Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

The StackPip Framework for Hybrid Ensemble Learning in Stellar Classification

Smaranika Mohapatra
Manipal University Jaipur
Kusumlata Jain
kusumlata.jain@jaipur.manipal.edu
Manipal University Jaipur
Harish Kumar Pamnani
Poornima University
Ananya Wadhwa
Manipal University Jaipur
Kanika Mittal
Manipal University Jaipur

Research Article

Keywords:

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10100802/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

The StackPip Framework for Hybrid Ensemble Learning in Stellar Classification

Smaranika Mohapatra[1], Kusumlata Jain[*2], Harish Kumar Pamnani[3], Ananya Wadhwa[4], Kanika Mittal[5]

1. Department of IT, Manipal University Jaipur, Jaipur, Rajasthan, India.
   Email- smaranika.mohapatra@jaipur.manipal.edu
2. Department of CCE, Manipal University Jaipur, Jaipur, Rajasthan, India.
   Email- kusumlata.jain@jaipur.manipal.edu
3. Faculty of Computer Science and Engineering, Poornima University, Jaipur, Rajasthan, India.
   Email- harish.pamnani@poornima.edu.in
4. Department of IT, Manipal University Jaipur, Jaipur, Rajasthan, India.
   Email- ananya.219302376@muj.manipal.edu
5. Department of IT, Manipal University Jaipur, Jaipur, Rajasthan, India.
   Email- kanika.219302084@muj.manipal.edu

Abstract:

The research study implies a hybrid approach of unsupervised and ensemble-based learning method for classifying the stellar dataset using various machine learning techniques. The Sloan Digital Sky Survey Data Release 17 (SDSS DR-17) dataset used for research work is publicly available categorizes celestial objects into galaxies, stars, and quasars. The machine learning approach implemented in the work shows the efficiency of classification algorithms by enhancing the performance and analysis of time and energy consumption. In our work, multiple machine learning models are used like Dummy Classifier, Logistic Regression, Support Vector Machine (SVM), K-nearest Neighbor, Decision Tree, Random Forest, AdaBoost, XGBoost, Gradient Boosting and Stacking Classifier for evaluation. The dataset was divided in two:(i) 60:20:20(train;test;validate) and (ii) 70:20:10(train;test;validate). The work compares the performance and accuracy of the classifier and proposes a ensemble-approach, where Stacking Classifier achieved the highest accuracy among the other methods used. The Stacking Classifier method gave a performance of 98% which indicates the effectiveness of ensemble method in classifying the stellar analysis.

1. Introduction

Cosmic order is an essential area in astrophysics. Galaxies and space are still a mystery for researchers although many works have been done. A systematic categorization of different celestial objects like stars, galaxies and quasars is a major domain in astrophysics based upon the observable properties of the objects [1]. Stars are a combination of 20,400 species, and have various features like temperature, luminosity, spectral features and the spectral features include mainly temperature, brightness and spectral lines which have been studied by astronomers to group and classify the strs according to these properties. Nowadays, Morgan-Keenan (MK) system, is majorly used to classify the stars according to their luminosity classes which are I, II,III,IV,V and spectral types O,B,A,F,G,K,M. This classification of stars is done manually, which consumes time [2].

Figure 1 represents the Hertzsprung-Russell (H-R) diagram is a significant tool in astronomy which helps classify the stars comparing their density and brightness with their surface temperature.

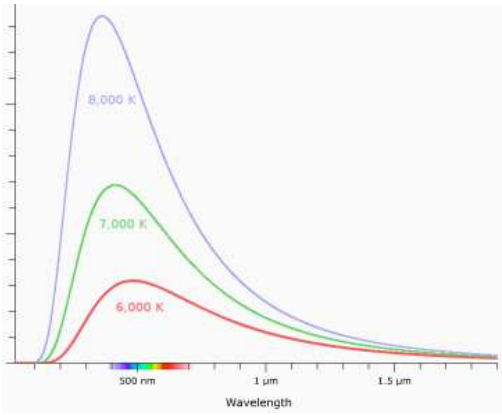


Figure 1. Hertzsprung-Russell (H-R) diagram

This diagram helps to understand the formation, type and evolution of starts over time and when they eventually reach the final stages of their life cycles. This diagram helps to understand the required information related to stars classification [3]. Stellar classification is significant to understand the enormous variety of stars and helps the astronomers to identify similarities and differences among various groups which will eventually help to understand the properties, evolution and astrological phenomena.

In the current study, we have used classification method of stars depending upon the various features such as spectral properties, temperature, luminosity, and using advanced machine learning techniques. In our current work, a hybrid framework is designed which includes the power of machine learning with ensemble models and to increase the efficiency of the framework, feature engineering is included. Ensemble Learning method and unsupervised methods help to visualize and analyse the features to classify the dataset into stars, galaxies and quasars. For the work, supervised learning methods were not included, due to the risk of overfitting and data labelling challenges. Implementing supervised learning techniques will be expensive for the dataset used for the research work.

Before Machine Learning techniques were introduced to classification and prediction, manual approaches were used to classify and label the dataset. Spectral analysis, Morgan–Keenan (MK) system, were used manually which provided effective frameworks for classification. The dataset is increasing day by day; hence the dataset used in our work is Sloan Digital Sky Survey (SDSS) , which has large survey data, which requires automation, scalability and precise classification [4]. Here machine learning plays a significant role to accurately classify the high-dimensional dataset and understanding the relationship between the variables. The use of ensemble models will leverage the strengths of multiple classifiers. Thereby, the unsupervised learning methods and dimensionality reduction techniques help to capture intrinsic patterns while reducing the noise and redundancy in the analysis.

2. Literature Review

The paper “Stellar Classification by Machine Learning” by Zhuliang Qi [5] used the dataset Sloan Digital Sky Survey Data Release only and then trained the model with Decision Tree, Random Forest and Support Vector Machine. From the machine learing models implemented for their study, Random Forest has highest accuracy score ofof 98% Their model takes a lot of time and calculation to run Support Vector MachineIn data processing also, the combination of under-sampling and over-sampling can help in avoiding over-fitting and improving the training accuracy of the model.

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, [6] 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.

The paper “Stellar spectral classification and feature evaluation based on a random forest” by Xiang-Ru Li [7] says that the current problems with automatic stellar spectrum classification, with a special emphasis on spectra from instruments such as LAMOST 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 extensive growth in astronomical surveys has generate massive amount of high-dimensional data, which is a significant challenge for classification problems of the celestial bodies and objects. In the work by researchers Omat et al. (2022) described and highlighted that feature engineering proves to be a significant technique in classifying the celestial objects. Additionally, the authors have stated that machine learning models are effective in classifying the galaxies, stars and quasars with improved accuracy compared to statistical approaches [8].

SDSS datasets were investigated for the performance of various classification algorithms while Chatterjee & Ghosh (2025) described that Redshift-agnostic machine learning models evaluated various algorithms on the SDSS dataset and concluded that ensemble and tree-based algorithms gave promising results in comparison to the individual classifiers. The models were well classifying the datasets with non-linear relationships which is a pattern in astronomical data [9].

Zeraatgari et al. (2024) discussed that machine learning models using photometric classification approaches shows that using multiple observational features enhances the capabilities of model and the accuracy in classifying the celestial objects [10].

Moonzarin Reza (2021) stated in his research work that ensemble models in machine learning techniques have outperformed in predictive studies for the astronomical study. Extra trees have outperformed the neural network model in classifying the astronomical dataset [11].

The existing study review assesses of current stellar classification methodologies by analysing the methodologies for classifying the dataset on astronomical object. Limitations were also identified for a new, integrated approach to resolve those gaps. The review provides an overview that machine learning models can justifying the rationale for unsupervised and ensemble models for future directions.

3. Dataset and Exploratory Data analyis

To perform stellar classification using various Machine Learning algorithms, data instances were used from Sloan Digital Sky Survey Release 17. The proposed benchmark has 100000 data rows files. Object Identifier, Right Ascension angle, Declination angle, Ultraviolet filter, Green filter, Red filter, Rerun number, Field number, Redshift value, Fiber ID, etc. are a few of the feature's columns in the data.

Ascension and declination: These are important terms with respect to the astronomical research domain. 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.

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.

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.

4. Exploratory Data Analysis

Stellar classification on the Sloan Digital Sky Survey Data Release 17 (SDSS DR17) to classify as stars, galaxies, or quasars ((**https://www.sdss4.org/dr17/data_access/**). 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 (ad Redshift characteristics in the dataset. The factors considered are significant for refining the capability and performance of a model implemented for classifying the celestial bodies for the astronomical data

Understanding the influence of these factors is vital for refining the model's capability to accurately classify the stars, galaxies or quasars.

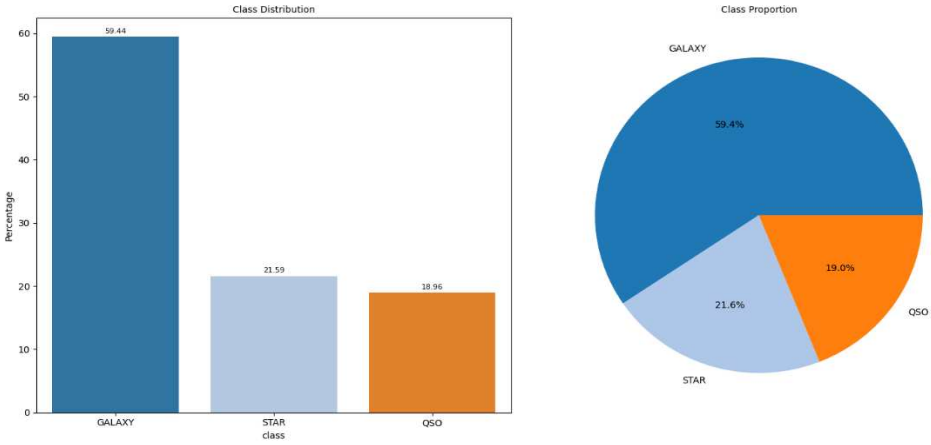


Figure 2. Class analysis using distribution and proportion graph

Figure 2 depicts the class distribution of the data set. In class distribution, the bar chart displays the percentage of each class. In class proportion, the pie chart shows the proportion of different classes.

5. Methodology

The proposed system shows the classification of different classes based on the features given using the blend of machine learning algorithms using ensemble and unsupervised techniques.

