STELLAR CLASSIFICATION DATASET STATISTICS, ANALYSIS AND PREDICTION

Ananya Wadhwa
Department of Computer Science & Engineering
Manipal University Jaipur
Email: ananya28wadhwa@gmail.com

Abstract –
Stellar classification plays a fundamental role in astronomy and cosmic ecosystem, improving the comprehension of the composition and structure of the universe. This research paper delves into the imperative role of stellar classification in unraveling the mysteries of celestial bodies such as stars , galaxy or quasar. The classification of stars is essential for discerning the impact of various astrophysical phenomena, including stellar evolution, gravitational interactions, and cosmic dynamics .This paper uses instances from the Sloan Digital Sky Survey Data Release 17 (SDSS DR17) to classify as stars, galaxies, or quasars using various machine learning algorithm.

The classification was made through supervised learning. A variety of machine learning models, such as Random Forest, XGBoost , K-Nearest Neighbors , Naïve Bayes Classifier, Decision Trees, and Logistic Regression, were developed. Random Forest performed the best with 98% accuracy and correctly classified all instances labelled as stars in the dataset. The worst-performing algorithm was Naïve Bayes, with 60% accuracy. The evaluation is helpful in understanding the results of the proposed stellar classification scheme and exploring its potential improvements in the future. This model will be a successful technique to analyze and predict the categories for classification.

I. INTRODUCTION

Stars are made up of hydrogen and helium, the building blocks of galaxies [1]. Galaxies are also made of gas and dust; there are numerous galaxies in the universe that scientists cannot count [2]. Quasars are found in some large galaxies with supermassive black holes at their centers and are considered active galaxies themselves. Quasars make up five to ten percent of large galaxies [3].Based on all these observations we form them in a category of Stellar Classification.Stellar classification is the classification of stars based on their spectral characteristics .Using a prism or diffraction grating to split the star's electromagnetic radiation into a spectrum that displays a rainbow of colors interspersed with spectral lines, the radiation can be studied[6]. Each line indicates a particular chemical element or molecule, with the line strength indicating the abundance of that element. The strengths of the different spectral lines vary mainly due to the temperature of the photosphere, although in some cases there are true abundance differences. The identification of eclipsing binary stars and the determination of stellar spectral types marked the beginning of the measurement of stellar masses. By the middle of the 20th century, it became apparent that stars have masses that vary greatly, starting at roughly 10% of the Sun's mass.

Stellar bodies are the celestial butterflies of the cosmic expanse, which play a major role in the intricate fabric of our universe. Stars constitute a significant portion of the cosmic order and are therefore essential to astrophysics. Stars are made up of about 20,400 species, or 9 percent of the world's order, and they have an extensive range of characteristics, such as luminosity, temperature, and spectral features.

The primary criteria for stellar classification include temperature, luminosity, spectral characteristics, and size. The most commonly used system for stellar classification is the spectral classification, which

is based on the star's spectrum obtained through spectroscopy. The star's temperature and other properties are indicated by the spectral classification, which is represented by a letter that is frequently followed by a number. O, B, A, F, G, K, and M are the stars in order of hottest to coolest; O-type stars are the hottest and M-type stars are the coolest.
A further essential component of star classification is the luminosity class, which is represented by Roman numerals and describes the brightness and size of a star. I (supergiants), II (bright giants), III (giants), IV (subgiants), and V (main-sequence or dwarf stars) are the luminosity classes.

The Hertzsprung-Russell (H-R) diagram is a graphical representation of stellar classification, plotting stars' luminosity against their temperature or spectral type. Astronomers can learn a lot about the life cycle devolutionary stages of stars by using this diagram.[5]

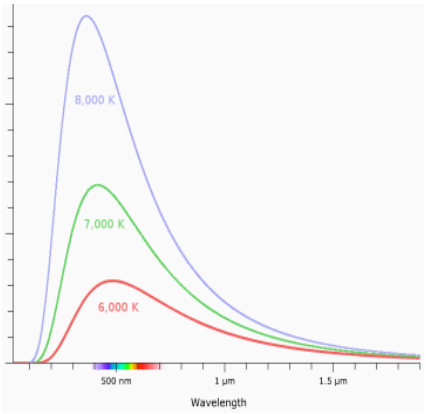


Fig. Hertzsprung-Russell (H-R) diagram

Stellar classification provides astronomers with a systematic framework for understanding the vast diversity of stars in the universe and serves as a foundation for various astrophysical studies and observations. This paper addresses the creative actions made to transform the conventional approaches to star identification. Traditional methods rely on complex analyses of luminosity and spectral features, which are laborious, time-consuming, and resource-intensive procedures, as opposed to different supervised machine learning algorithms used in this research are quick and aid in classification.
In this study, we (i) aim the Sloan Digital Sky Survey Data Release 17 (SDSS DR17) (ii) propose to use various machine learning algorithms(iii) compare the results of top-1, top-3, top-5 accuracies, training loss, and validation loss of trained architectures on the built SDSS DR17 dataset to design an accurate Stellar classification system. Random forest is used to model the relationship between different characterstics and predicted density of different class in the dataset

