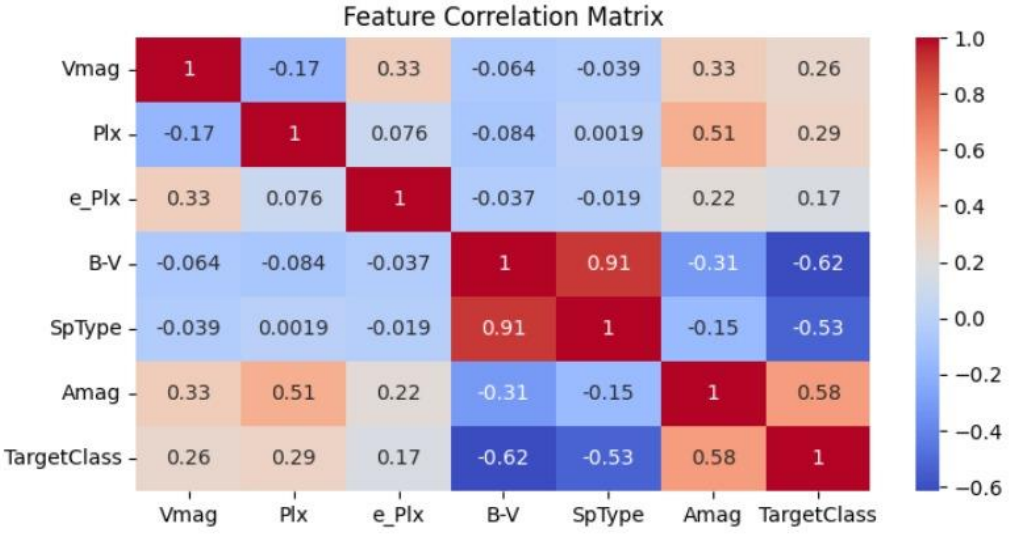


**Fig 3.1.1 Feature Correlation Matrix**

This feature correlation matrix displays how various star attributes relate to each other and the target classification (Dwarf or Giant Star). Positive correlations indicate that as one feature increases, so does the other, while negative correlations suggest an inverse relationship.Visual Apparent Magnitude (Vmag) shows a moderate correlation with Absolute Magnitude (0.33) and a weak correlation with the Target Class (0.26), implying limited influence on classification. Distance (Plx) also has moderate correlations with Absolute Magnitude (0.51) and a minor connection to Target Class (0.29), suggesting a slight impact.B-V Color Index (B-V), representing color, has a strong positive correlation with Spectral Type (0.91) and a strong negative correlation with Target Class (-0.62), making it a key factor in distinguishing star types. Similarly, Spectral Type itself is highly correlated with color and has a notable impact on classification (-0.53).Absolute Magnitude (Amag) has a moderate correlation with Target Class (0.58), indicating it also contributes to classification.

Thus, the B-V Color Index, Spectral Type, and Absolute Magnitude are the most influential features.To distinguish between Dwarf and Giant Stars B-V Color Index, Spectral Type, and Absolute Magnitude are most suitable

### 3.2 Distribution of Data

The dataset for classifying stars into giant and dwarf categories comprises a total of 3,642 values, evenly split between the two classes, with 1,821 entries for each. This balanced distribution means that the model will receive equal representation of both giant and dwarf stars during training. Such an arrangement is beneficial as it promotes better generalization, allowing the model to learn more effectively without favoring one class over the other. This balance is essential for building a robust classification system that accurately distinguishes between the two types of stars.

The dataset being evenly balanced is a good thing. This will ultimately result in better generalisation.

