Stellar Classification

Milestone: Final Project Report

Group 16

Student 1 Sanidhya Karnik

Student 2 Digvijay Raut

617-407-1206 (Tel of Student 1)

857-492-3195 (Tel of Student 2)

karnik.san@northeastern.edu

raut.di@northeastern.edu

Percentage of Effort Contributed by Student 1: **50%**

Percentage of Effort Contributed by Student 2: **50%**

Signature of Student 1: **Sanidhya Karnik**

Signature of Student 2: **Digvijay Raut**

Submission Date: **04/12/2024**

Project Selection and Problem Definition

In the expansive field of astronomy, the classification and understanding of celestial bodies observed in the night sky have always been paramount. The evolution of advanced telescopic technology and comprehensive digital surveys, notably the Sloan Digital Sky Survey (SDSS), has resulted in the accumulation of extensive datasets on various astronomical objects. The vastness of these datasets presents a formidable challenge: the manual classification of each observed object—be it stars, galaxies, or quasars—is beyond practicality. To circumvent this bottleneck, there's a pressing need for automated classification methodologies. These methods not only promise efficiency in processing astronomical data but also significantly enhance our comprehension of cosmic phenomena.

This project aims to tackle the challenge by developing an automated model capable of accurately classifying astronomical objects into one of three categories: stars, galaxies, or quasars. This classification will be based on an analysis of their spectral and photometric properties, leveraging the rich data available through modern astronomical surveys.

Data Source and Collection

The foundation of this project rests on data sourced from the Sloan Digital Sky Survey (SDSS), renowned for producing the most detailed three-dimensional mappings of the Universe to date. The survey has captured deep multi-color images covering over one-third of the sky and has compiled spectra for more than three million astronomical objects. This data is freely accessible via the SDSS website and its various data release publications.

For this project's purposes, we will use a subset of the SDSS data of 100,000 observations. Each observation is encapsulated by 17 features and is associated with a target class column, making this dataset an excellent candidate for supervised machine learning challenges, especially those focused on classification. The objective is to employ these 17 features to effectively predict the 'class' column, which assigns each observation into one of three categories: star, galaxy, or quasar, thereby facilitating an automated approach to the classification of astronomical objects.

Data source link: https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17

Data Exploration

The correlation matrix serves as a pivotal tool for understanding the interdependencies among variables, thereby guiding data interpretation and decision-making. Through this analysis, we discovered a correlation among the 'u', 'g', 'r', 'i', 'z' columns.