II. RELATED WORK

In 1802, Dark lines in the spectrum of sunlight were noticed by William Wollaston in 1802, indicating that there were naturally occurring borders between colors. Joseph Fraunhofer, in 1814, identified around 600 dark lines in the solar spectrum, measuring the wavelength of 324. The elemental compositions of stars were discovered in 1864 when he compared these lines in the spectra of other stars to materials found on Earth. This laid the foundation for modern spectroscopy.
Prior to the discovery of spectra, scientists looked for ways to classify stars Spectral observations in d distinct patterns, leading to a powerful classification tool. The Harvard Observatory, early 20th century, refined the spectral classification. Henry Draper's work, initiated in 1872, continued by Annie Jump Cannon, classified stars based on temperature-sensitive spectral lines. The scheme, published in the Henry Draper Catalogue (HD) and Henry Draper Extension (HDE), categorized 225,000 stars by temperature-related lines, emphasizing hydrogen Balmer lines, helium lines, iron lines, and other key features.[8]

The paper "Stellar spectral classification and feature evaluation based on a random forest" by Xiang-Ru Li says that the current problems with automatic stellar spectrum classification, with a special emphasis on spectra from instruments such as LAMOST that do not have absolute flux calibration. When used with real spectral data, the suggested scheme based on Random Forest (RF) performs better than current approaches. Most notably, the study thoroughly investigates how classification performance is affected by flux normalization and continuum normalization. In addition, the study investigates the assessment of spectral features, providing insight into plausible physical interpretations and supporting the development of more efficient classification schemes. The results further improve the current state of automatic stellar spectral classification.

The article in "Encyclopedia of Spectroscopy and Spectrometry,1999" says that utilizing Doppler shifts to provide radial motion information. Details regarding stellar outflows in a variety of star types, such as carbon stars, emission line stars, K and M giants, are revealed by high-resolution spectra. The velocity data obtained from stellar spectra turns into an effective instrument for examining the star dynamics inside galaxies.In modern times, methods like passing stellar light through iodine gas or using stable, fiber-fed spectrometers with simultaneous wavelength calibration are used to search for planets orbiting solar-type stars using Doppler shifts. Using cross-correlation techniques, it is possible to measure the radial velocities of stars with an accuracy of approximately 15 m s−1. This allows for the discovery of planetary companions with masses ranging from 0.5 to 10 Jupiter masses and orbits similar to our solar system. The 'Copernican' revolution is completed with this revolutionary capability in stellar spectroscopy, which broadens the knowledge of planetary systems and celestial dynamics.

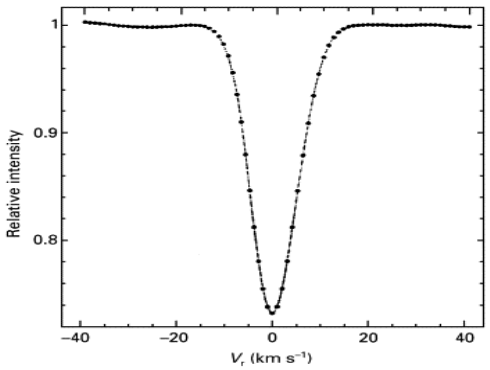


Fig. Copernican revolution graph

III. DATA SET MATERIALS

In this section, we introduce our the Sloan Digital Sky Survey Data Release 17 (SDSS DR17) to classify as stars, galaxies, or quasars using a machine learning algorithm .The proposed benchmark has 100000 data rows files .Every data point is described by 17 feature columns and 1 class column which identifies it to be either a star, galaxy, or quasar [1].Object Identifier, Right Ascension angle, Declination angle, Ultravoilet filter, Green filter, Red filter, Rerun number, Field number, Redshift value, Fiber ID, etc. are a few of the features columns in the data. Some features in the dataset are significantly useful such as navigation angles, ascension ,declination and filters of the photometric system — u, g, r, i, z, redshift.

Ascension and declination :These are important terms in astronomy and navigation in space. In the celestial sphere, ascension indicates an object's position to the left or right, and declination indicates its position to the up or down [4].

Photometry: The measurement of light intensity that is visible to the human eye is known as photometry. Ultraviolet, Blue, and Visual, or UBV photometric system, also referred to as the Johnson system, is a photometric system that is typically used to categorize stars based on their color. The system was the first to be standardized in photometry [5, 6].

Redshift: The term refers to the literal stretching of a light's wavelength, which causes the light to be perceived as "shifted" toward the red portion of the spectrum. It shows us how a star, planet, or galaxy is moving in relation to us when it is in space. It makes it possible for astronomers to calculate the distances to the oldest and most distant objects in our universe [7].

IV. EXPLORATORY DATA ANALYSIS

Stellar classification on the Sloan Digital Sky Survey Data Release 17 (SDSS DR17) to classify as stars, galaxies, or quasars. During the analysis phase, the focus is on identifying key features and parameters that significantly impact the accuracy of the classification model. These features may include Ascension angle , declination angle, photometry(u, g, r, i, z) and Redshift characteristics inherent in the dataset. Understanding the influence of these factors is vital for refining the model's capability to accurately classify the stars, galaxy or quasars.