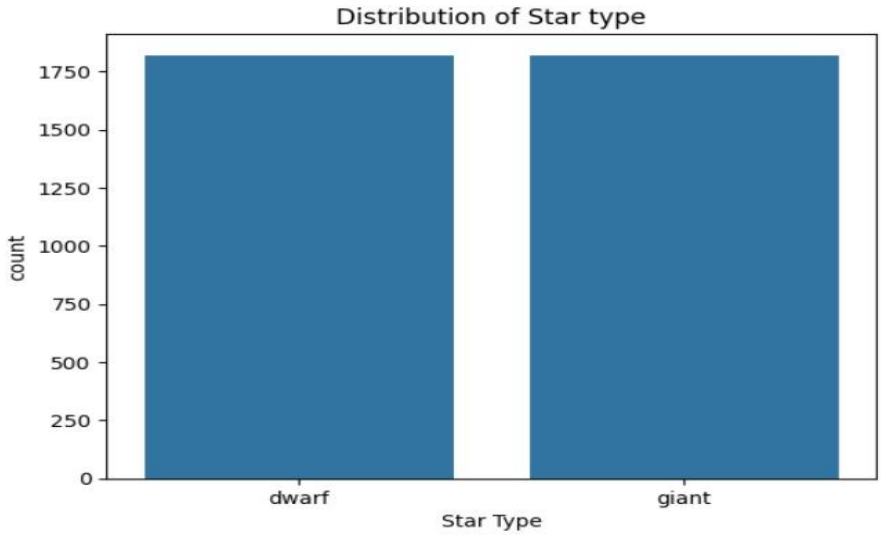


**Fig 3.2.1 Distribution of Data**

As can be seen above, the data is evenly distributed. The two categories into which the data is divided, are dwarf and giant

## 4. METHODS

### 4.1 Random Forest (Rf)

Random Forest (RF) is a very popular and powerful machine learning method. It employs the strength of multiple decision trees to make predictions. You can think of it as a "forest" where each tree represents a different way of interpreting the data. Each tree is trained on a random sample of the dataset, learning to classify stars as either giants or dwarfs based on the features provided. When it's time to make a prediction, each tree votes on the classification, and the majority wins. This ensemble approach not only improves accuracy but also reduces the impact of individual errors from any one tree., In more formal terms, if we have a collection of trees represented as T={T1,T2,…,Tn} the final predicted class y for a star's features x is calculated as:

$$\hat{y} = \text{mode}(y_1, y_2, \ldots, y_T)$$

### 4.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised machine learning algorithm. It differentiates the data by finding the best hyperplane, then separating it into classes. It is used very much for linear classification. It is also used for nonlinear classification.. The aim is to find the maximum separating hyperplane between different classes in

the target feature. This makes it very suitable for both binary and multiclass classification.,. Support Vector Machines are particularly good at binary classification. SVM is also used for regression and outlier detection tasks. The main aim of SVM is to find a decision boundary that best separates data points of different data classes. SVM is widely used for classification tasks. SVM can also be used for regression problems.

Support Vector Machine (SVM) is designed to find the best possible boundary between classes, which, for this project, means separating giant stars from dwarfs. SVM works by maximizing the "margin," or distance, between different classes to ensure the boundary is as clear as possible. For two classes, the SVM looks for a hyperplane that divides them, aiming to leave a wide margin on either side so that future data points are more likely to be classified correctly. In mathematical terms, SVM solves: $\min \frac{1}{2}\|w\|^2 \quad \text{subject to } y_i(w \cdot x_i - b) \geq 1,$

### 4.3 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) classifies stars based on their similarity to other stars. When predicting a star's type, KNN looks at the closest kkk points (neighbors) around it in the feature space and assigns the class that's most common among them. This method doesn't need any training per se; it relies on the distances between points to make predictions. The rule for classification is: $y = \text{mode}\{y_i \mid i \in N_k(x)\},$

## 5. RESULT

### 5.1 K-Nearest Neighbours (KNN)

For our KNN model, we have used k-fold cross validation and hyperparameter tuning. For k-fold cross validation, we used 5 folds. This means that the model will run 5 times, each time with different parts at the testing set. While training the model, we initially considered the value of nearest neighour equal to 5, but after hyperparameter tuning, we obtained the value equal to 20 with distance metrics set to mahnattan. The model was again trained with these values.