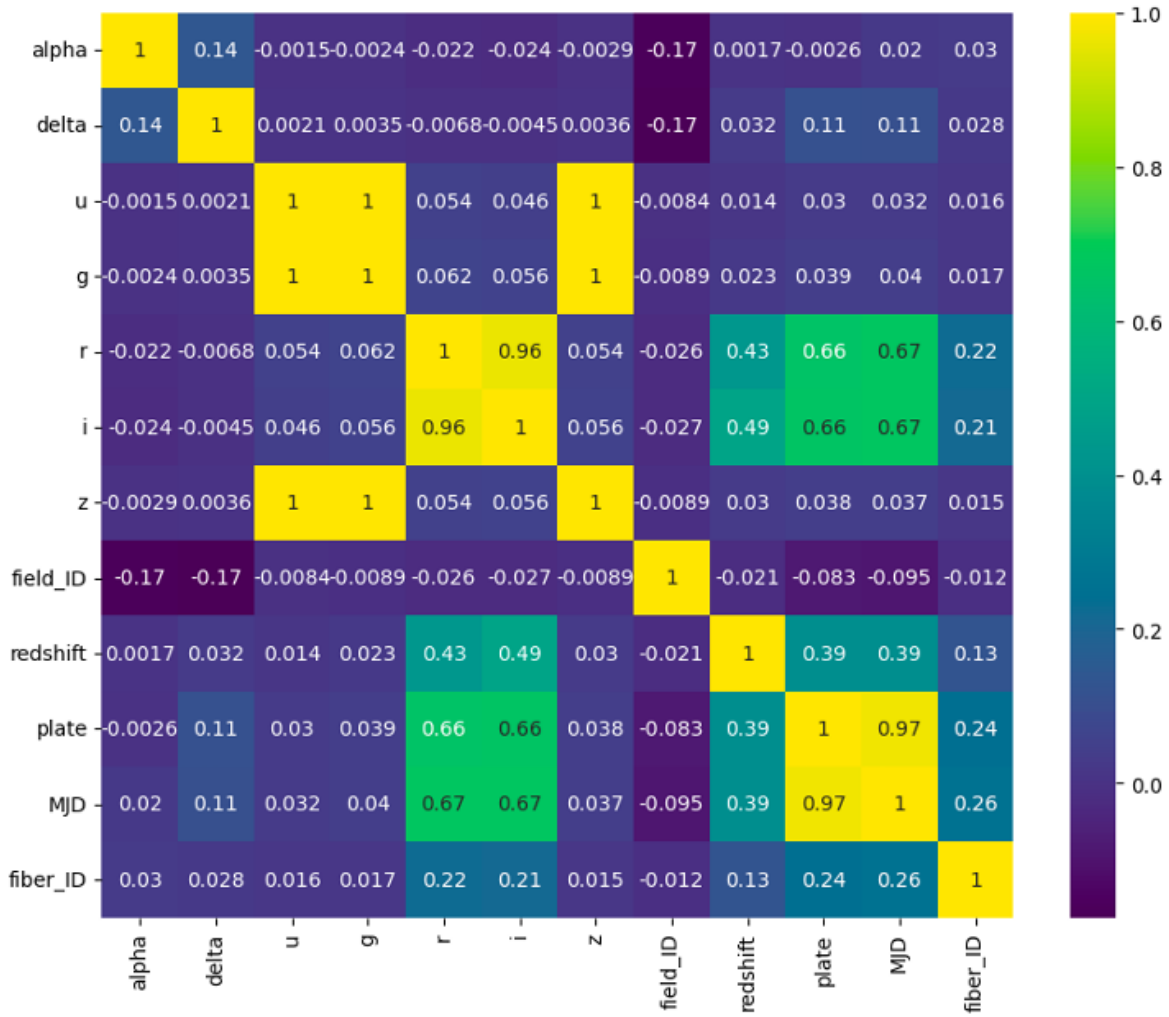

Fig 1

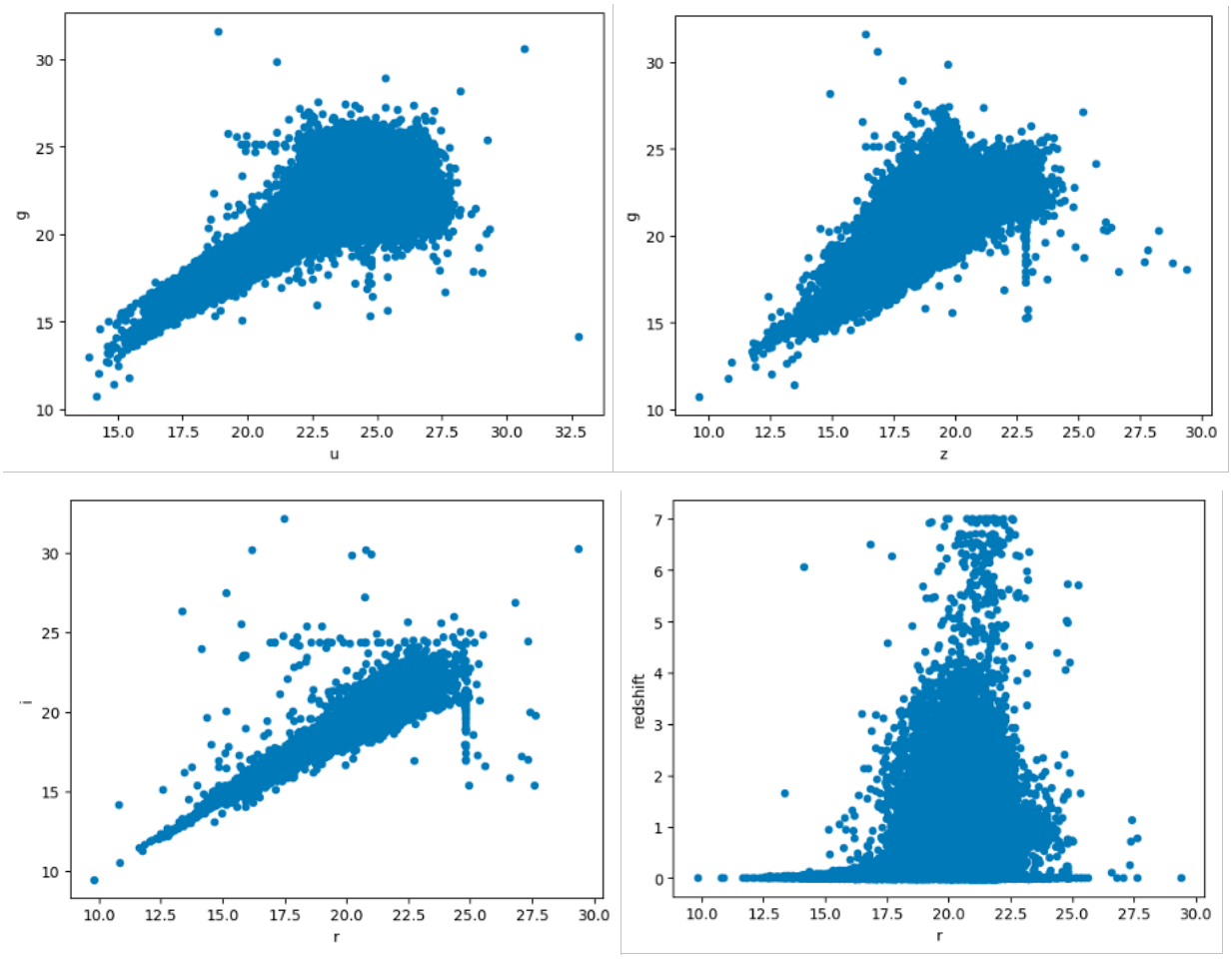

Fig 2

In Fig 2, we visualized scatter plots between ‘g’ vs ‘u’, ‘g’ vs ‘z’ and ‘i’ vs ‘r’ which showed a strong positive correlation whereas ‘redshift’ vs ‘r’ showed a weak positive correlation.

A subsequent box-plot analysis revealed an outlier with an unusually low value (refer Fig 3a), leading to its removal. Fib 3b shows a better readable boxplot after the outlier was removed. Further outlier removal was conducted using sklearn.neighbours library, resulting in a refined dataset of 89,999 records.