Challenges and biases present within the dataset are crucial considerations in the analysis. The inclusion of various features representing Run Number, a Rerun Number , field numbers , Unique ID ,etc, the visual cues, distinguishing these number may exhibit subtle variations in the classifications. Additionally, challenges may arise due to some null values in the dataset and also due to some large variations in class column of the dataset. Identifying and addressing these challenges is essential for ensuring the robustness and generalizability of the classification models.
Biases within the dataset, whether due to overrepresentation or underrepresentation of specific categories, can significantly impact the model's performance.

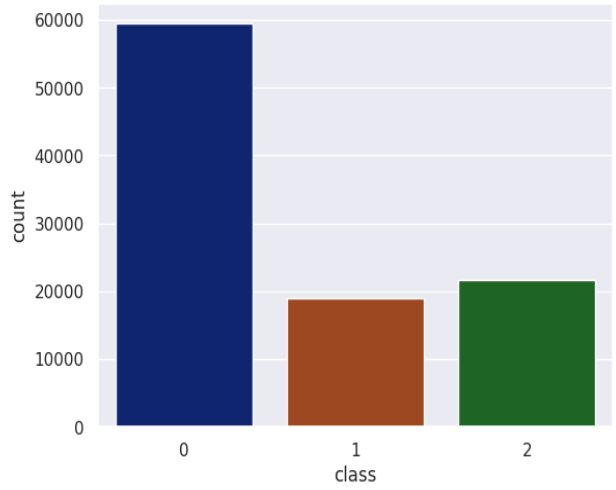


Fig .Class distribution and proportion

By thoroughly scrutinizing the datasets, analyzing key features, and addressing challenges and biases, this research aims to improve the effectiveness and fairness of the classification models. This contributes to the broader objectives of achieving accurate Stellar classification.

V. DATA VISUALISATION

Data visualization is presentation of information in a graphical format and people understand data , features and interpret patterns, trends in a visual form. It helps to understand different spectral features that provide insights in to the characterstics and helps to select relevant features for the model. The univariate analysis takes individual data features to visualise the data.

Distribution plot: This is often used in Seaborn's in python and is used to visualize the distribution of univariate dataset. It help to identify outliers and extreme values in the dataset. Skewness (asymmetry) and kurtosis of the distribution can be visually assessed.It provides a smooth curve that identifies the underlying probability of the density functions of the data with delta , redshift and plate,etc.

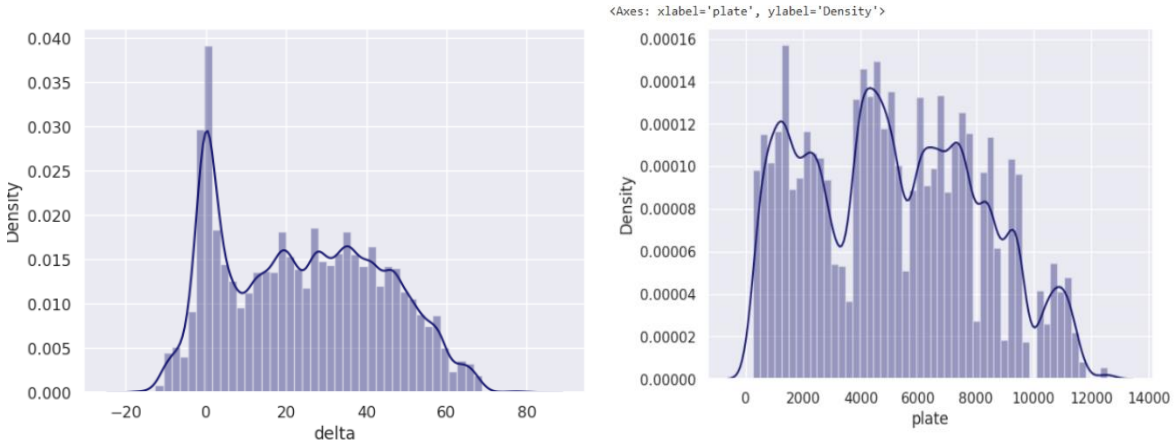


Fig. Visualization of Distribution Plot of delta and plate with density

Histogram: This is often used in Seaborn's in python and is used to visualize the distribution of univariate dataset. It helps to understand the frequency distribution of the data using bins. The different shapes of the graph like bell-shaped, multimodal helps to identify the nature of the graph