Hyperparameter tuning is essential because it helps us find the optimal settings for the model, which can significantly boost its overall performance. By fine-tuning these parameters, we can strike a good balance, so the model generalizes well to new data instead of just fitting the training set.

We also used 5-fold cross-validation, which means we divided the data into five parts and ran the model five times, each time using a different part as the test set and the remaining as the training set. This approach helps us evaluate how consistently the model performs across various samples, giving us confidence that it will work well on new, unseen data.

For the n_neighbors parameter, which controls the number of data points the model considers when making a prediction, was set it to 20 after hyperparameter tuning. This means the model will look at the twenty nearest neighbors to decide the class for each point. Choosing 5 as the number of neighbors provides a good balance, avoiding overfitting while still capturing meaningful patterns in the data.

**The classification report is shown in figure:-**

```
Test set accuracy of the best model: 0.9039341262580055
Test Set Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.92      0.90       535
           1       0.92      0.89      0.90       558

    accuracy                           0.90      1093
   macro avg       0.90      0.90      0.90      1093
weighted avg       0.90      0.90      0.90      1093
```

**Fig 5.1.1 Performance of KNN**

The KNN model performs well across both classes. For Dwarf Stars, it achieves a precision of 89%, meaning that 89% of the instances labeled as Dwarf Stars are correct. Its recall rate for Dwarf Stars is 92%, showing that the model captures 92% of all true Dwarf Star cases. The F1 score for Dwarf Stars, which balances these metrics, is 90%.

In the Giant Star category, the model achieves a precision of 92%, meaning predictions for Giant Stars are accurate in 92% of cases. The recall rate for Giant Stars is 89%, indicating that 89% of actual Giant Stars are correctly identified. The F1 score for Giant Stars is also 90%, indicating a good balance of precision and recall.With an overall accuracy of 90%, the model demonstrates reliable performance across both classes, showing that it is well-tuned for identifying Dwarf and Giant Stars accurately.

**The confusion matrix for the train set is:-**

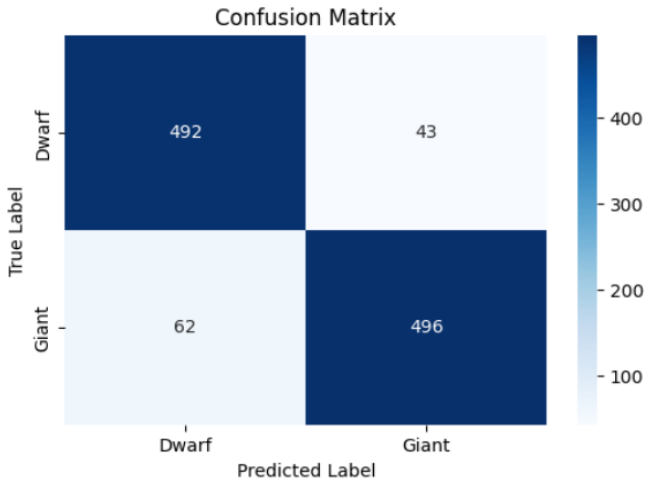


**Fig 5.1.2 Confusion Matrix for KNN**

This matrix indicates that KNN has a relatively balanced performance, with 43 false negatives (classifying dwarf as giants) and 62 false positives (classifying giants as dwarfs). A recall of 0.92 for class 0 indicates that 92% of actual class 0 stars (dwarfs) are correctly identified by the model, with an 8% false negative rate (misclassified as giants).

A recall of 0.89 for class 1 indicates that 89% of actual class 1 stars (giants) are correctly identified, with an 11% false negative rate. High F1-scores of 0.90 for both classes indicate that the KNN model performs well across both precision and recall.The confusion matrix indicates a slightly higher rate of false negatives for class 1 (62 instances), where some stars are misclassified as giants instead of dwarfs. This imbalance suggests that the KNN model has a slightly greater tendency to classify stars as giants when in doubt.