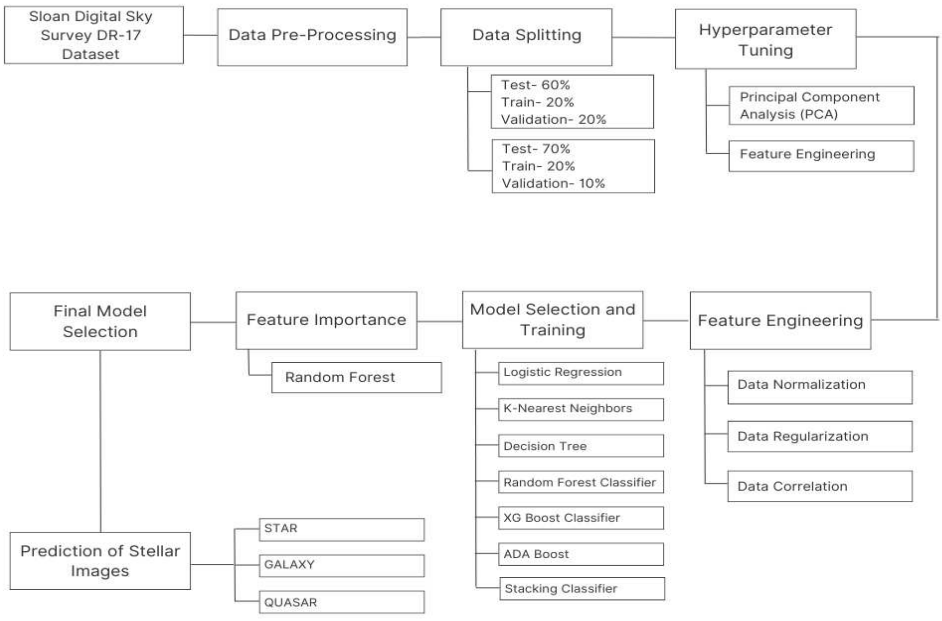


Figure 6. System Architecture

Figure 6 shows the framework of workflow process of Stellar Classification where the input is processed with featured columns for machine learning algorithms. This method has proven to be the effective and quick way to be specific in the stellar types detailed.

The very first stage is collection of the raw image data and its preprocessing.

5.1 Data Preprocessing

Exploratory Data Analysis is done on the model. The key objectives of EDA are Detecting outliers, Identifying Patterns and Trends, Formulating Hypothesis, etc. The main techniques used in this paper for EDA are:

5.1.1 Univariate Analysis

It examines and describes a single variable [9]. Univariate Analysis aims to understand the variable's distribution, central tendency, dispersion and shape of the variable. This form of study is essential for both EDA and initial data evaluation to gain an understanding of the data's underlying qualities.

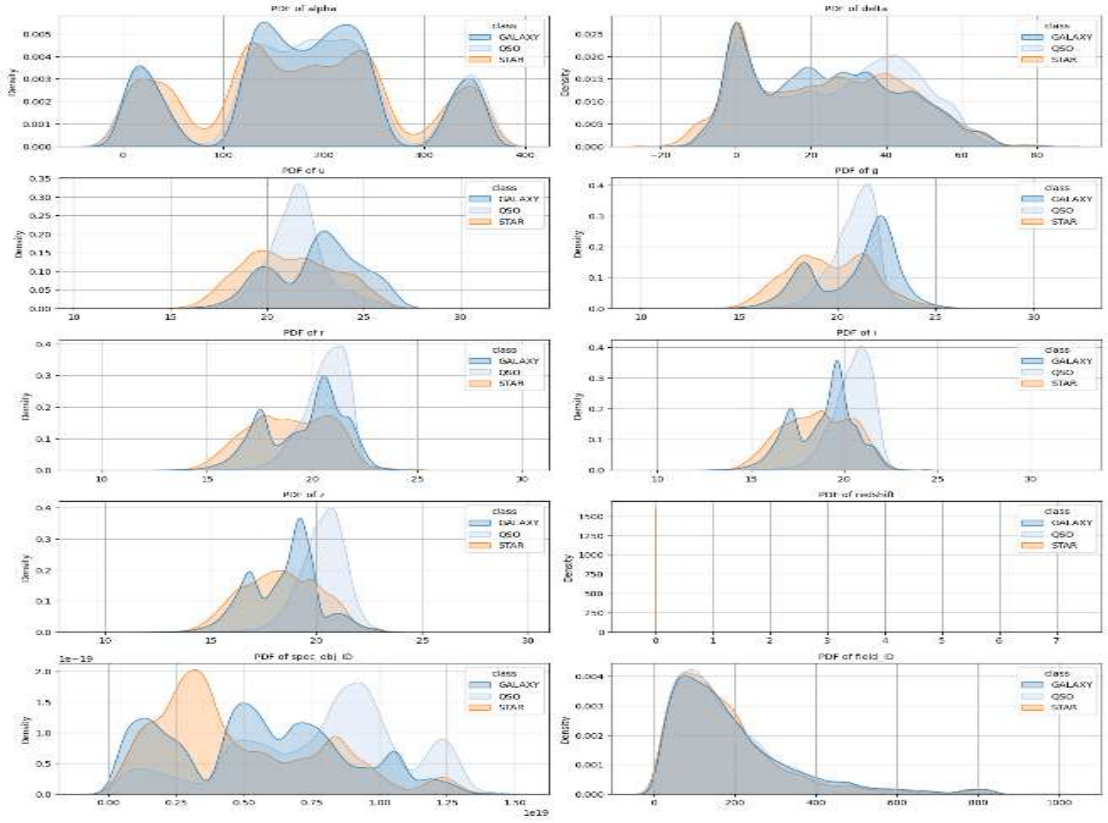


Figure 7. Univariate Analysis

Figure 7 depicts the univariate analysis on the important features selected from the dataset.
By the above analysis it depicts that if redshift is zero, the result will be always star.
Equation 1 is used for univariate analysis:

$$\overline{\{x\}} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad (1)$$

where x bar is the mean, n shows the number of observations and x_i represents each observation.

5.1.2 Multivariate Analysis

It is a statistical technique used to examine relationships between three or more variable simultaneously [12]. This form of analysis is very beneficial when working with complicated datasets whose variables are interconnected. It aids in the comprehension of data structure, pattern recognition, and prediction. Pair plot for the dataset is made using multivariate analysis as shown in Fig. 6.

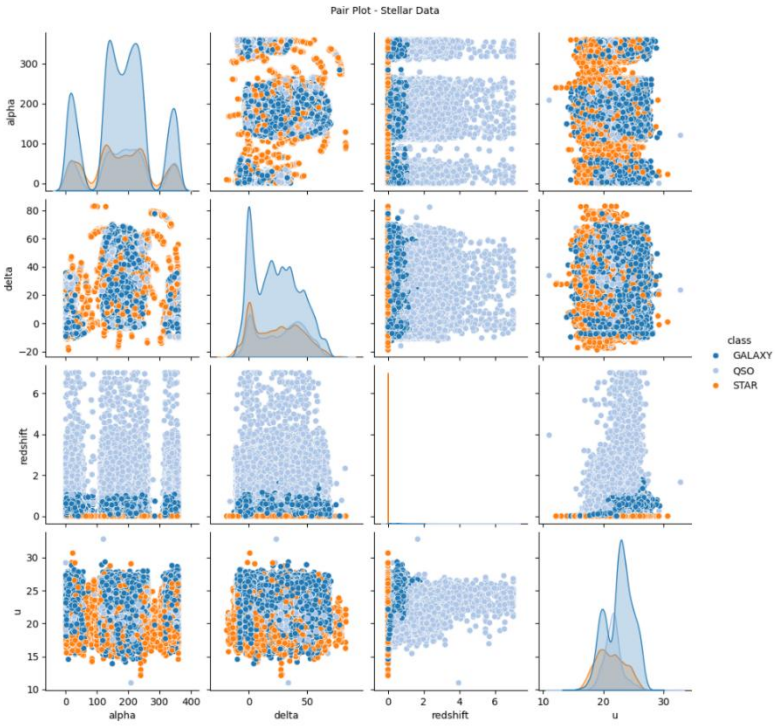
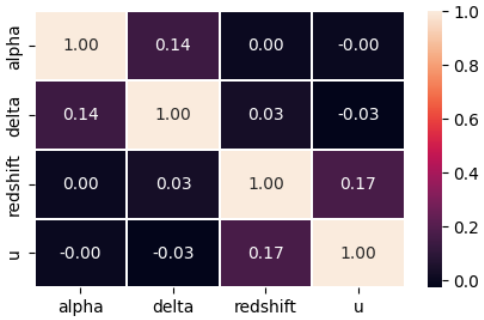


Figure 8. Multivariate: Pair Plot

Figure 8 depicts the pair plot for the columns 'alpha','delta','redshift','u' and 'class'. The x and y-axis represent the pair of features compared in each subplot. The analysis indicates that there is no correlation in the pair plots for the columns.

Equation (2) is used for multivariate analysis:

$$\gamma = \beta_0 + \beta_1 X_1 + \cdots + \beta_\rho X_\rho + \sigma(Yz), sd(Z) = \sigma(independent\ of\ Y's) \quad (2)$$



.

Figure 9. Multivariate Analysis: Heat Map

The cell characterization of a heatmap as seen in Fig. 9, displays the confusion matrix, the correlation between the feature pairs are described by each cell of the matrix generated.