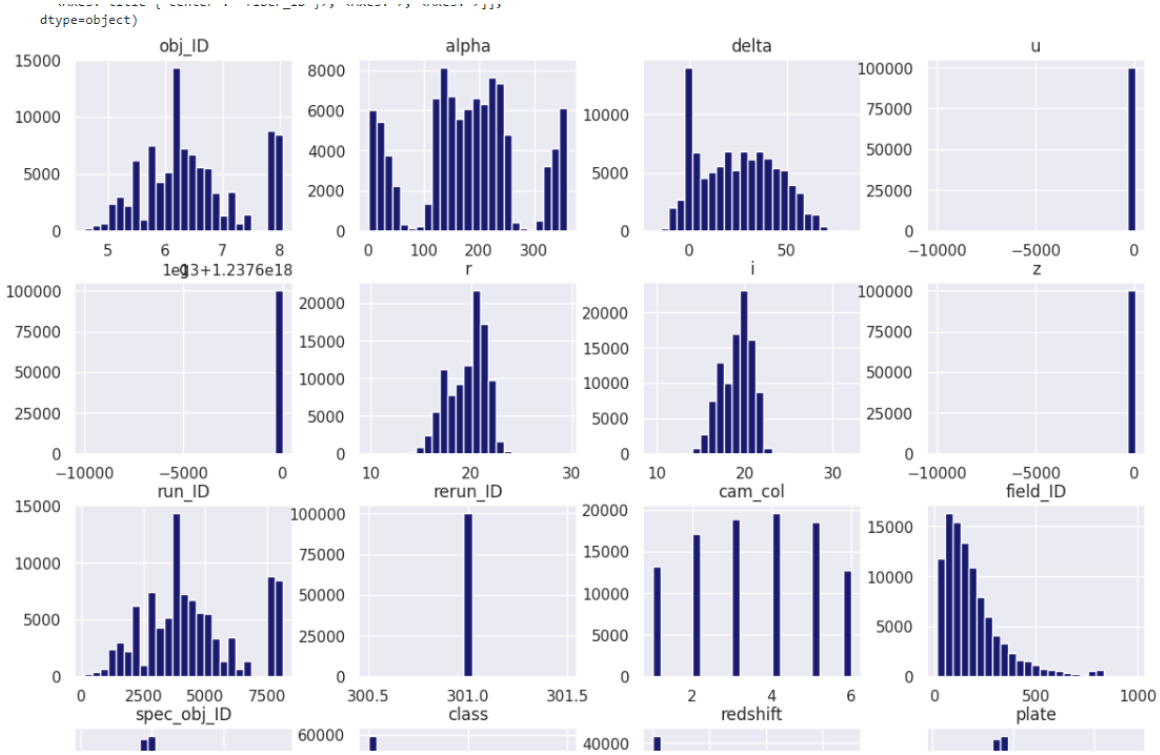


Fig. Visualization of Histogram with distribution of data

Scatter plot : This is often used in Seaborn's in python and is used to visualize the data points on two dimensional graph. Each point on graph represents two values of two variables . They are usefull for assessing the relationships and patterns between two continuous variables like detla and alpha.

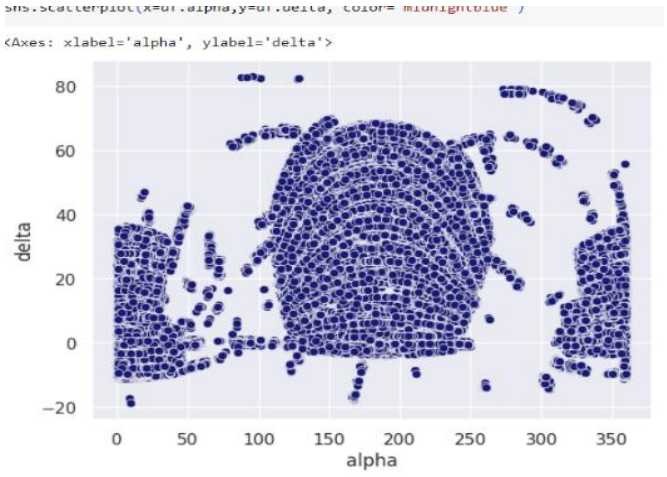


Fig. Visualization of Scatter Plot of delta and alpha

HeatMap: A heatmap in python is useful in creating correlation matrix for the variables in the dataframes. This is oftenly used to explore the relationships between different variables