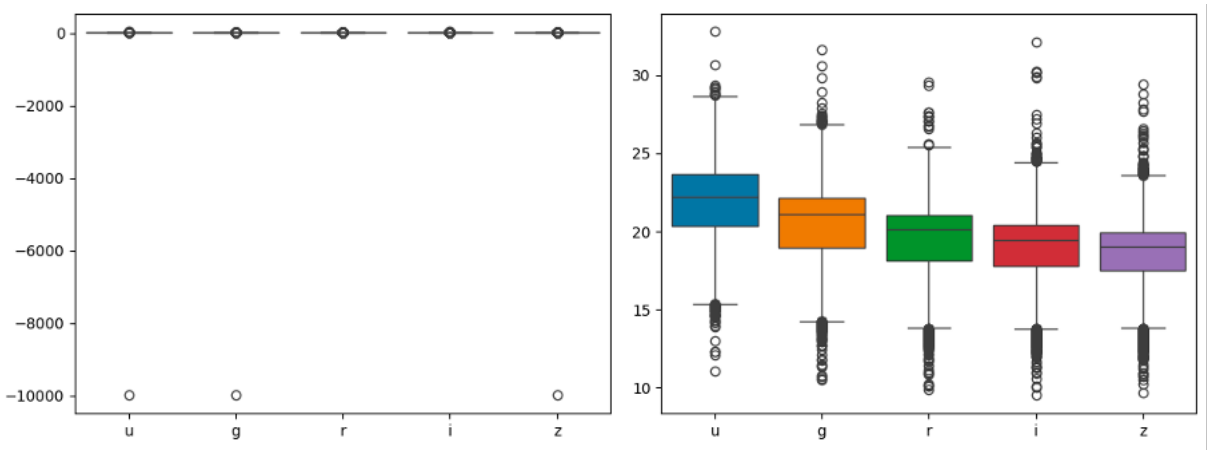

Fig 3 (a, b)

Given the presence of three target variables, we employed a bar graph to visualize their distribution, identifying a predominant class among them.

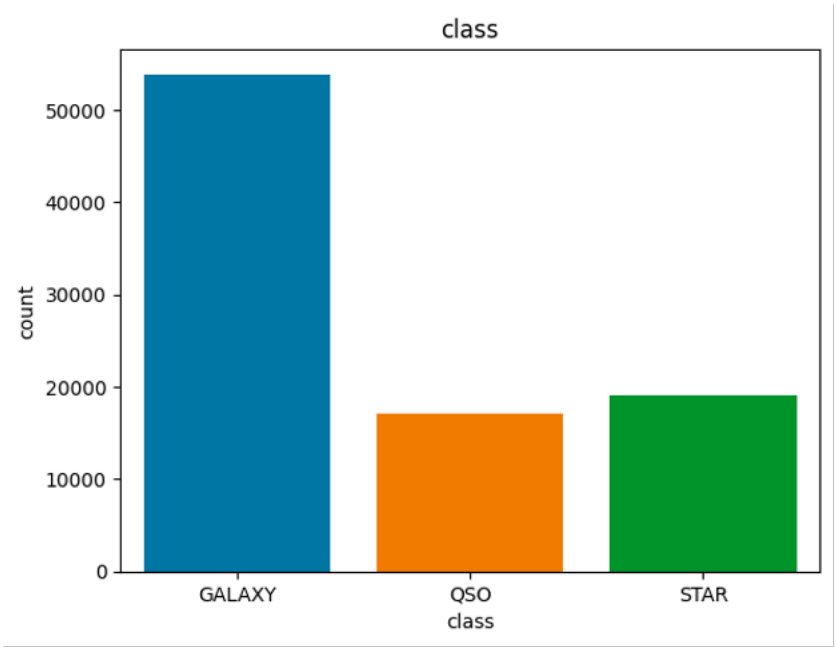

Fig 4(a)

A Principal Component Analysis (PCA) was executed on the dataset and the first 7 components that encapsulate ~95% of variance, were picked to be the principal components. This process effectively reduced the dataset from 17 features to 7 and from 100,000 records to 89,999.