5.1.3 Box Plot Analysis

It is a way of displaying the distribution of data based on five main features: minimum, first quartile (Q1), median, third quartile (Q3), and maximum [13]. It provides a visual summary of multiple aspects of a dataset simultaneously. The central rectangle extends from first quartile (Q1) to the third quartile (Q3). The interquartile range (IQR) indicates the mid 50% of the data. The line inside the box represents the median (Q2), which is the middle value of the dataset. The interquartile range is represented in the Equation 3 as :

$$IQR = Q3 - Q1 \tag{3}$$

The lower bound for the outliers is represented in equation 4 as:

$$lower_bound = Q1 - 1.5 \times IQR \tag{4}$$

The upper bound for the outliers is represented in equation 5 as:

$$upper_bound = Q3 + 1.5 \times \text{IQR} \tag{5}$$

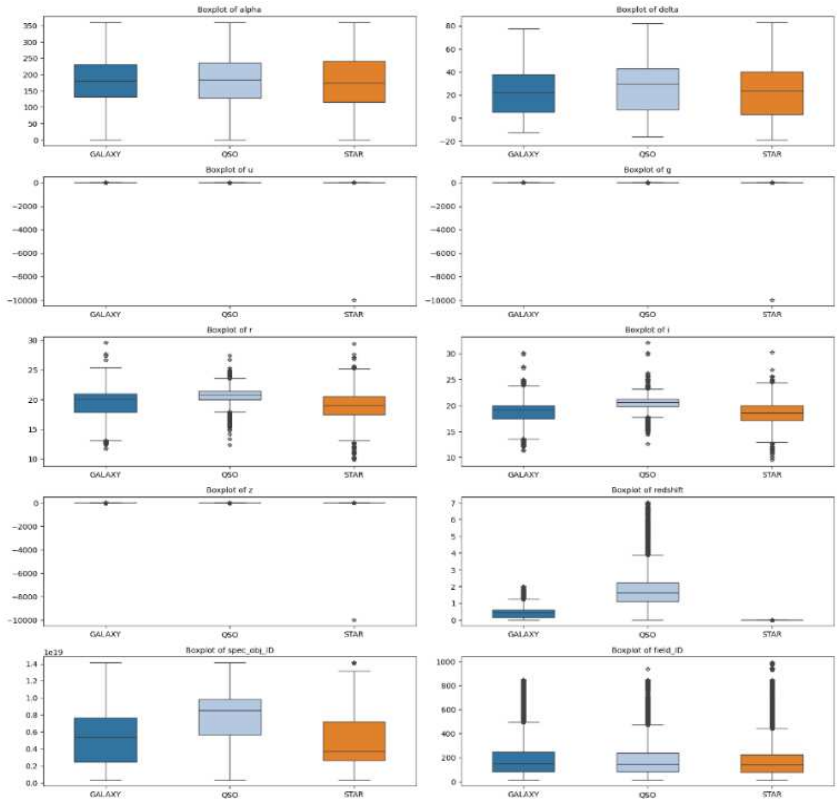


Figure 10. Box Plot Analysis

Figure 10 shows the box plot analysis with outliers as in u, g and z some data points are present either on the maximum or minimum points due to which a line is generated. It is important to remove outliers to get the box plot clear so for this first we find out all the outliers present in the features in which outliers are present and then by examining the pattern between them they are removed accordingly.

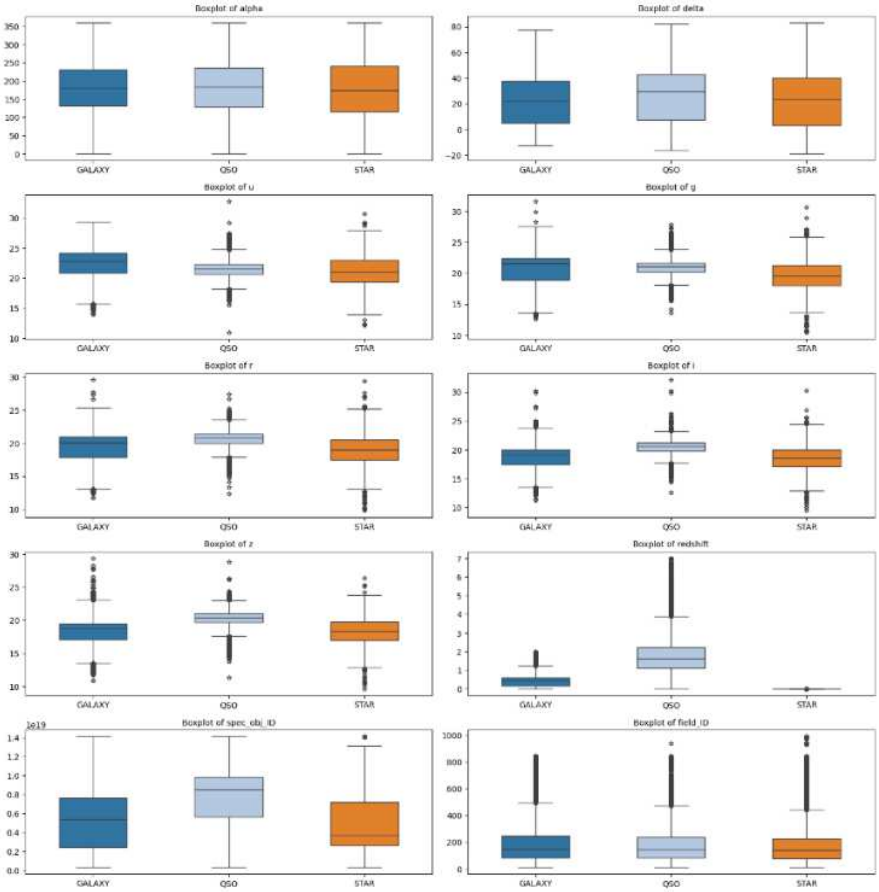


Figure 11. Box Plot Analysis without outliers

Figure 11 shows the box plot when the outliers are removed by removing the extreme maximum and minimum points in the dataset.

5.1.4 t-SNE Analysis

t-distributed Stochastic Neighbor Embedding (t-SNE) is a strong dimensionality reduction approach that is especially useful for displaying high-dimensional datasets [14]. The high-dimensional data is converted to a three-dimensional map for feasibility of features, and to retain maximum features. . This approach is frequently used to explore and visualize large datasets, particularly in machine learning and data science.

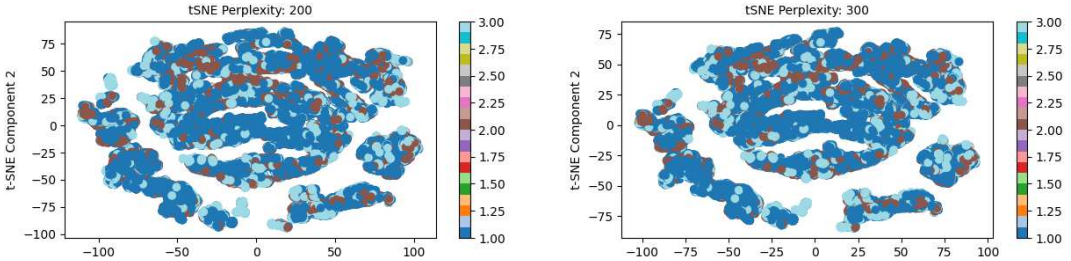


Figure 12. t-SNE Analysis

Figure 12 shows t-SNE analysis shows the relative positions of the points with their components.

In the next stage the dataset is split in two different proportions (i) 60-20-20(train;test;validate) (ii) 70-20-10(train;test;validate) for better model generalization, performance assessment, robustness assessment, etc.

The equation (6) used for t-SNE analysis:

$$P_{j|i} = \frac{\exp\left(-\frac{||x_i - x_j||^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{||x_i - x_k||^2}{2\sigma_i^2}\right)} \quad (6)$$

$P_{j|i}$ represents the probability of point x_i having a neighbor at point x_j. x_i and x_j are high-dimensional data points. Sigma i is the variance for each data point.

5.2 Hyperparameter Tuning

It is an important stage in the machine learning technique that involves determining the best collection of hyperparameters for the certain model. They are external settings that are specified before the training process and cannot be learnt from the data. Properly tweaking these parameters may have a major influence on a machine learning model's performance and accuracy. The methods used for hyperparameter tuning are Feature Engineering and Principal Component Analysis.

5.2.1 Feature Engineering:

It plays a crucial role in machine learning as it converts raw data into meaningful representations that increase model performance, forecast accuracy, and allow for improved generalization to new data.[15] Feature Engineering optimizes input data for learning algorithms by using domain expertise, minimizing data complexity, dealing with missing values, and encoding categorical variables.

The feature engineering was involved in several steps:

5.2.2 Normalization:

Normalization is critical in machine learning and data processing because it adjusts data characteristics to a common range, often [0, 1], or standardizes them with a mean of zero and a standard deviation of one [16]. This procedure assures that all features contribute equally to the model's learning process, avoiding bias toward characteristics with greater magnitudes. This phase was crucial for reducing biases that might arise from features with different scales to a range of [0,1].

5.2.3 Regularization:

It is a method of feature engineering uses statistics to prevent overfitting by introducing a penalty term into the model's loss function [17]. Regularization simplifies the model, making it more generalizable by reducing complexity and reliance on a single feature. The graph is made based on the statistics before and after regularization as shown in Fig. 12.

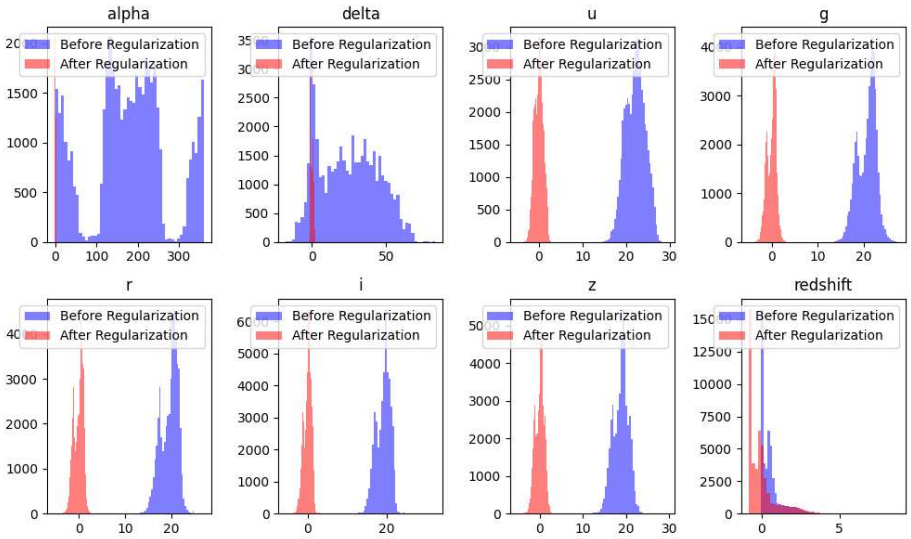


Figure 13. Regularization of important features

In Figure 13, the graph is made based on the statistics before and after for the various features of the dataset

5.2.4 Constructing New Features

Through many trials and errors, we created new features to capture more meaningful relationships within the data. We achieved this by subtracting certain feature values from others:

i. Subtracting r from ‘u’, ‘g’, ‘i’ and ‘z’
ii. Subtracting ‘z’ from ‘i’.