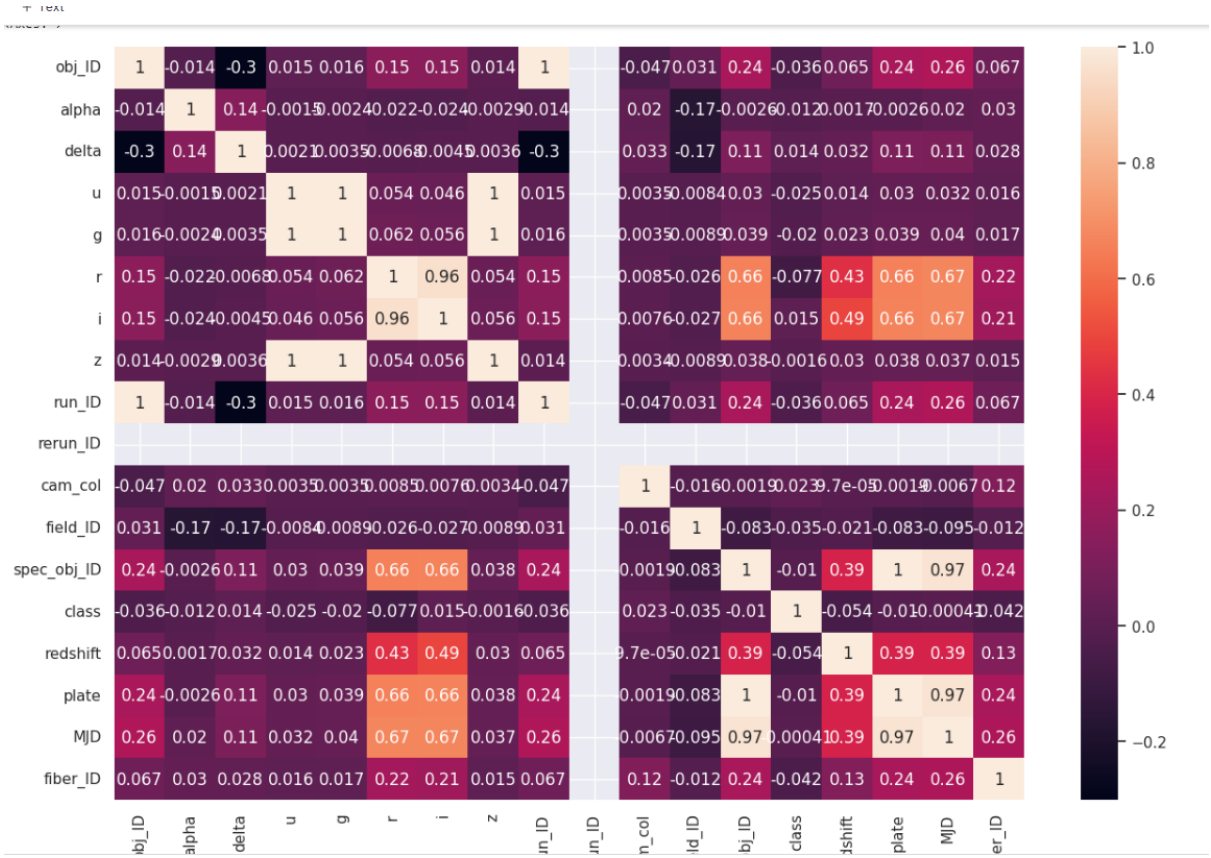


Fig. Visualization of heatmap with different variables

Thus , these different types of visualization on various features of the dataset help us to undersand various trends , characterstics of the distribution and help us to further perform the training and analysis.

VI. METHODS

1. Logistic Regression

Logistic regression is a supervised machine learning algorithm mainly used for binary classification where we use a logistic function, also known as a sigmoid function that takes input as independent variables and produces a probability value between 0 and 1.This statistical model also reffered as logit model is often used for classification and predictive analytics. The independent variable in this logistic regression equation is x, and the dependent or response variable is logit(pi). In most cases, maximum likelihood estimation (MLE) is used to estimate the beta parameter, or coefficient, in this model. The log likelihood function is the result of each of these iterations, and logistic regression aims to maximize this function in order to determine the optimal parameter estimate. In the model ,we implement LR model using scikit-learn in Python, trained it on training data and made predictions on test data and calculated recall score for weighted average. Thus the recall value calculated came out to be **0.943567**

2. Gaussian Naïve Bayes:

A machine learning classification method called Gaussian Naive Bayes (GNB) is based on a probabilistic methodology and a Gaussian (normal) distribution. It is predicated on the idea that every parameter also referred to as features or predictors has the ability to independently predict the output variable. It may predict the probability that a dependent variable will fall into each group. It helps in

identifying extreme values or outliers in the distribution. The features don't depend on the class variable. The algorithm calculates the mean and standard deviation of every feature for every class. It is extensively used in domains like spam filtering, medical diagnosis, and text classification. In the model ,we implement LR model using scikit-learn in Python, trained it on training data and made predictions on test data and calculated recall score for weighted average. Thus the recall value calculated came out to be **0.604185**

3.Random Forest:

Random forest is a most commonly-used machine learning algorithm trademarked by Leo Breiman and Adele Cutler, which combines the output of multiple decision trees to reach a single result. The random forest algorithm is an extension of the bagging method as it utilizes both bagging and feature randomness to create an uncorrelated forest of decision trees. Low correlation between decision trees is ensured by feature randomness, commonly referred to as "the random subspace method" or feature bagging. It creates a random subset of features. Prior to training, the three primary hyperparameters must be set for the training consisting of the size of the nodes, the count of trees, and the quantity of features sampled. Regression and classification issues can then be resolved using the random forest classifier. In the analysis, the loop is used train different estimators from 1-20 and is trained , and displays how the recall score changes with different estimators. Thus the recall value calculated came out to be **0.982011.**