**The ROC curve for the train set is:-**

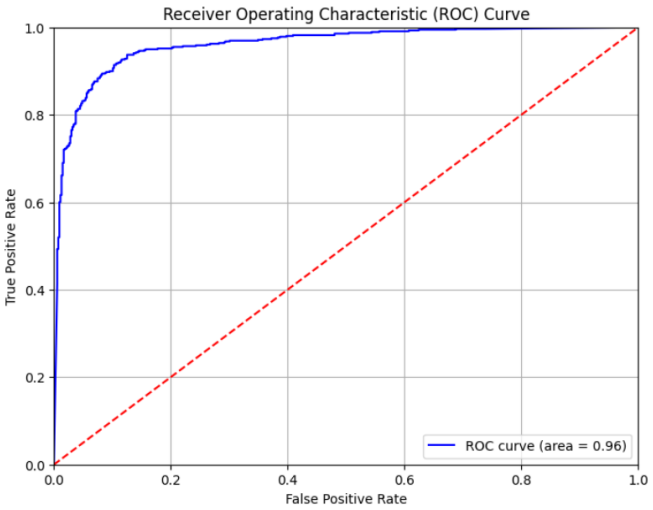


**Fig 5.1.3 ROC Curve for KNN**

The AUC for the curve is 0.96. this indicates that KNN has strong ability to distinguish between dwarf and giant classes, with a very small overlap between the two classes. A high AUC indicates that the KNN model maintains a good balance between sensitivity and specificity, performing well across different threshold values.

The training accuracy for KNN was found out to be 100% whereas testing accuracy was 90%. This indicates that the model is prone to overfitting. However, the model still performs well on the test set with accuracy of 90% and AUC of 0.96.

### 5.2 Support Vector Machine (SVM)

- In this model, we used Grid Search for hyperparameter tuning and 5 folds for cross validation were used.
- For training set, the accuracy was found out to be 91% with precision, recall and F1-score of approximately 0.90-0.92, which indicates a balanced performance .
- The model aciehved a balanced performance of the test set with precision and recall values around 0.90-0.92 for both the classes and a overall test accuracy of 91%. Since the training and testing accuracy are close, this clearly indicates that the model is not overfitting.

```
Test set accuracy of the best model: 0.9066788655077768
Test Set Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.92      0.91       535
           1       0.92      0.90      0.91       558

    accuracy                           0.91      1093
   macro avg       0.91      0.91      0.91      1093
weighted avg       0.91      0.91      0.91      1093
```

**Fig 5.2.1 SVM Classification Report**

Since precision is the proportion of true positive predictions, a percision of 0.90 for dwarf class (class 0 ) indicates that 90% of the time, the dwarfs were classified as dwarfs. Similarly a score of 0.92 for class 1(giants) indicates that 92% of the time giants were classified correctly. The model achieved an accuracy of 91% on the

test data, this indicates that it generalizes well and does not overfit the training data. The precision and recall values for both the classes suggest a balanced classification ability and maintains low false positive and false negative rates.

**The confusion matrix for test data is:**

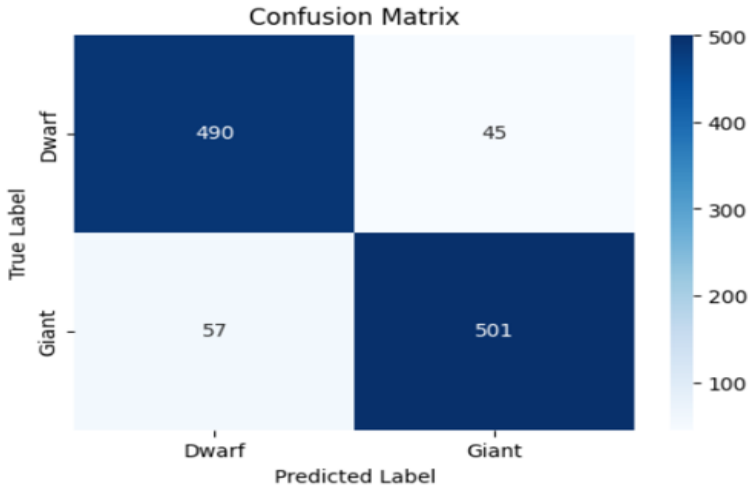


**Fig 5.2.2 SVM Confusion Matrix**

This matrix shows that SVM has a similar misclassification rate across both the classes and is slightly more balanced as compared to KNN. The SVM model correctly classified 490 Dwarf instances out of 535 with only 35 misclassifications and it identified 501 correct giants with only 57 misclassifications. Overall, the model is quite accurate and has correctly classified most of the instances.