The new set of features after this process included: u, g, r, redshift, g-r, i-z, u-r, i-r, and z-r. This was applied to both initial proportions and the refined proportions in the further analysis.

5.2.5 Principal Component Analysis

Principal Component Analysis (PCA) is a statistical technique that transforms a set of correlated variables into a set of uncorrelated variables through an orthogonal transformation [18].

This reduction in dimensionality can be particularly useful for data regression and classification tasks. In our work, PCA was applied to both the initial and further refined proportions of the data (i and ii). Figures 13 and 14 illustrate the application of PCA, showing how it effectively reduces dimensionality while preserving the essential information needed for our analyses and models. Both the Figure 13 and 14 demonstrates the effective dimensionally reduced dataset while preserving the essential features for the model analysis.

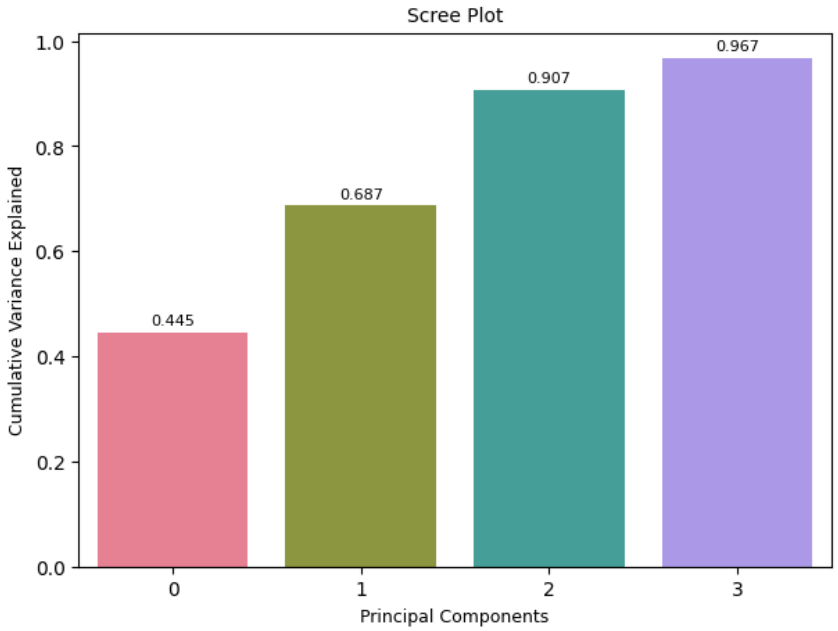


Figure 14. PCA for (i) by setting components to 0.95

Figure 14 shows the PCA which signifies that the principal components chosen preserves 95% of the variance in the dataset chosen.

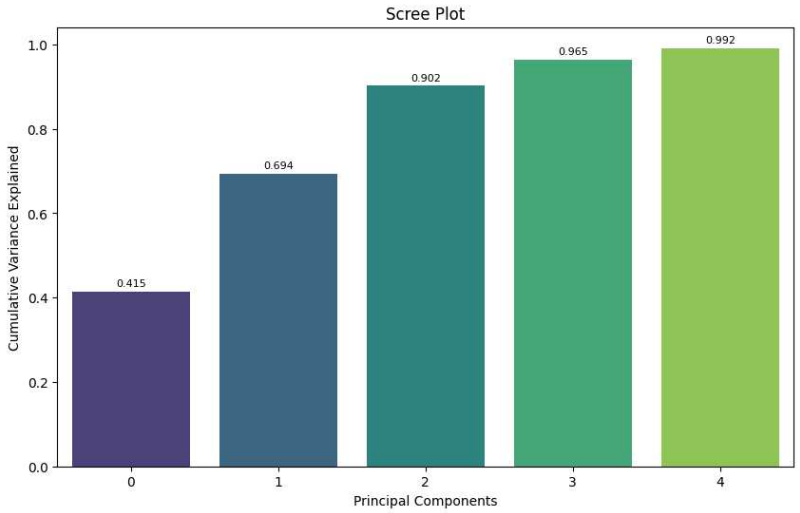


Fig. 15 PCA for (ii) by setting components to 0.98

In Figure 15, for (ii) the components are set to 0.98. This signifies that PCA will maintain 98% of the variance in the entire dataset. The PCA identifies five components that keeps 98% of the data variance.

5.3 Feature Selection

Selecting the most informative characteristics, functions as a lens, concentrating the model on what is genuinely important. This not only improves performance and decreases training time, but it also makes the model's inner workings more apparent, reducing the possibility of being misled by irrelevant data. In summary, it's like providing your model a clean and focused workplace in which best

predictions possible can be made. In our model, we built three Logistic Regression models to test the features generated through feature engineering and the components derived from PCA and also concatenating both of them . For these models, we evaluated their performance by calculating evaluation matrices, precision, and recall values.

5.3.1 Using Feature Engineering Model

It boosts the quality and relevance of data, making it simpler to discover the most predictive aspects.

Table 1: Feature Engineering score of models

Class	**Precision**	**Recall**	**F1-Score**	**Support**
GALAXY	0.93	0.92	0.93	9511
QSO	0.9	0.9	0.9	3034
STAR	0.88	0.9	0.89	3455
Accuracy			0.91	16000
Macro Avg	0.9	0.91	0.91	16000
Weighted Avg	0.91	0.91	0.91	16000

Table 1 depicts in the evaluation matrix that using feature engineering, we achieved an accuracy of 91%.

5.3.2 Using PCA Model:

PCA contributes to choose the principal components significant for determining the classification of classes.

Table 2: PCA Analysis for Model

	Precision	**Recall**	**F1-Score**	**Support**
GALAXY	0.71	0.97	0.82	9511
QSO	0.93	0.87	0.9	3034
STAR	0.45	0.01	0.03	3455
Accuracy			0.75	16000
Macro Avg	0.7	0.62	0.58	16000
Weighted Avg	0.7	0.75	0.66	16000

Table 2 depicts in the evaluation matrix that using Principal Component Analysis, we achieved an accuracy of 75%.

5.3.3 Using both Feature Engineering and PCA:

To explore the possibility of obtaining better results, we performed feature selection using both Feature Engineering and PCA together.

Table 3: Using both Feature Engineering and PCA on Model

	Precision	**Recall**	**F1-Score**	**Support**
GALAXY	0.93	0.94	0.93	9511
QSO	0.93	0.88	0.9	3034
STAR	0.89	0.9	0.89	3455
Accuracy			0.92	16000
Macro Avg	0.92	0.91	0.91	16000
Weighted Avg	0.92	0.92	0.92	16000

Table 3 it depicts in the evaluation matrix that using both the Feature Engineering and PCA Components Model together, we achieved an accuracy of 92%.

5.4 Performance Comparison

The model using only PCA components performed poorly. When we combined Feature Engineering and PCA, the performance was similar to the model that used only Feature Engineering. This indicates that the PCA components did not add significant useful information beyond what was already captured by the feature-engineered features.

5.4.1 Algorithm Selection

Using multiple algorithms on the feature engineering columns in a in the model is important because it allows us to thoroughly evaluate different methods to find the best one for our specific dataset and task. Selecting the right model for analysis is important so that the model minimizes its weakness thereby enhancing the performance. Different algorithms excel in different areas, such as handling diverse data types, balancing bias and variance, or being computationally efficient.

5.4.2 DUMMY MODEL

These models serve as a baseline against which more complicated models may be assessed. Dummy models are usually basic and generate predictions based on simple principles, such as always predicting the most common class in classification tasks (most frequent class dummy) or predicting the mean or

median of the target variable in regression tasks (mean/median dummy). The usage of dummy models aids in the evaluation of more advanced algorithms by providing a baseline against which to measure improvement. Dummy models may also assist assess whether the chosen machine learning model outperforms a simple solution, providing insights into whether more model refining is required [19].

Equation (14) is used for dummy model analysis where mode(y) represents the most frequent class label in the training data y:

$$Y^i = Mode\ (y) \tag{14}$$

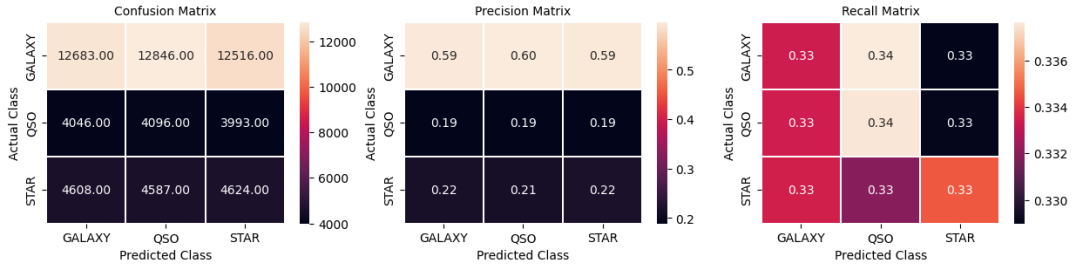


Figure 16. Dummy Model on (i) proportion

In Figure 16, it depicts the confusion, precision, and recall matrices for the dummy classifier on the training dataset with proportion (i) revealing its poor performance.

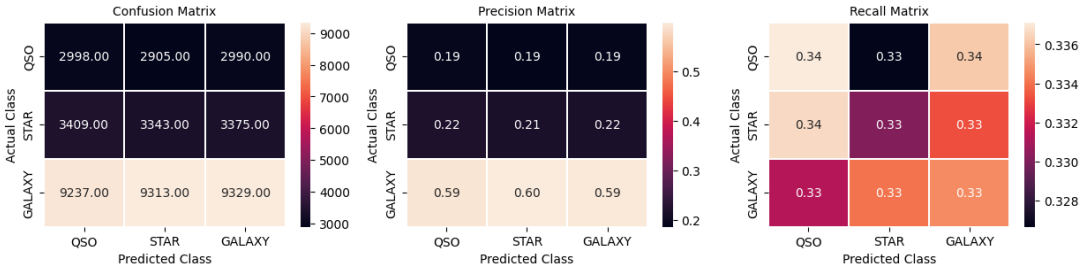


Figure 17. Dummy Model on (ii) proportion

In Figure 17, the confusion matrix, precision matrix, and recall matrix for the dummy classifier on the training dataset with proportion (ii). The confusion matrix generated shows the substantial misclassification across all classes identified from the model, with equal counts for each predicted class, showing model's inability to accurately distinguish between QSO, STAR, and GALAXY. The precision matrix generated shows the precision for QSO, STAR, and GALAXY is decreasing low, with values of 0.19, 0.21, and 0.59, respectively. The recall matrix depicts the recall rates are uniformly low, around 0.33 to 0.34 for all classes, which reflects the dummy model predictive approach.