Variance distribution: [49.62369147 12.06979656 9.07481334 8.22269234 7.51893126 7.07511107 6.41496395]

Variance captured: 95.22%

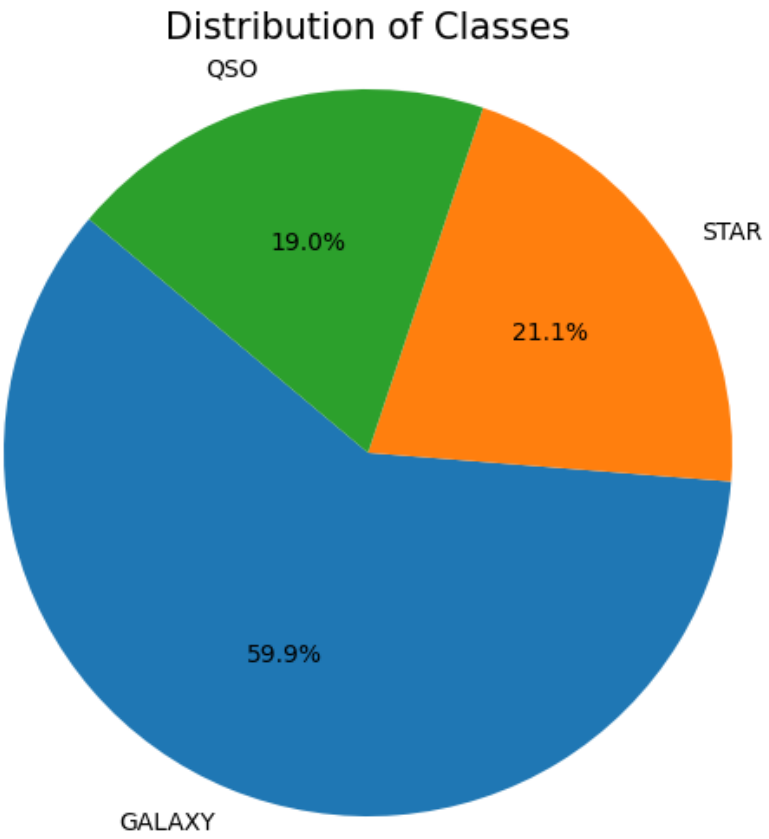

Fig 4(b)

Statistics of the final dataset acquired after performing Principal component analysis can be seen in Fig 5 below:

	pc1	pc2	pc3	pc4	pc5	pc6	pc7
count	8.999900e+04	8.999900e+04	8.999900e+04	8.999900e+04	8.999900e+04	8.999900e+04	8.999900e+04
mean	3.063263e-17	1.894802e-17	-8.052907e-18	2.968523e-17	3.663283e-17	-6.947606e-18	6.316006e-19
std	2.381216e+00	1.174368e+00	1.018294e+00	9.693070e-01	9.268990e-01	8.991268e-01	8.561531e-01
min	-7.847770e+00	-3.328760e+00	-3.902767e+00	-3.155973e+00	-2.423295e+00	-2.902796e+00	-4.183111e+00
25%	-1.785001e+00	-8.192642e-01	-6.860848e-01	-7.155897e-01	-6.442044e-01	-5.845256e-01	-5.242312e-01
50%	-6.435983e-01	-1.921105e-01	-6.152306e-02	1.780223e-02	-1.312410e-01	-1.426135e-02	-1.002526e-02
75%	1.943868e+00	6.127270e-01	6.231924e-01	7.411645e-01	5.348122e-01	5.122695e-01	4.992081e-01
max	9.929775e+00	4.970803e+00	5.786991e+00	4.250988e+00	5.855950e+00	3.678938e+00	6.149007e+00

Fig 5

In the final stage, we examined a correlation matrix of the principal components, revealing negligible correlation among them, indicating successful dimensionality reduction.