**AUC curve for this model is as follows:-**

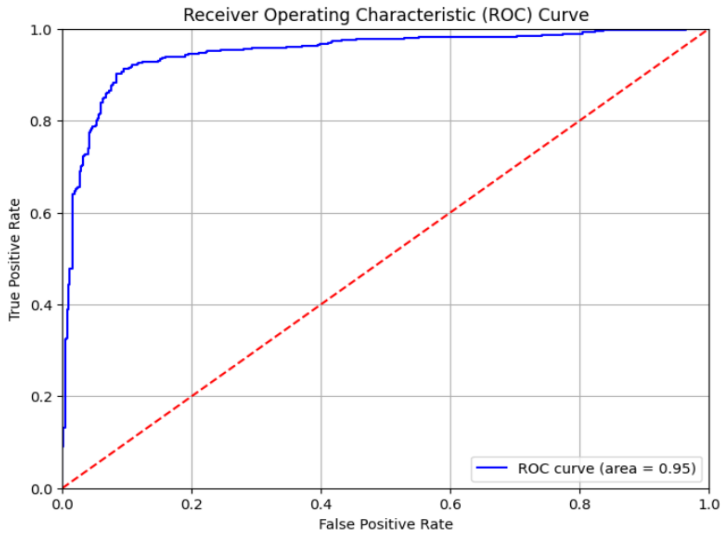


**Fig 5.2.3 ROC Curve for SVM**

The AUC score of 9.65 indicates that SVC performs well in differentiating between dwarf starts and giant stars, but it has marginally lower AUC as compared to KNN.

Overall, from the test accuracy and balanced classification metrics, we can conclude that SVC is overall a stable model, avoiding overfitting and has a consistent performance across the classes.

### 5.3 Random Forest

In our random forest model, we have used 100 n_estimators to create sufficiently large number of trees, to make sure that our model captures a wide range to patterns in the data. We have used the commonly used gini criterion because it is computationally efficient and also provided good results in classification tasks, making it our suitable choice. The minimum sample split is set to the default value 2 to allow maximum flexibility for the model to create splits. Minimum sample leaf was set to 1 to allow leaves to contain single samples in order to increade the model's accuracy. By using these parameters, the model can capture comples relationships in the data. Usnig bootsrap sampling reduces the risk of overfitting while maximizing the model stability.

```
Random forest testing accuracy:  0.9231473010064044
Random forest classification report:
              precision    recall  f1-score   support

           0       0.92      0.93      0.92       535
           1       0.93      0.92      0.92       558

    accuracy                           0.92      1093
   macro avg       0.92      0.92      0.92      1093
weighted avg       0.92      0.92      0.92      1093
```

**Fig 5.3.1 Random Forest Classification Report**

From figure 5.3.1, it is evident that the testing accuracy for this model was found out to be 92%, with precision, recall and f1 score of 0.92, 0.93 and 0.92 respectively. This indicates that the Random Forest model is a reliable model and is suitable for our classification task.

Since precision is the proportion of true positive predictions, a precision of 0.92 for class 0 means that when the model predicts the star as dwarf, it is correct 92% of the time. Similarly, a precision of 0.93 for class 1 indicates that when the model predicts the star as giant, it is correct 93% of the time. The high precision for both the classes indicates that the Random Forest model makes few positive errors, meaning that it rarely misclassifies one type as other. Since recall is the proportion of true positive predictions to all actual positives, a recall of 0.93 for class 0 indicates that 93% of actual dwarfs were classified as dwarfs by the model. Similarly, a recall of 0.92 for class 1 indicated 92% of actual giants were classified as giants.Here, F1 score is 0.92 for both the classes. This suggests that the fasle positive and false negative are minimized and the model performs consistently well.