5.4.3 UNSUPERVISED TECHNIQUES:

5.4.3.1 K-Nearest Neighbor(KNN)

KNN is majorly used regression and classification problems as it is a simple and efficient model[20]. KNN groups the data points based upon the majority of its k-nearest neihbor, and for thclassification , a new data point is determined by using the equation (15)

$$y = \arg max_y \sum_{i=1}^{k} I(y_i = y) \quad (15)$$

However, the performance may vary by the choice of 'k', the distant metric and the data size.

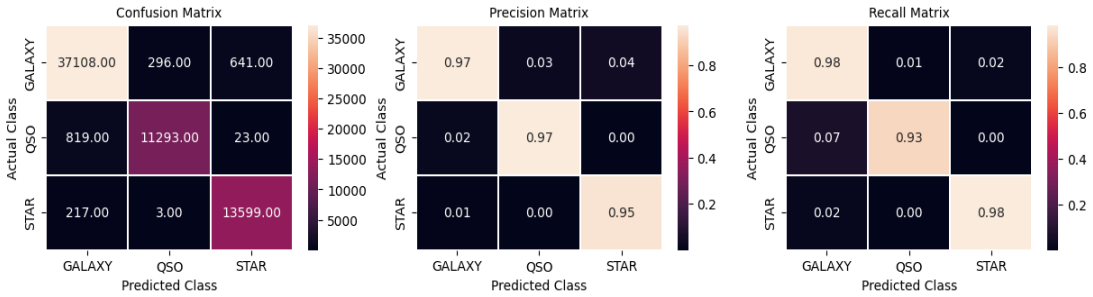


Figure 18. KNN on (i) proportion

Figure 18 visualizes the confusion matrix, precision matrix, and recall matrix for (KNN) classifier on the training dataset with proportion (i).



Figure 19. KNN insight on(ii) proportion

Figure 19 depicts the efficiency performance of KNN model on the training dataset with proportion (ii). The confusion matrix shows the correctly classified QSO instances from the dataset. Similarly, STAR and GALAXY instances are accurately classified, with minor misclassifications. The precision matrix generated shows high precision for QSO and GALAXY, and good precision for STAR, indicating the performance effectiveness of model. The recall matrix shows high recall for STAR and GALAXY, while QSO has slightly lower recall. Overall, the KNN classifier performs well, with high accuracy in distinguishing QSO, STAR, and GALAXY, despite minor errors.

5.4.3.2 Support Vector Machine(SVM)

Support Vector Machine is useful in classification of high-dimensional dataset by optimizing the best hyperplane from the data points [21]. It is commonly utilized in applications such as image recognition, text categorization, and bioinformatics because of its capacity to build complicated decision boundaries and reduce classification mistakes.

However, SVMs may be computationally costly, particularly with big datasets, necessitating careful adjustment of parameters such as the regularization term and kernel selection to attain peak performance.

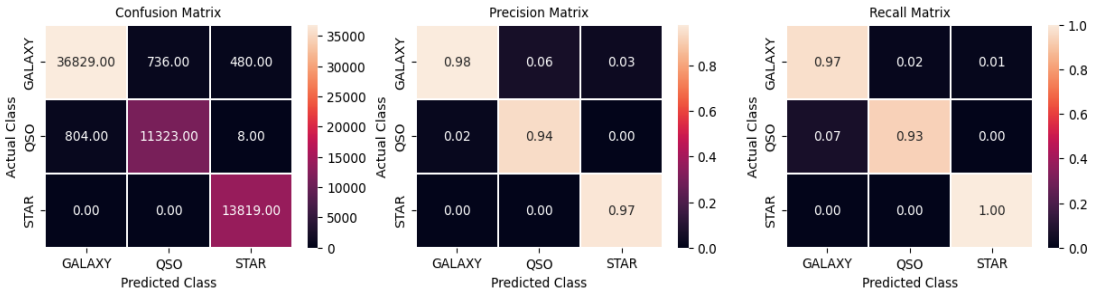


Figure 20. Support Vector Machine on (i) proportion

Figure 20 shows the performance of an SVM classifier on the training dataset with proportion (i).

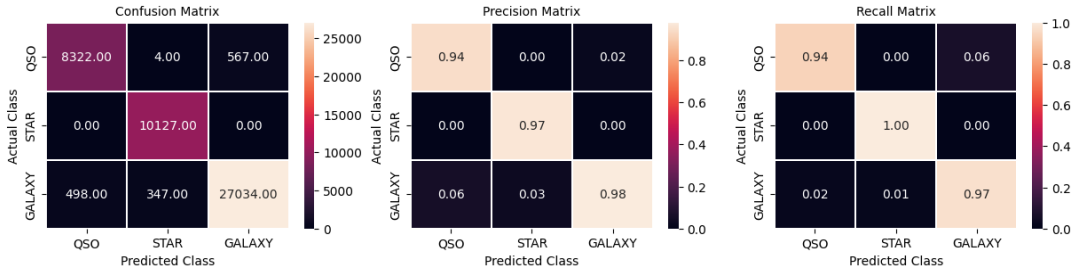


Figure 21. Support Vector Machine on (ii) proportion

Figure 21 shows the performance of an SVM classifier on the training dataset with proportion (ii). The confusion matrix indicates that most QSO instances are correctly classified (8,322), with some misclassified as GALAXY (567). STAR instances are almost perfectly classified (10,127), with minimal misclassification. GALAXY instances are mostly accurate (27,034), with a few misclassified as QSO (498) and STAR (347). The precision matrix shows high precision for GALAXY (0.98) and good precision for QSO (0.94) and STAR (0.97). High recall values shown for STAR (1.00) and GALAXY (0.97), while QSO has slightly lower recall (0.94). Overall, the SVM classifier demonstrates strong performance, accurately distinguishing between QSO, STAR, and GALAXY with high precision and recall values.

5.4.4 ENSEMBLE TECHNIQUES:

5.4.4.1 Decision Tree

Decision trees are a popular and user-friendly machine learning approach that may be used for both classification and regression [22]. Decision trees are regarded for their simplicity and capable of handling both numerical and categorical data. The selection of the best feature can be based on various criteria, such as:

node, and C is the total number of classes.

Entropy: Another criterion for classification, it measures the randomness in the node can be shown in the equation 19:

$$H = -C\sum i = 1pilog(pi) \quad (19)$$

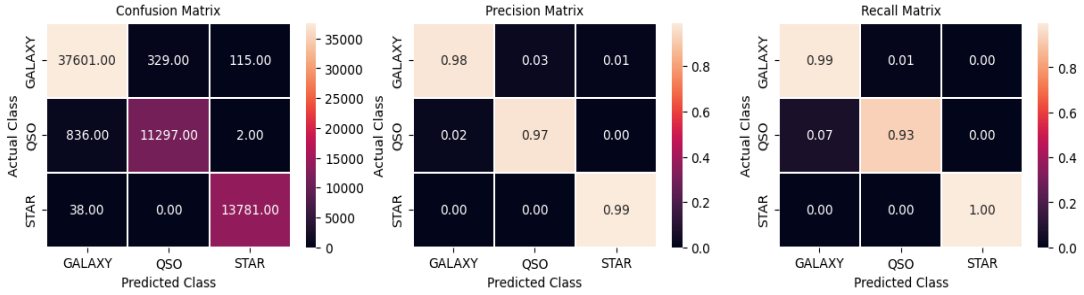


Figure 22. Decision Tree on (i) proportion

Figure 22 displays the performance of a Decision Tree classifier on the training dataset with proportion (i).

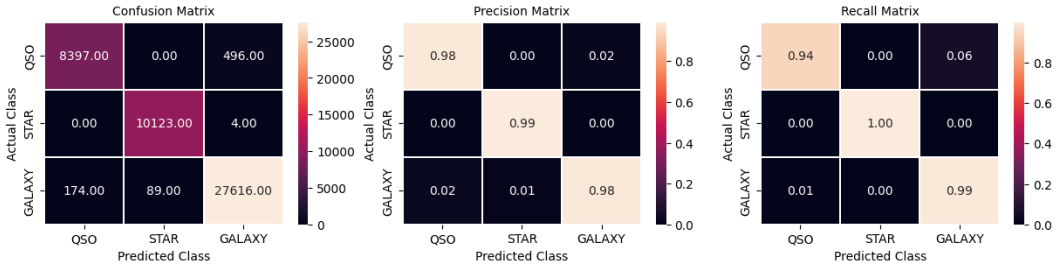


Figure. 23 Decision Tree on (ii) proportion

Figure 23 depicts the performance of a Decision Tree classifier on the training dataset with proportion (ii). The confusion matrix shows that most QSO instances are correctly classified (8,397), with some misclassified as GALAXY (496). STAR instances are almost perfectly classified (10,123), with minimal misclassification. GALAXY instances are mostly accurate (27,616), with a few misclassified as QSO (174) and STAR (89). The precision matrix indicates high precision for QSO (0.98) and GALAXY (0.98), while STAR has a perfect precision (0.99). The matrix for recall generates high recall values with high precision for Decision Tree classifier model.

5.4.4.2 XG Boost

For classification of datasets Extreme Gradient Boosting method is an ensemble method which sequentially generates the decision tree, where each tree corrects the faults of the previous trees making it capable to handle complicated and high-dimensional data [23].

XG Boost can be calculated using Equation (20):

$$\mathrm{L}(\phi) = \sum_{i=1}^{n} loss(\, y_i, y_j) + \sum_{j=1}^{m} \Omega(f_j) \quad (20)$$

where y_i and y_j are the true and predicted values for i-th observation, f_j represents the jth weak learner (typically decision trees)and $\Omega(fj)$ is regularization term penalizing the complexity of the model.