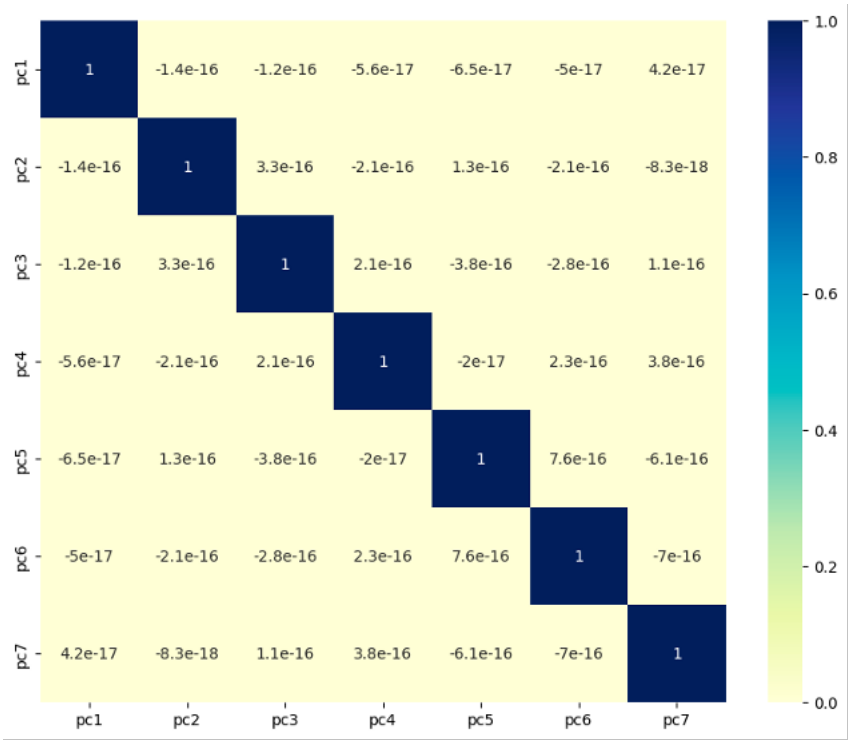

Fig 6

Dimension reduction and variable selection

Upon analysis, we determined that the columns 'obj_ID', 'run_ID', 'rerun_ID', 'cam_col', and 'spec_obj_ID' do not influence the target class in a meaningful scientific manner. Notably, 'rerun_ID' remains constant across the dataset, rendering its presence in the model redundant. By excluding these columns, we are left with 13 relevant columns for our analysis.

Model exploration and model selection

In our project, which aims to categorize astronomical entities into three distinct groups—stars, galaxies, and quasars—by analyzing their spectral and photometric attributes, a variety of machine learning models are applicable. The following outlines some of the prevalent models for classification tasks, assessing their relevance and effectiveness for our challenge:

Logistic Regression:

Despite its name, logistic regression is a linear model suitable for binary classification tasks. However, it can be extended to multi-class classification using techniques like one-vs-rest or SoftMax regression. For this problem, logistic regression might not be the best choice because it's a simple linear model and may not capture complex relationships between features and classes.

Decision Trees:

Decision trees are intuitive and can handle both numerical and categorical data. They can capture nonlinear relationships and interactions between features. However, decision trees tend to overfit the data, especially when the tree depth is not properly controlled.

Random Forest:

Random Forest is an ensemble learning method based on decision trees. It constructs multiple decision trees during training and outputs the mode of the classes (classification) or the mean prediction (regression) of the individual trees. Random Forest generally performs well in practice, handles high-dimensional data well, and is less prone to overfitting compared to individual decision trees.

Support Vector Machines (SVM):

SVM is a powerful supervised learning algorithm used for classification tasks. It works well in high-dimensional spaces and is effective in cases where the number of dimensions exceeds the number of samples. SVM aims to find the hyperplane that best separates the classes in the feature space. SVM can be adapted for multi-class classification using techniques like one-vs-one or one-vs-rest. However, SVM might not be the best choice for very large datasets due to its training time complexity.

Gradient Boosting Machines (GBM):

GBM is an ensemble learning technique that builds a strong learner by combining multiple weak learners (typically decision trees) sequentially. It builds the model in a stage-wise fashion and tries to correct the errors of the previous models. GBM is known for its high predictive accuracy and ability to handle complex relationships in the data.

Neural Networks:

Neural networks, particularly deep learning architectures, have shown remarkable success in various domains, including image recognition and natural language processing. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) are commonly used for image and sequential data, respectively. For this problem, a deep learning approach, especially with architectures tailored for tabular data, could be explored.

k-Nearest Neighbors (k-NN):

k-NN is a simple and intuitive classification algorithm that can work well with reduced-dimensional data. It makes predictions based on the majority class among the k-nearest neighbors in the feature space.

Gaussian Naive Bayes:

Naive Bayes classifier assumes that features are conditionally independent given the class label. Although this assumption may not hold after PCA, Gaussian Naive Bayes can still be applied and can perform well, especially if the features are approximately normally distributed.

Given the characteristics of our problem (classification based on spectral and photometric characteristics) and the dataset size (100,000 observations), models like **Random Forest, Gradient Boosting Machines, k-NN, and Neural Networks** could be promising choices.
Experimentation and model evaluation on a validation set would ultimately determine which model performs the best for this specific task and dataset.

Model performance evaluation

In this part, we provide a comparative analysis of four distinct machine learning models: Gradient Boosting, Random Forest, Neural Network, and k-Nearest Neighbors (k-NN). These models were evaluated based on their performance in a classification task, using four key metrics: Accuracy, Precision, Recall, and F1 Score.