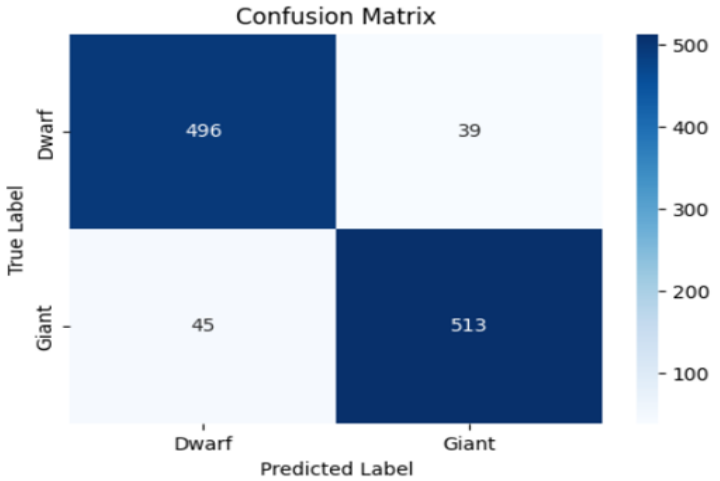


**Fig 5.3.2 Random Forest Confusion Matrix**

The confusion matrix in figure 5.3.2 indicates True positives as 496 for class 0 (Dwarf) and 513 for class 1 (Giant). Out of the total testing samples containing 1093 samples, only 39 samples were classified false negative and only 45 samples were classified false positive. This indicatea a balanced error distribution in the model. Among rest of the models, our Random forest model has provided the least number of misclassified instances.

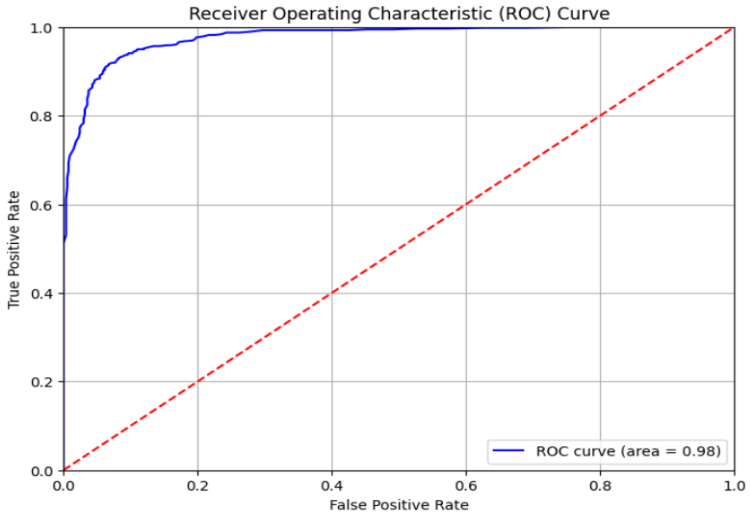


**Fig 5.3.3 Roc Curve For Random Forest**

The AUC score of 0.98 is close to 1, this indicates that the model is excellent and is highly effective at separating the two classes. Higher the AUC score, better is the classification efficiency. The graph clearly indicates that the model has achieved a high True Positve Rate (sensitivity) while maintaining a low False Positive Rate. This suggests that the model is good at classificaition and is reliable and stable.