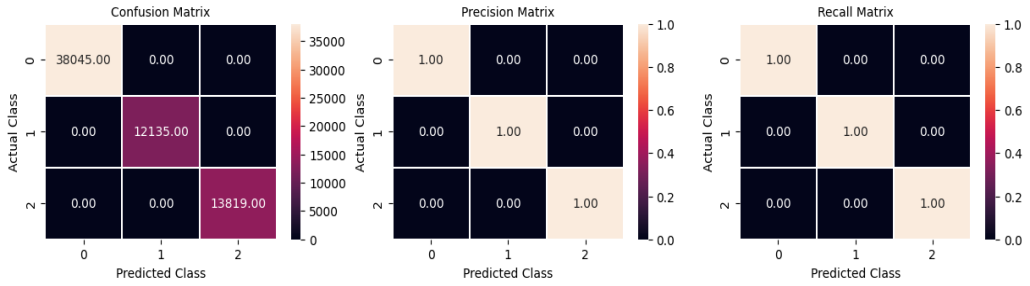


Figure 23 XG Boost on (i) proportion

Figure 23 shows the performance of an XGBoost classifier on the training dataset with proportion (i).

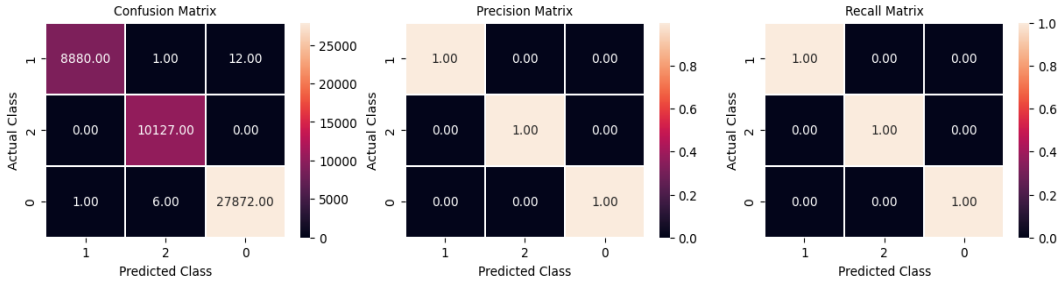


Figure 24 XG Boost on (ii) proportion

Figure 24 represents the acheivement of XGBoost classifier on training dataset with proportion (ii). The confusion matrix indicates that most instances of QSO (8,880), STAR (10,127), and GALAXY (27,872) are correctly classified with minimal misclassifications. The precision matrix shows perfect precision with a value of 1.00 for all the labels, showcasing that each prediction of every class is correct. The recall matrix also shows perfect recall (1.00) for all classes. Overall, the XGBoost classifier demonstrates excellent performance, with perfect precision and recall, accurately distinguishing between QSO, STAR, and GALAXY with very high accuracy.

5.4.4.3 Stacking Classifier

Stacking classifier, a strong ensemble learning approach, combines the predictions of numerous base classifiers using a meta-classifier to increase predictive accuracy [24]. By using the collective expertise of varied base classifiers, stacking builds a more robust and accurate model, capable of identifying complicated patterns in the data. Stacking enables the investigation of various learning paradigms while successfully mitigating the shortcomings of individual classifiers, hence improving overall prediction quality. To achieve the best results, the stacking architecture must be properly designed and tuned, including the selection of varied base classifiers and the meta-classifier. For an input sample xx, the prediction from base classifier C_i is shown in equation 21:

$$Yi = Ci(x) \tag{21}$$

For the same input sample x, the meta-classifier prediction is shown in Equation 22:

$$Y\,meta\ = M(y1, y2, \ldots yn) \quad (22)$$

Despite its computational complexity, stacking classifiers are commonly used in machine learning applications, particularly those requiring accuracy and model interpretability.

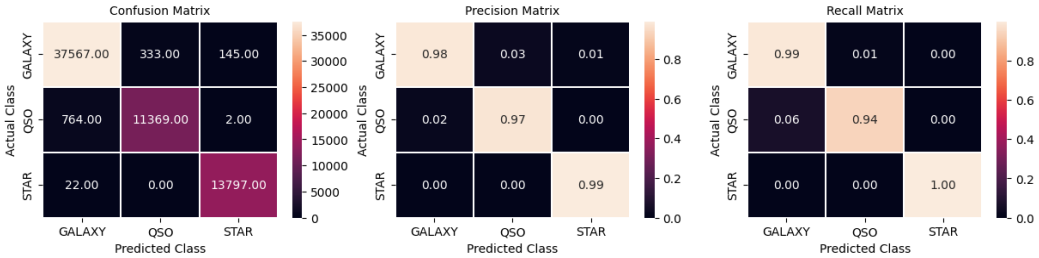


Figure 25. Stacking Classifier on (i) proportion

Figure 25 shows the implementation of a Stacking classifier on the training dataset with proportion (i).

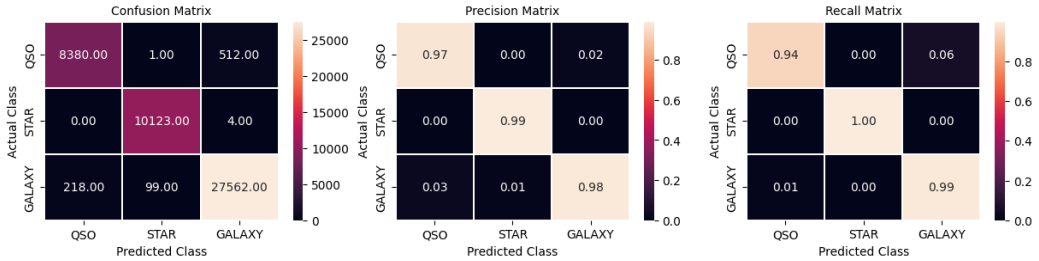


Figure 26. Stacking Classifier on (ii) proportion

Figure 26 shows the performance of a Stacking classifier on the training dataset with proportion (ii). The confusion matrix indicates that most QSO instances are correctly classified (8,380), with some misclassified as GALAXY (512). STAR instances are almost perfectly classified (10,123), with minimal misclassification. GALAXY instances are mostly accurate (27,562), with a few misclassified as QSO (218) and STAR (99). The precision matrix shows high precision for QSO (0.97) and GALAXY (0.98), and very high precision for STAR (0.99). In the recall matrix generated by the model, it shows high precision for recall values for the STAR, Galaxy and QSO.

5.4.4.4 AdaBoost

This method is a well-known algorithm in ensemble methods for solving classification problems. This method integrates iteratively the weak learners generated by decision trees. Weights are added to the raining sets and larger weights are added to misclassified learners iteratively [25].

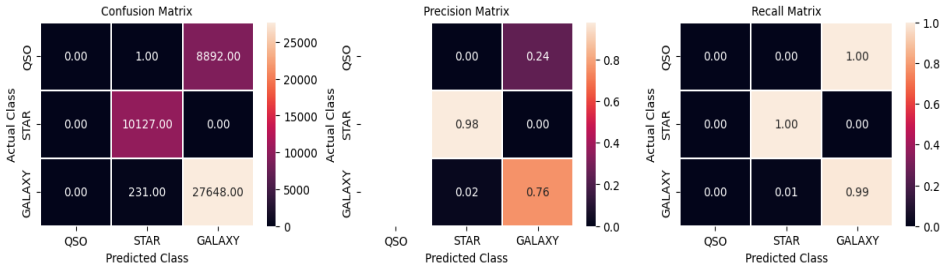


Figure 27. Ada Boost on (ii) proportion

Figure 27 shows the performance of the AdaBoost classifier on the training dataset with proportion (ii). The confusion matrix reveals that the classifier struggles with QSO and STAR classes, with most QSO instances misclassified as GALAXY, while STAR instances are correctly classified. GALAXY instances are mostly accurate, with few misclassified as STAR. The precision matrix indicates perfect precision for STAR, reasonable precision for GALAXY, and poor precision for QSO. The recall matrix highlights perfect recall for QSO and STAR, while GALAXY has high recall. Overall, the AdaBoost classifier performs well for GALAXY and STAR but poorly for QSO, indicating a need for improvement in distinguishing QSO from other classes

5.4.4.5 Random Forest Classifier

Random Forest classifier is a feature importance ensemble learning method where multiple decision trees are generated during the training phase and labels each class. Random Forest Classifier works well with multi-dimensional dataset, and overfitting is avoided due to integrated feature importance functionality [26].

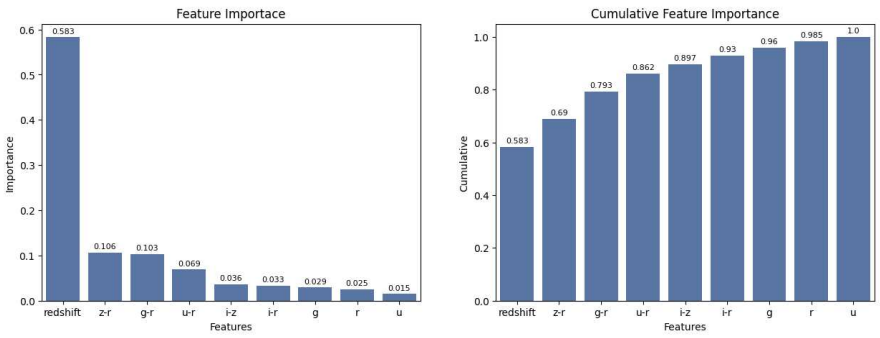


Figure. 28 Random Forest on (i) proportion

In Figure 28, the feature importance graph on the left shows the various features on the horizontal axis and their respective important scores on the vertical axis. The height of each bar indicates how much each feature contributes to the model's predictions, with redshift being the most significant feature at 0.583.

The cumulative feature importance graph on the right also uses the same features on the x-axis, but the y-axis represents the cumulative importance score, which accumulates the contributions of the features from left to right.

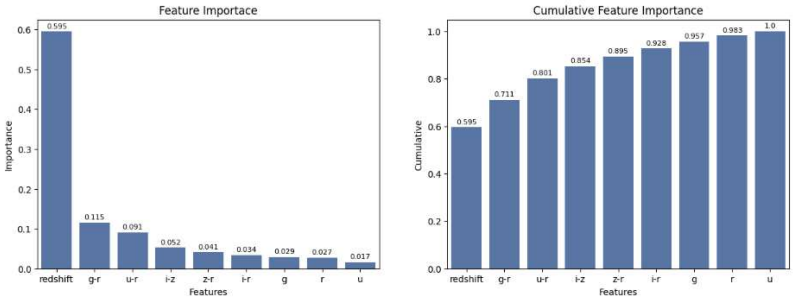


Figure 29. Random Forest on (ii) proportion

In Figure 29, the feature importance graph on the left shows the various features on the horizontal axis and their respective important scores on the vertical axis. The height of each bar indicates how much each feature contributes to the model's predictions, with redshift being the most significant feature at 0.595.