Gradient Boosting

Gradient Boosting exhibited an accuracy of 81.39%, precision of 81.17%, recall of 81.39%, and an F1 score of 80.04%. This ensemble learning method builds models sequentially, each correcting its predecessor, which is especially effective for handling bias and variance in data. However, its performance, while robust, was not the top among the models assessed, possibly due to its sensitivity to overfitting and the complex nature of the data. Confusion matrix for Gradient Boosting is as follows:

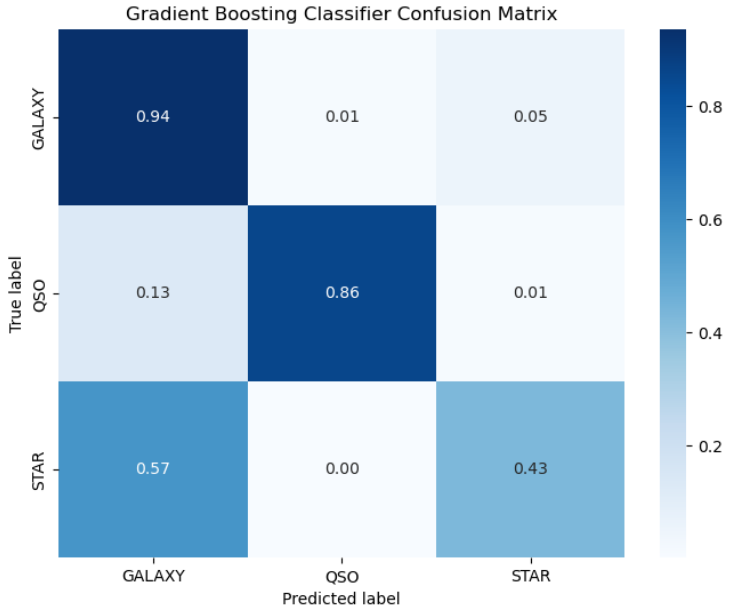

Fig 7

Advantages:

- **High Performance:** Gradient Boosting often provides high accuracy and is effective for complex datasets with non-linear relationships.
- **Flexibility:** It can optimize on different loss functions and provides several hyperparameter tuning options that can make the model robust.
- **Handling of Heterogeneous Features:** Well-suited for datasets where features have different types of data scales and distributions.

Disadvantages:

- **Computational Complexity:** Training can be time-consuming due to the sequential nature of boosting.
- **Overfitting:** Without proper tuning and regularization, Gradient Boosting models can easily overfit to the training data.
- **Hyperparameter Sensitivity:** Requires careful tuning of hyperparameters to avoid overfitting and to optimize performance.

Random Forest

The Random Forest model showed improved performance with an accuracy of 87.86%, precision of 88.00%, recall of 87.86%, and an F1 score of 87.46%. As an ensemble of decision trees, Random Forest reduces overfitting risks and handles bias and variance more effectively than a single decision tree. Its higher scores across all metrics compared to Gradient Boosting suggest it is better suited for this classification task, likely due to its ability to manage complex interactions and dependencies in the data. Confusion matrix for Random Forest is as follows:

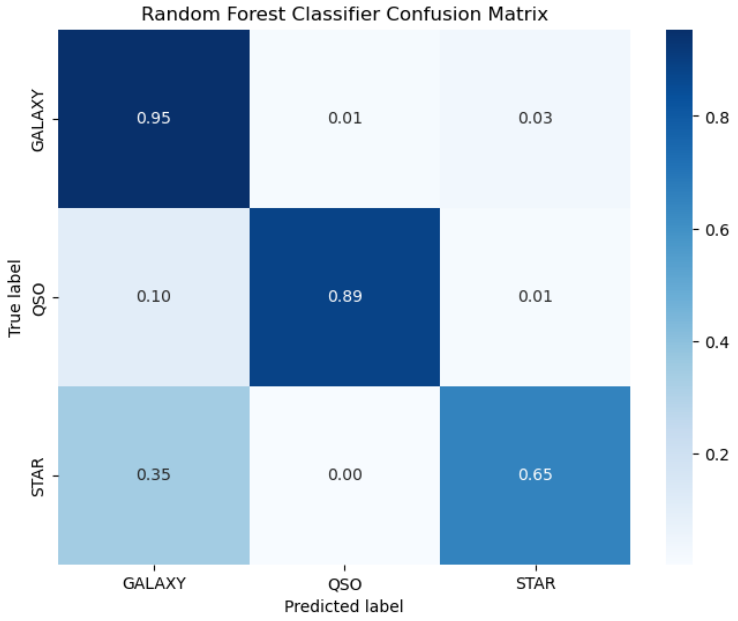

Fig 8

Advantages:

- **Versatility:** Effective for both classification and regression tasks and works well with both categorical and continuous variables.
- **Robust to Overfitting:** The ensemble approach, building multiple decision trees, helps in reducing the risk of overfitting.
- **Importance of Features:** Offers insights into feature importance, which can be valuable for understanding the data.