	Estimators	Recall score
0	1.0	0.966602
1	2.0	0.968374
2	3.0	0.977772
3	4.0	0.978221
4	5.0	0.979836
5	6.0	0.979970
6	7.0	0.980666
7	8.0	0.980688
8	9.0	0.981069
9	10.0	0.981316
10	11.0	0.981406
11	12.0	0.981451
12	13.0	0.981720
13	14.0	0.981810
14	15.0	0.981810
15	16.0	0.981899
16	17.0	0.981742
17	18.0	0.982011
18	19.0	0.981810
19	20.0	0.981967

Fig. Estimators table ranging from 1-20

4. XGBoost:

XGBoost stands for “Extreme Gradient Boosting”is an optimized distributed gradient boosting library designed for efficient and scalable training of machine learning models. It is an ensemble learning technique that generates a stronger prediction by combining the predictions of several weak models.its fundamental capacity to operate with large data sets and provide cutting-edge results in numerous machine learning applications, including regression and classification.Because of its effective handling of missing values, it can handle missing values in real-world data without requiring a lot of pre-processing. In addition, XGBoost comes with built-in support for parallel processing, which enables training models on big datasets quickly.It facilitates cross-validation as well.It uses regularization strategies to prevent overfitting. . In the model ,we implement LR model using scikit-learn in Python, trained it on training data and made predictions on test data and calculated recall score for weighted average. Thus the recall value calculated came out to be **0.980234**

5. K-Nearest Neighbours:

We implemented a machine learning pipeline for feature extraction using a pre-trained neural network and subsequently applied the k-nearest neighbours (KNN) algorithm for classification. Initially, an intermediate layer model is created to extract features from the penultimate layer of a pre-existing neural network (denoted as 'model'). The dataset is then passed through this intermediate model to obtain features for training, validation, and testing sets. The extracted features are flattened to be compatible with the input requirements of the K Neighbours Classifier. It is instantiated with a specified number of neighbours (k=20). The flattened features from the training set are used to train the KNN classifier. Subsequently, the classifier is employed to predict the labels for the test set based on their extracted features. Thus the recall value calculated came out to be **0.931051**

	Neighbors	Recall score
0	1.0	0.931051
1	2.0	0.913579
2	3.0	0.914902
3	4.0	0.905616
4	5.0	0.903777
5	6.0	0.897160
6	7.0	0.894940
7	8.0	0.889198
8	9.0	0.887942
9	10.0	0.881998
10	11.0	0.879553
11	12.0	0.874574
12	13.0	0.872936
13	14.0	0.866970
14	15.0	0.865198
15	16.0	0.860780
16	17.0	0.859434
17	18.0	0.855509
18	19.0	0.853894
19	20.0	0.849430

Fig. 20 Neighbors table in KNN

6. Decision Tree

A Decision Tree classifier for a machine learning task using features extracted from a pre-trained neural network is used. Initially, an intermediate layer model is created to obtain feature vectors from the penultimate layer of an existing neural network ('model') for the training, validation, and test datasets. These features are then flattened into 2D arrays to comply with the input requirements of the DecisionTreeClassifier from the scikit-learn library.

A Decision Tree classifier is instantiated, and the flattened feature vectors from the training set are used to train the classifier. Following training, the classifier is applied to predict labels for the test set based on their respective feature vectors.In summary, this algorithm is extraction of features from a neural network's intermediate layer and employs a Decision Tree classifier to predict labels for a test set, Thus the recall value calculated came out to be **0.971066**

VII. EMPIRICAL STUDY OF STATE-OF-ARTS

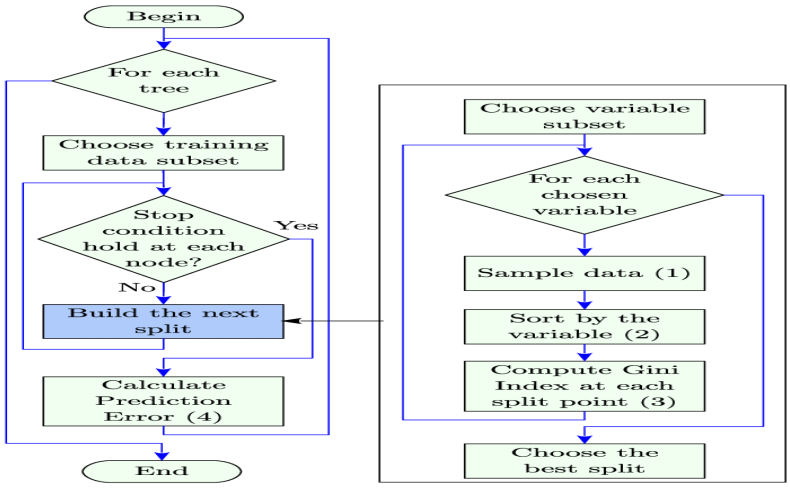
In this section, we analyze the performance of leading stellar classification approaches on our proposed benchmark. In our proposed evaluation setting, we have conducted a detailed analysis of various percentages of training samples. The objective of this analysis is to evaluate the robustness of the current approaches in various challenges for stellar classification and identify the existing limitations to stimulate further research advances. In the conducted analysis, we consider the existing dataset SDSS DR17 with different algorithms, i.e., Random Forest, Logistic Regression ,K-Nearest Neighbor , Decision Tree, XGBoost and Gaussian NB approaches, which all have achieved excellent performance

on the large-scale challenges for Stellar classification. All of these considered networks are trained on the various features classification task and fine-tuned with the X_train dataset for Stellar classification. The experiments are conducted by randomly splitting the Dataset into the training and testing set according to the given training ratio, which ranges from 5% to 80%. Note that, 75% annotated data of the whole dataset are employed for fine-tuning the network while the rest 25% are employed for the testing. Since the validation set is required by the considered methods, we randomly select several training samples to avoid overfitting. For a fair comparison, we train each approach on our proposed benchmark and evaluate the test set following the same setting. As for the evaluation metric, we employ the recall value of all the algorithms.