The cumulative feature importance graph on the right also uses the same features on the x-axis, but the y-axis represents the cumulative importance score, which accumulates the contributions of the features from left to right.

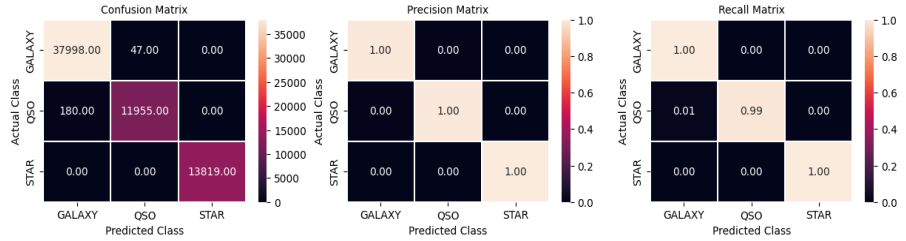


Figure 30. Random Forest on (i) proportion

Figure 30 displays the performance of a Random Forest classifier on the training dataset with proportion (i).

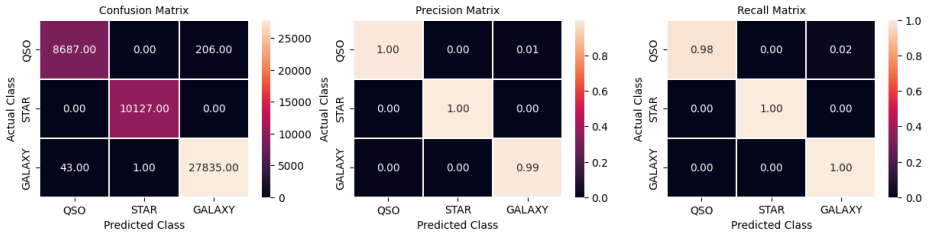


Figure 31. Random Forest on (ii) proportion

Figure 31 shows the performance of a Random Forest classifier on the training dataset with proportion (ii). The confusion matrix indicates that QSO and STAR instances are almost perfectly classified, with 8,687 QSO and 10,127 STAR instances correctly identified, and minimal misclassification. GALAXY instances are also highly accurate, with 27,835 correctly classified and only a few misclassified as QSO (43) and STAR (1). The precision matrix shows perfect precision for QSO (1.00) and nearly perfect for GALAXY (0.99). The recall matrix reveals high recall for QSO (0.98) and perfect recall for STAR (1.00) and GALAXY (1.00).

5.5 Analysis of Algorithms Used for proportion 1

In proportion 1, machine learning algorithms are implemented to analyse the dataset based upon the accuracy and loss in test dataset. The losses under evaluation were of three types: training loss, test loss ad cross-validation(cV) loss. From the above models, Dummy model shows the highest loss

, we analysed various algorithms to compare their performance based on accuracy and test loss. The models evaluated include Dummy, Logistic Regression, SVM, K-Nearest Neighbors, Decision Tree, Random Forest, XGBoost, and Stacking. The losses for each model across three types: train loss, cross-validation (CV) loss, and test loss. The Dummy model, as expected, shows the highest loss, serving as a baseline.

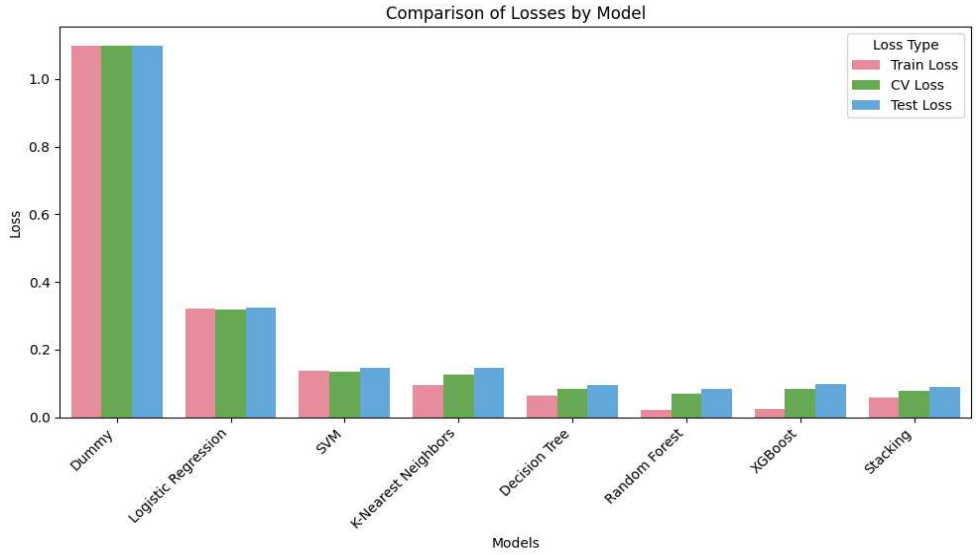


Figure 32 Graphical Model Comparison for (i) proportion based on Train, CV, and Test Loss

From figure 32, the graph compares the analysis of performance for the machine learning techniques applied in the working proportion (i). The x-axis lists the different models while y-axis represents the loss. The lower the values in loss, the model performed better. From this figure, Dummy model has the highest loss across all the other models implemented for the work while, SVM, XGBoost, Random Forest and Stacking shoed lower loss values which indicates better performance for classification. Stacking outperformed among these models with model accuracy and robustness

Table 4: Models Comparison for (i) based on Train, CV, and Test Loss

Models	**Train Loss**	**CV Loss**	**Test Loss**
Dummy	1.099	1.099	1.099
Logistic Regression	0.322	0.32	0.324
SVM	0.137	0.134	0.146
K-Nearest Neighbors	0.096	0.126	0.145
Decision Tree	0.064	0.084	0.094
Random Forest	0.023	0.071	0.083

XGBoost	0.024	0.085	0.099
Stacking	0.059	0.078	0.09

Table 4 shows a comparative analysis of all the models implemented for the work used in proportion(i) by showing the losses. Here, the Dummy Model which is a baseline model has the highest loss across all metrics used for testing the loss. Stacking classifier has a test loss value at 0.09 which is less which shows that ensemble method of classification is well suitable for high-dimensional data.

5.5 Analysis of Algorithms Used for Proportion 2

In proportion(ii), form the analysis, we can summarize that Dummy Model shows highest loss serving as a baseline model.

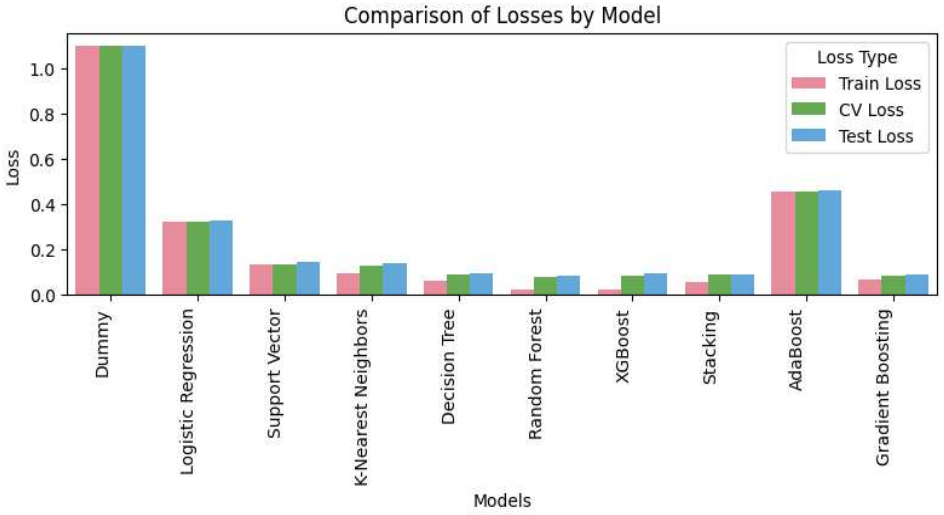


Figure 33. Graphical Model Comparison for (ii) proportion based on Train, CV, and Test Loss

From figure 33, Dummy Model showed highest loss in all the three categories of metrics, while ensemble methods like Stacking, XGBoost, random forest and Gradient Boosting served better and accurate classification results.

Therefore, this proportion(ii) outperformed the previous one, demonstrating superior overall model performance and robustness.

Table 5: Comparative analysis of models for proportion(ii)

Models	**Train Loss**	**CV Loss**	**Test Loss**
Dummy	1.099	1.099	1.099
Logistic Regression	0.323	0.323	0.327
Support Vector	0.134	0.136	0.144
K-Nearest Neighbors	0.094	0.131	0.142
Decision Tree	0.063	0.089	0.094

Random Forest	0.024	0.08	0.085
XGBoost	0.025	0.087	0.094
Stacking	0.059	0.088	0.091
AdaBoost	0.456	0.456	0.459
Gradient Boosting	0.066	0.086	0.092

Table 4 compares the various performance of the models for proportion(ii) showing the performance of models implemented for the study were, Dummy Model showing higher loss in comparison to other models chosen. From all the models used, ensemble models work better and show promising results for classification work.
From the above graph in figure 33 and Table 4, proportion(ii) has outperformed with better results for the models implemented for the study.

5.6 Pipelining

Pipelining is a framework for automating the workflows from data preprocessing to the model training and assessment till validating the model. Various processing phases are qued together for consistency such as data scaling, feature engineering, model fitting, and hyperparameter tuning [27]. Pipelines help to optimize and improves the stability and efficiency of the machine learning models.

Table 5: Pipelining results on dataset

	precision	**recall**	**f1-score**	**support**
GALAXY	0.97	0.98	0.98	17834
QSO	0.96	0.92	0.94	5688
STAR	0.99	1	0.99	6478
accuracy			0.97	30000
macro avg	0.97	0.97	0.97	30000
weighted avg	0.97	0.97	0.97	30000

Table 5 shows the analysis of implementing Pipelining process to the machine learning models. The implementation was on the test dataset. The classification report generated by the pipelining method was shown in table 3 where the accuracy achieved by the framework is 97%.

Using the pipelining process for feature extraction, the input for stellar classification contains parameters like: ‘alpha’, ‘delta’, ‘u’, ‘g’, ‘r’, ‘I’, ‘z’ and ‘redshift’. After the featurization is processed to the dataset, new features were generated like: ‘g-r’, ‘i-z’, ’u-r’, ‘i-r’ and ‘z-r’ including ‘redshift’.