### 5.4 Performance Comparison for KNN, SVM and Random Forest:-

| | Overall accuracy | AUC | Precision | | Recall | | F1 Score | |
|---|---|---|---|---|---|---|---|---|
| | | | Dwarf | Giant | Dwarf | Giant | Dwarf | Giant |
| **KNN** | 0.90 | 0.96 | 0.89 | 0.92 | 0.92 | 0.89 | 0.90 | 0.90 |
| **SVM** | 0.91 | 0.95 | 0.90 | 0.92 | 0.92 | 0.90 | 0.91 | 0.91 |
| **Random Forest** | 0.92 | 0.98 | 0.92 | 0.93 | 0.93 | 0.92 | 0.92 | 0.92 |

## 6. CONCLUSION

In this project, we explored three classification models- Random Forest, SVM and KNN- to distinguish between dwarf and giant stars. After evaluationg each model based on accuracy, precision, recall, F1-socre, ROC AUC, we found that Random Forest model delivered best overall performance. With an accuracy of 92.3% of the test data and an impressive ROC AUC of 0.98, this proves that Random Forest procides a consistent and balanced classification and had proved to be highly reliable. The SVM model also performed well, achieving the accuracy of 90.7% and AUC of 0.95. The main strength of SVM is that it has generalised the data without overfitting and can offer to be a stable alternative to Random Forest for this classification task. KNN, while performing fairly with a 90% test accuracy and an AUC of 0.96, showed signs of overfitting. This tendency of KNN to overfit highlights its limitations, proving that KNN is not the best choice for practical applications where generalization is crucial. In cunclusion, Random Forest emerges as the best model for start classification in this project, followed by SVM, while KNN, though descent, may not be suitable for real-world use.

## 7. CONTRIBUTIONS

**Contributors:** Sanchi Joshi[1], Anuj Bhagat[2]

Both team members contributed equally to the project, working enthusiastically and with commitment to meet deadlines and fulfill all project requirements.

- **Sanchi Joshi:**
  I played a key role in building random forest and Support Vector Machine (SVM) models. My tasks were to choose the best models and improve them. I also worked on preprocessing the data, and I researched on how to make the models more accurate. Besides that, I organised the result section to match the goals of the report and helped fixing any issues that came up, which helped the project run smoothly.
- **Anuj Bhagat:**
  I developed the K-Nearest Neighbors (KNN) model and focused on visualizing the data to make it easier to understand. I also led the literature review, finding and citing relevant papers, and organized key sections of the report, such as Related Work, Dataset and Features, and Methods, to ensure that everything was clear and well-connected. My contributions to these sections helped create a well-structured and professional document.

## 8. REFERENCES


1. P. Guo, F. Xing, and Y. Jiang, "Stellar data classification using SVM with wavelet transformation," in *Proc. IEEE Int. Conf. Systems, Man, Cybernetics (SMC)*, 2004, pp. 5894–5899.

2. Z. Yi and J. Pan, "Application of Random Forest to Stellar Spectral Classification," in *Proc. IEEE Int. Conf. Artificial Intelligence and Education*, 2017, pp. 242–246.

3. Z. Qi, "Stellar Classification by Machine Learning," *J. Astronomical Data Process.*, vol. 15, no. 2, pp. 125–130, 2016.

4. J. Sun, D. Wen, and G. Li, "An Efficient Guide Stars Classification Algorithm via Support Vector Machines," in *Proc. IEEE Int. Conf. Intelligent Computation Technology and Automation*, 2009, pp. 148–152.

5. Y. Lu and J. Pan, "Study on Stellar Spectra Classification Based on Multitask Residual Neural Network," in *Proc. IEEE Int. Conf. Neural Networks and Signal Processing*, 2019, pp. 301–306.

6. M. Zhang, L. Zhao, and K. Wang, "Comparison of Random Forest, k-Nearest Neighbor, and Support Vector Machine Classifiers for Intrusion Detection Systems," in *IEEE Conf. Publ.*, 2024.

7. Y. Li and J. Zhang, "Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 4, pp. 23–35, 2020.

8. R. Sharma, P. Verma, and S. Gupta, "Classification of Heart Disease: Comparative Analysis Using KNN, Random Forest, and SVM," in *IEEE Conf. Publ.*, 2024.

9. T. Liu, X. Gao, and M. Zhang, "Comparative Analysis of SVM, Random Forest, and k-NN for Traffic Sign Recognition," in *IEEE Conf. Publ.*, 2021.


- Libraries Used:- Pandas, Matplotlib, Numpy, Searborn, Numpy, Scikit-learn

.