Recall score	Algorithm
0.982011	Random Forest
0.980217	XGBoost
0.971066	Decision Tree
0.943567	LogisticRegression
0.931051	KNN
0.604185	GaussianNB

1. Overall Performance Evaluation

We start our analysis by reporting the overall Stellar classification performance of each compared approach and summarizing the results. On the test set of our proposed benchmark, among all approaches, Random Forest performs consistently better than others by a short margin. This demonstrates the superior performance of Random Forest . However, as one can observe, Random Forest can only obtain 98.20% recognition accuracy when the percentage of training samples is 75%.



We now train our dataset with Random Forest classification, apply some necessary techniques (e.g., Synthetic Minority Over-sampling Technique), and produce a classification report and confusion matrix for the same distribution of the dataset.

SMOTE: It is a popular machine learning technique for addressing class imbalance in classification problems. When one minority class is underrepresented in the dataset relative to another class or classes, it is referred to as a class imbalance. One potential consequence of imbalanced datasets is biased models that underperform on the minority class. To overcome this we apply SMOTE and balance our distribution for the traing dataset.

```
]
  rf = RandomForestClassifier(max_depth=7 , max_features=3,n_estimators= 100)
  rf.fit(x_train_smote, y_train_smote )
  RandomForestClassifier(max_depth=7, max_features=3)
```

```
RandomForestClassifier
RandomForestClassifier(max_depth=7, max_features=3)
```

Fig. SMOTE for Random classifier

Classification Report: Classification report is a performance evaluation report of the machinge learning model that provides various fields to assess the model predictive performance.The fields on respect to which the report is generated are :Precision, Recall,F1-score,Support.The metrics over which the report is calculated is accuracy, macro avg, weighted avg, etc.

```
print (classification_report(y_test  , rf.predict(X_test)))
```

	precision	recall	f1-score	support
0	0.96	0.97	0.96	14894
1	0.98	0.96	0.97	14892
2	0.99	1.00	1.00	14798
accuracy			0.97	44584
macro avg	0.97	0.97	0.97	44584
weighted avg	0.97	0.97	0.97	44584

Fig. Classification Report for Random classifier

Confusion Matrix: Confusion matrix is a visual representation of graph used to describe the set of values who's true values are known. The components of confusion matrix are :True positive, True negative,false Positive , False Negative.