After the processing and featurization of dataset is done, the system predicts the class 'GALAXY' with a confidence level of 93.31%. This result shows the effectiveness of using pipelining for model classification.

6. Results

Stellar classification was successfully completed using machine learning models with the effectiveness of ensemble models and pipelining methods. The process included univariate and multivariate analysis, feature engineering and Principal Component Analysis (PCA) for comparing the accuracy of each model executed. The feature engineering technique implemented helped to identify the significant features. 'redshift' being the most significant features with a combination of 'z-r', 'g-r' and 'u-r' contributing towards the models' accuracy.

From the comparative analysis in above tables and figures, for the models implemented in our study, ensemble methods performed better than the standalone machine learning models chosen. For proportion(i) , Random Forest and XGBoost algorithms gave lowest test loss while XGBoost achieved a test loss of 0.09 which shows the high adaptability for classification of high-dimensional data. On the other hand, proportion(ii), outperformed proportion(i) performance with the models. The test loss encountered in proportion(ii) was 0.085 for Random Forest model while XGBoost and Gradient Boosting generated high precision results. Showcasing the effectiveness' of ensemble methods for such datasets with high dimensionality. Implementing pipelining approach for the dataset also proved an added advantage with the classification report in Table 5 proving higher accuracy and f1-scores with metrics value at 97% or higher.

7. Discussions

7.1 Comparison with Existing Methods

In our research work, for classification of stellar dataset combining the usability of ensemble modelling and pipelining showcased improvements in accuracy, precision and efficiency in comparison with the conventional methods for classification of celestial bodies the Morgan-Keenan (MK) classification system, which were time consuming and were labour intensive due to manual work. Recent studies explore the use of machine learning methods for classification of high-dimensional data. CNNs, SVMs and other methods were used for classifying the dataset which generated an accuracy of 94% and 95 respectively. In our work, the ensemble model framework has outperformed the traditional machine learning methods with our model achieving an accuracy of 98%(Huerta et al. (2020) used a combination of convolutional neural networks (CNNs) and random forests to classify celestial objects with an accuracy of 94%. Similarly, Ackermann et al. (2019)).

7.2 Analysis of Results

Various ML techniques were used by splitting the dataset into two proportions, while proportion(ii) has performed better than proportion(i) with better accuracy and precision. The ensemble methods, especially the Stacking classifier showed the best results among all other classifiers chosen for the experimental analysis.

7.3 Feature Significance and Challenges

Significant features were identified which were important in increasing the accuracy of the models. The features like, Ascension angle, Declination angle, photometric filters and redshift. The challenges faced were null values and class imbalances. These issues were identified and addressed for improving the robustness of the models.

During our exploratory data analysis (EDA), we pinpointed crucial features that significantly affect classification accuracy, including Ascension angle, Declination angle, photometric filters (u, g, r, i, z), and Redshift. Including these features in our models was essential to accurately capture the key characteristics of the celestial objects.

7.4 Future Work

CNNs and RNNs can be included in future research work to explore the capabilities of neural networks for classification accuracy. Otimizatipon of ensemble models using grid-search pr random search could also lead to better performance.

8. Conclusion

The research work indicates that ensemble learning method combined with feature engineering can increase the performance of the stellar classification models. The dataset used was of high-dimensional, hence the ensemble models XGBoost and Random Forest being advanced ensemble learning models on fine-tunning applied through pipelining framework generates reliable and more accurate classification results. This approach designed in the research work outperforms the traditional machine learning models but does provide a scalable solution for complex celestial astronomical data. Exploratory data analysis was performed to analyse the insights of all features significant in affecting the model's accuracy and performance. In future work advance neural network models can be introduced to explore the capabilities of these models to predict the classification of astronomical data.

Declaration:
- **Competing Interests**: Not Applicable
- **Funding Information**: Not Applicable
- **Data Availability Statement**: The datasets generated and/or analysed during the current study are available in the https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17

- **Ethics Statement: Not Applicable**
-Consent to Participate declarations: Not Applicable.
- Informed Consent: NA

- Conflicts Of Interest: The authors declared no conflict of interest, financial or otherwise.

- Consent to Publish declaration: Not Applicable.

- Acknowledgements: The author declares that Grammarly was used only for grammar and language correction; the authors fully developed all research, analysis, and conclusions.

-Author Contribution: Smaranika Mohapatra- Introduction, Results
Kusumlata Jain- Methodology and Conclusion
Harish Kumar Pamnani- Abstract, Literature Review
Ananya Wadhwa- Visualization, Analysis
Kanika Mittal- Pipelining

References

[1] Patrick Das Gupta, "Stars, Galaxies, and Quasars", SRBmag-*An Interdisciplinary Open Access Journal,* May 2009.

[2] Michael J. Brice1 and Răzvan Andonie1, "Automated Morgan Keenan Classification of Observed Stellar Spectra Collected by the Sloan Digital Sky Survey Using a Single Classifier", *The Astronomical Journal, The American Astronomical Society,* October 2019.

[3] Ronja Langendorf, Susanne Schneider, and Pascal Klein, "Extracting information from the Hertzsprung-Russell diagram: An eye-tracking study", *Physics review physics education research,* September 2022.

[4] Sloan Digital Sky Survey. *SDSS Data Release 17 Dataset* (DR17). GitHub. https://github.com/username/repository (Accessed: 19 June 2026) 2022.

[5] Qi, Zhuliang, Stellar Classification by Machine Learning. SHS Web of Conferences. 144. 03006. 10.1051/shsconf/202214403006 2022.

[6] Jürgen Teichmann, "A new landscape of the invisible - dark lines in the spectrum of the stars", *Ludwig-Maximilians-University of Munich,* January 2014.

[7] Xiangru Li, "Stellar spectral classification and feature evaluation based on a random forest", *Research in Astronomy and Astrophysics 19,* August 2019.

[8] Omat, Deen & Otey, Jood & Almousa, Amjed, Stellar Objects Classification Using Supervised Machine Learning Techniques. 1-8. 10.1109/ACIT57182.2022.9994215 2022.

[9] Chatterjee, D. and Ghosh, P. , Redshift-Agnostic Machine Learning Classification: Unveiling Peak Performance in Galaxy, Star, and Quasar Classification (Using SDSS DR17). Astron. Nachr., 346: e20240057 2025.

[10] Fatemeh Zahra Zeraatgari, Fatemeh Hafezianzadeh, Yanxia Zhang, Liquan Mei, Ashraf Ayubinia, Amin Mosallanezhad, Jingyi Zhang, Machine learning-based photometric classification of galaxies, quasars, emission-line galaxies, and stars, *Monthly Notices of the Royal Astronomical Society*, Volume 527, Issue 3, Pages 4677–4689 2024.

[11] Moonzarin Reza, Galaxy morphology classification using automated machine learning, Astronomy and Computing, Volume 37,100492, ISSN 2213-1337 2021.

[12] Shiker, Mushtak, Multivariate Statistical Analysis. British Journal of Science. 6. 55-66 2012.

[13] Mazarei, A., Sousa, R., Mendes-Moreira, J. et al. Online boxplot derived outlier detection. Int J Data Sci Anal 19, 83–97 2025.

[14] P. Hajibabaee, F. Pourkamali-Anaraki and M. A. Hariri-Ardebili, "An Empirical Evaluation of the t-SNE Algorithm for Data Visualization in Structural Engineering," 20th IEEE International Conference on Machine Learning and Applications (ICMLA), Pasadena, CA, USA, 2021, pp. 1674-1680 2021.

[15] Verdonck, T., Baesens, B., Óskarsdóttir, M. et al. Special issue on feature engineering editorial. Mach Learn 113, 3917–3928 2024.

[16] Starovoitov, V.V. and Golub, Y.I., September. Data normalization in machine learning. In Informatics ,Vol. 18, No. 3, pp. 83-96 2021.

[17] Kotsilieris, T., Anagnostopoulos, I. and Livieris, I.E., Regularization techniques for machine learning and their applications. Electronics, 11(4), p.521 2022.

[18] Hossein Shafizadeh-Moghadam, Fully component selection: An efficient combination of feature selection and principal component analysis to increase model performance, Expert Systems with Applications, Volume 186, 115678, ISSN 0957-4174 2021.

[19] Judith Hurwitz, and Daniel Kirsch, “Machine Learning”, IBM, 2018.

[20] Sha’Abani, M.N.A.H., Fuad, N., Jamal, N. and Ismail, M.F., 2020, March. kNN and SVM classification for EEG: a review. In InECCE2019: proceedings of the 5th international conference on electrical, control & computer engineering, kuantan, pahang, Malaysia, (pp. 555-565). Singapore: Springer Singapore 2019.

[21] Abdullah, D.M. and Abdulazeez, A.M.,Machine learning applications based on SVM classification a review. Qubahan Academic Journal, 1(2), pp.81-90 2021.

[22] Priyanka and Kumar, D.,Decision tree classifier: a detailed survey. International Journal of Information and Decision Sciences, 12(3), pp.246-269 2020.

[23] Niazkar, M., Menapace, A., Brentan, B., Piraei, R., Jimenez, D., Dhawan, P. and Righetti, M., Applications of XGBoost in water resources engineering: A systematic literature review (Dec 2018–May 2023). Environmental Modelling & Software, 174, p.105971 2024.

[24] Dey, R. and Mathur, R., May. Ensemble learning method using stacking with base learner, a comparison. In International conference on data analytics and insights (pp. 159-169). Singapore: Springer Nature Singapore 2023.

[25] Ding, Y., Zhu, H., Chen, R. and Li, R., An efficient AdaBoost algorithm with the multiple thresholds classification. Applied sciences, 12(12), p.5872 2022.

[26] Purwanto, A.D., Wikantika, K., Deliar, A. and Darmawan, S., Decision tree and random forest classification algorithms for mangrove forest mapping in Sembilang National Park, Indonesia. *Remote Sensing*, *15*(1), p.16 2022.

[27] Urbanowicz, R., Zhang, R., Cui, Y. and Suri, P., STREAMLINE: a simple, transparent, end-to-end automated machine learning pipeline facilitating data analysis and algorithm comparison. In *Genetic Programming Theory and Practice XIX* (pp. 201-231). Singapore: Springer Nature Singapore 2023.