Disadvantages:

- **Model Size:** Can require a lot of memory as it builds numerous large trees.
- **Computation Time:** Training time can be long, especially with large datasets and a high number of trees.
- **Performance:** While it generally performs well, it might not achieve the high accuracy levels of more complex models like Gradient Boosting in some cases.

Neural Network

The Neural Network achieved the highest scores among the evaluated models, with an accuracy of 89.85%, precision of 89.89%, recall of 89.85%, and an F1 score of 89.68%. Neural networks are powerful tools for modeling complex relationships through their deep layers and many parameters. Their superior performance in this analysis underscores their capability in handling nonlinearities and interactions in data, albeit at the cost of requiring extensive computational resources and data for training. Confusion matrix for Neural Network is as follows:

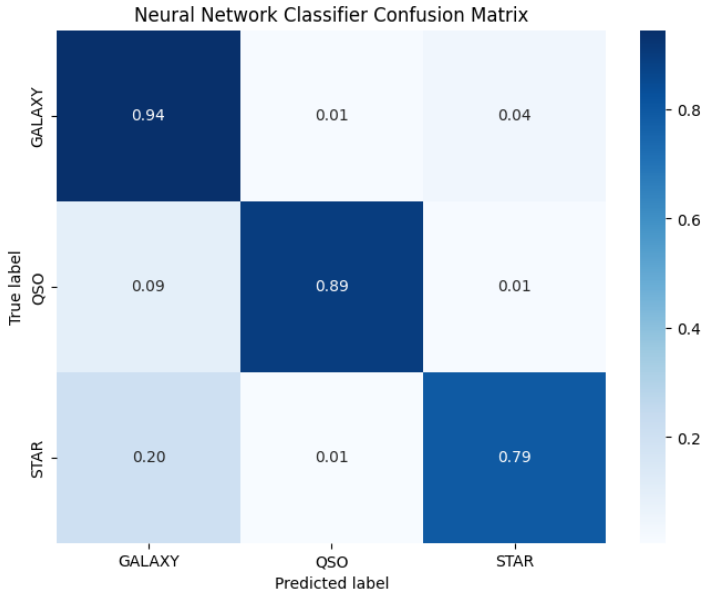

Fig 9

Advantages:

- **Flexibility:** Capable of modeling complex and non-linear relationships due to its layered structure.
- **Scalability:** Works well with large datasets and can benefit significantly from GPU acceleration.
- **Generalization:** With proper regularization, neural networks can generalize well to unseen data.

Disadvantages:

- **Opacity:** Often referred to as "black boxes" because their decision-making process is not as interpretable as simpler models.
- **Computational Resources:** Training large neural networks requires significant computational resources and time.
- **Hyperparameter Tuning:** Requires careful tuning of numerous parameters, and the training process can be sensitive to the initial settings.

k-Nearest Neighbors (k-NN)

k-Nearest Neighbors presented an accuracy of 87.27%, precision of 87.52%, recall of 87.27%, and an F1 score of 86.79%. k-NN is a simple, intuitive method that classifies samples based on the majority vote of their neighbors. While it is less computationally intensive during training, its performance is slightly below that of Random Forest and significantly lower than the Neural Network. This might be due to its dependency on a suitable distance metric and the challenge of choosing an optimal number of neighbors. Confusion matrix for k-NN is as follows:

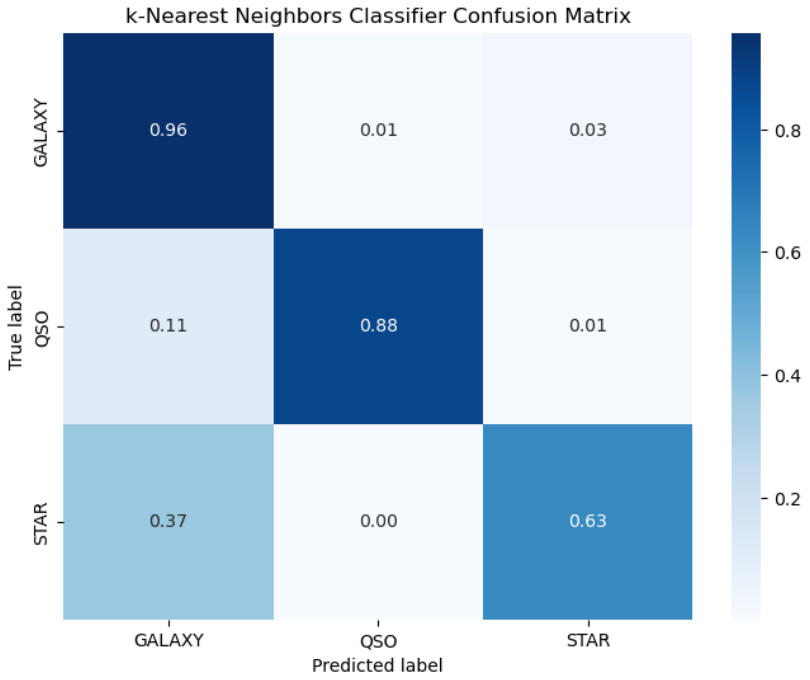

Fig 10

Advantages:

- **Simplicity:** One of the simplest machine learning algorithms to understand and implement.
- **Flexibility:** No assumption about the data distribution, making it versatile for various types of data.
- **Ease of Use:** Does not require extensive training or parameter tuning in its basic form.

Disadvantages:

- **Scalability:** Computationally expensive and slow with large datasets because it searches for the nearest neighbors for each query.
- **Dimensionality:** Performs poorly in high-dimensional spaces due to the curse of dimensionality.
- **Sensitive to Noisy or Irrelevant Features:** Performance can significantly degrade with the presence of noisy or irrelevant features, requiring feature selection or dimensionality reduction techniques.

Project Results

The analysis demonstrates that Neural Networks offer the highest performance for this classification task, closely followed by Random Forest and k-Nearest Neighbors, with Gradient Boosting trailing slightly behind. Each method's effectiveness varies based on the specific characteristics of the data and the computational resources available. Therefore, the choice of model should consider the trade-off between accuracy and computational efficiency, alongside the specific requirements and constraints of the application.

Below plot contains area under curve for all four methods we used:

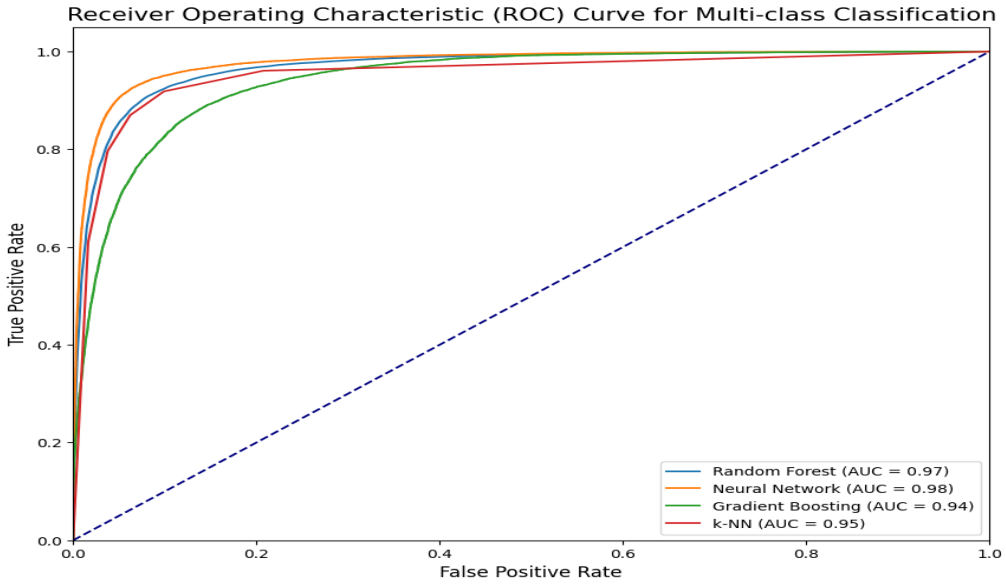

Fig 11

The table below shows accuracy, precision, recall and F1 score for all four models:

	Accuracy	Precision	Recall	F1 Score
Random Forest	0.877000	0.878330	0.877000	0.872883
Neural Network	0.902556	0.902312	0.902556	0.901710
Gradient Boosting	0.813944	0.811650	0.813944	0.800372
k-NN	0.872667	0.875198	0.872667	0.867865

Fig 12

Impact of project

The project focused on classifying astronomical objects into stars, galaxies, and quasars based on their spectral and photometric characteristics creates significant value in several key areas:

Advancing Astronomical Knowledge

Efficiency in Classification: Automating the classification process allows for the rapid and accurate categorization of astronomical objects. This efficiency is crucial given the massive volumes of data generated by telescopes and surveys like the SDSS. By quickly classifying objects, astronomers can more readily identify objects of interest for further study.

Deepening Understanding: With accurate classifications, researchers can deepen their understanding of the universe's structure, the life cycle of stars, the distribution of galaxies, and the nature of quasars. This foundational knowledge is vital for further exploration and theoretical development in astrophysics.

Public Engagement: The project's ability to categorize and make sense of the cosmos contributes to public interest and engagement with science. It highlights the role of technology and data science in expanding our understanding of the universe, thereby fostering a greater appreciation for scientific research.