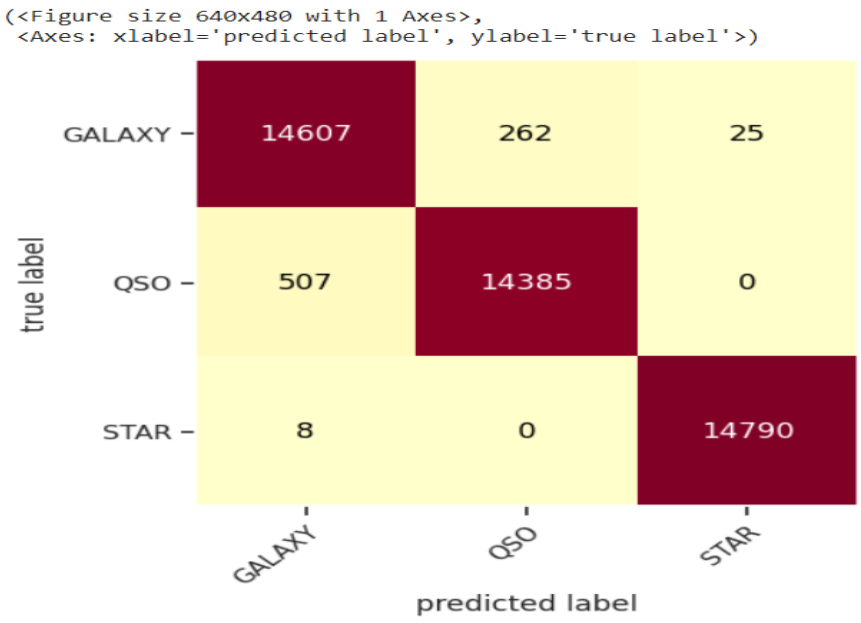


Fig. Confusion Matrix for Random classifier

VIII. EXPERIMENT AND DISCUSSION

In this study, various deep-learning models are used to classify the Class(stars, galaxy, quasar) in the dataset. The most widely used models in the literature have been trained on the SDSS DR17 dataset and have achieved high success. In this study, fine-tuned transfer learning methods are used for the classification of the stars. Seven characteristics of Ultravoilet filter, Green filter, Red filter, Near Infrared Filter, Infrared filter, Redshift, and Plate are used in the conventional identification theory to distinguish different stellar classification classes. Using these seven conventional features, the model predicts the output whether the given features resembles a STAR, GALAXY OR A QUASAR.

```
model = rf

u = float(input("Enter u value: "))
g = float(input("Enter g value: "))
r = float(input("Enter r value: "))
i = float(input("Enter i value: "))
z = float(input("Enter z value: "))
redshift = float(input("Enter redshift value: "))
plate = float(input("Enter plate value: "))

input_features = [u, g, r, i, z, redshift, plate]

y_out = predict_stellar_class(input_features, model)

# Assuming CATEGORIES is a list of class labels used during training
CATEGORIES = ['GALAXY', 'STAR', 'QSO']
predicted_label = CATEGORIES[y_out[0]]

print(f'PREDICTED OUTPUT: {predicted_label}')
```

```
Enter u value: 21.32
Enter g value: 21.17
Enter r value: 20.92
Enter i value: 20.60
Enter z value: 20.42
Enter redshift value: 0.58
Enter plate value: 11069
PREDICTED OUTPUT: STAR
```

Fig. Code snippet that predicts the output using Random classifier

IX. CONCLUSION

In this paper, a SDSS DR17 dataset was created using various features such as redshift, Infrared angle, green filter ,etc which are classified by expert astonomers from space and celestial bodies. The input fields of stars with deep learning architectures .Transfer learning was carried out using pre-trained models. Comparison and evaluation of the experimental results obtained using different network structures are conducted. According to the results, the highest success was achieved by Random Forest. Although the dataset was imbalance andwas also need to be encoded , approximately 97% success was achieved for both test and training data.
A new class(star, galaxy, quasar) determination benchmark has been proposed to promote stellar classification research. Innovative and distinctive features of the benchmark are scalability, diversity difficulty, and public availability. Further validation and testing are needed on adding attributes and part locations for each species to make the benchmark useful for attribute prediction and stellar verification.

X. REFERENCES

[1] Abdurro'uf et al., The Seventeenth data release of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar and APOGEE-2 DATA (Abdurro'uf et al. submitted to ApJS) [arXiv:2112.02026].

[2] Beitia-Antero, L., Yáñez, J. & de Castro, A.I.G. On the use of logistic regression for stellar classification. Exp Astron 45, 379–395 (2018). https://doi.org/10.1007/s10686-018-9591-4

[3] Stellar Classification Dataset - SDSS17. (n.d.). https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17

[4] C. A. L. Bailer-Jones, Bayesian inference of stellar parameters and interstellar extinction using parallaxes and multiband photometry, Monthly Notices of the Royal Astronomical Society, Volume 411, Issue 1, February 2011, Pages 435–452, https://doi.org/10.1111/j.1365-2966.2010.17699.x

[5] Bellas-Velidis, I., Kontizas, M., Dapergolas, A., Livanou, E., Kontizas, E., et al.: Unresolved galaxy classifier for ESA/gaia mission: support vector machines approach. BlgAJ 18(2), 3 (2012) https://ui.adsabs.harvard.edu/abs/2012BlgAJ..18b...3B/abstract

[6] T. Mehta, N. Bhuta and S. Shinde, "Experimental Analysis of Stellar Classification by using Different Machine Learning Algorithms," 2022 International Conference on Industry 4.0 Technology (I4Tech), Pune, India, 2022, pp. 1-8, doi: 10.1109/I4Tech55392.2022.9952964.

[7] Kuntzer, T., Tewes, M., & Courbin, F. 2016 ,"Stellar classification from single-band imaging using machine learning" A&A, 591, A54 https://doi.org/10.1051/0004-6361/201628660

[8] Xi Wang, Fei Xing, and Ping Guo "Comparison of discriminant analysis methods applied to stellar data classification", Proc. SPIE 5286, Third International Symposium on Multispectral Image Processing and Pattern Recognition, (25 September 2003); https://doi.org/10.1117/12.538644

[9] P. Dubath, L. Rimoldini, M. Süveges, J. Blomme, M. López, L. M. Sarro, J. De Ridder, J. Cuypers, L. Guy, I. Lecoeur, K. Nienartowicz, A. Jan, M. Beck, N. Mowlavi, P. De Cat, T. Lebzelter, L. Eyer, Random forest automated supervised classification of Hipparcos periodic variable stars, Monthly Notices of the Royal Astronomical Society, Volume 414, Issue 3, July 2011, Pages 2602–2617, https://doi.org/10.1111/j.1365-2966.2011.18575.x

[10] Li, Xiang-Ru, Yang-Tao Lin, and Kai-Bin Qiu. "Stellar spectral classification and feature evaluation based on a random forest." Research in Astronomy and Astrophysics 19.8 (2019). https://doi.org/111.10.1088/1674-4527/19/8/111