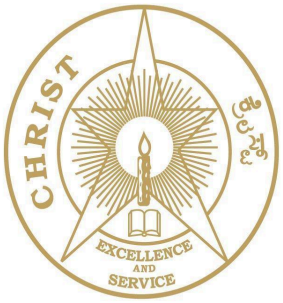


CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

II Trimester MSc (AI & ML)
Advanced Machine Learning

Department of Computer Science

STELLAR OBJECT CLASSIFICATION AND CLUSTERING: STARS, GALAXIES, AND QUASARS WITH MACHINE LEARNING

by

Anushka Mazumdar (2348505)
Soheli Paul (2348561)
Tanisha Das (2348569)

25th January 2024

CHRIST
(DEEMED TO BE UNIVERSITY)
BANGALORE · INDIA

CERTIFICATE

This is to certify that the report titled ***Stellar Object Classification and Clustering: Stars, Galaxies, and Quasars with Machine Learning*** *is a bona fide record of work done by* ***Anushka Mazumdar (2348505)***, ***Soheli Paul (2348561) and Tanisha Das (2348569)*** *of CHRIST (Deemed to be University), Bangalore, in partial fulfillment of the requirements of II Trimester of Msc Artificial Intelligence and Machine Learning during the year 2023-24.*

Valued-by: (Evaluator Name & Signature)

1.

Course Teacher

2.

Date of Exam: 25/01/2024

TABLE OF CONTENTS

TEAM DETAILS

Reg. no	Name	Summary of tasks performed
2348505	**Anushka Mazumdar**	**Preprocessing Tasks:** Missing Value Detection, SMOTE **Classification Tasks:** Random Forest, XGBoost **Clustering Tasks:** BIRCH
2348561	**Soheli Paul**	**Preprocessing Tasks:** Label Encoding, Outlier Detection **Classification Tasks:** Decision Tree, Bagging Classifier **Clustering Tasks:** K-Means Clustering
2348569	**Tanisha Das**	**Data Visualization, Preprocessing Tasks:** Feature Scaling, Feature Selection **Classification Tasks:** Naive Bayes, Light GBM

1. ABSTRACT

This research constitutes a comprehensive investigation into the intricate realm of stellar object classification. Leveraging a diverse suite of machine learning methodologies, including k-nearest Neighbors (kNN), Decision Tree Classifier, Bagging Classifier, Gaussian Naive Bayes, Random Forest Classifier, and XGBoost Classifier, the study endeavors to unravel the nuances and complexities inherent in distinguishing between stars, galaxies, and quasars. The dataset, meticulously curated and preprocessed, is the foundation for this exploration. Extensive data preprocessing procedures, encompassing cleaning, normalization, and feature engineering, lay the groundwork for a robust analysis. Additionally, exploratory data analysis techniques have been applied to uncover hidden patterns, trends, and outliers, providing valuable insights into the underlying structures within the dataset. Following the initial classification efforts, where the class variable delineates the stellar objects, a subsequent phase involves the removal of this categorical label. This unique approach allows for assessing classification techniques in an unsupervised setting, illuminating the potential for these models to discern inherent patterns without predefined class distinctions. Furthermore, the investigation extends beyond classification into the realm of clustering. Employing k-means clustering on the feature-rich dataset aims to uncover latent groupings and relationships among stellar objects, fostering a more holistic understanding of celestial phenomena. In summation, this study contributes to the advancement of stellar classification methodologies and underscores the significance of robust data preprocessing, exploratory analysis, and the fusion of supervised and unsupervised learning paradigms in enhancing our comprehension of celestial bodies.

2. INTRODUCTION

The cosmos, with its myriad celestial bodies, has long captivated the imagination of astronomers and astrophysicists. In the pursuit of understanding the vast expanse of the universe, advancements in machine learning (ML) have become invaluable tools for classifying and unraveling the mysteries of stellar objects. This research embarks on a journey to explore and harness the power of various ML classification techniques, including k-nearest Neighbors (kNN), Decision Tree Classifier, Bagging Classifier, Gaussian Naive Bayes, Random Forest Classifier, and XGBoost Classifier.

In astronomy, stellar classification is the classification of stars based on spectral characteristics. The classification scheme of galaxies, quasars, and stars is one of the most fundamental in astronomy. The early cataloging of stars and their distribution in the sky has led to the understanding that they make up our galaxy. Following the distinction that Andromeda was a separate galaxy from our own, numerous galaxies began to be surveyed as more powerful telescopes were built. This dataset aims to classify stars, galaxies, and quasars based on spectral characteristics.

Stars: Stars are celestial objects that emit light and heat due to nuclear reactions occurring in their cores. These luminous plasma spheres are fundamental building blocks of the universe and play a crucial role in various astrophysical processes.

Galaxy: A galaxy is a vast, gravitationally bound system of stars, stellar remnants, interstellar gas, dust, dark matter, and other celestial objects, all bound together by gravity. Galaxies are fundamental building blocks of the universe and come in various shapes and sizes.

Quasars: Quasars, short for "quasi-stellar radio sources," are extremely bright and energetic celestial objects located at the centers of distant galaxies. These objects exhibit unique characteristics that set them apart from other astronomical phenomena.

The dataset under scrutiny encapsulates the multifaceted nature of stellar phenomena, featuring diverse celestial entities such as stars, galaxies, and quasars. Before delving into ML algorithms, the dataset undergoes meticulous data preprocessing. This preparatory phase involves data

preprocessing, standardization, and strategic feature engineering to enhance the dataset's quality and reveal latent patterns.

To deepen our insights into the dataset's characteristics, exploratory data analysis techniques are applied, unraveling underlying trends, correlations, and anomalies. Armed with a comprehensive understanding of the dataset, the study examines the efficacy of various ML classification techniques. Initially, the classification task is approached supervised, where the class variable serves as a guide for training the models.

A distinctive facet of this research lies in the subsequent removal of the class variable, initiating an unsupervised exploration. Here, the models are tasked with discerning patterns without explicit class distinctions, offering a novel perspective on the dataset's inherent structures.

In addition to classification endeavors, the study extends its purview to clustering methodologies. By applying k-means clustering, the research seeks to unearth latent groupings and relationships among stellar objects, adding a layer of complexity to the analysis.

This interdisciplinary approach, blending astronomy with machine learning, holds promise for advancing our understanding of celestial bodies and fostering new insights into the universe's intricacies. Through this convergence of astrophysics and data science, we journey to decipher the cosmic tapestry, one algorithm at a time**.**

3. DATA PRE-PROCESSING AND EXPLORATION

In the pursuit of unraveling the celestial mysteries embedded within the dataset of stellar objects, a pivotal initial step involves meticulous data pre-processing. This preparatory phase ensures the dataset's integrity, enhances its quality, and enables subsequent machine-learning analyses to yield meaningful insights.

The data pre-processing pipeline comprises several key steps. Initially, data-cleaning procedures are employed to address missing values, outliers, and inconsistencies. This step ensures the dataset's homogeneity and prepares it for subsequent transformations. Normalization techniques are then applied to standardize the scale of features, eliminating potential biases introduced by varying units or scales. This process is crucial for algorithms sensitive to the magnitude of input features, ensuring a fair and unbiased representation.

Feature engineering is a strategic aspect of data pre-processing, where new features are crafted or existing ones are transformed to extract more meaningful information. This step involves selecting relevant features, creating composite variables, or applying mathematical transformations to enhance the dataset's representational power.

With a pre-processed dataset in hand, the exploration phase begins. Exploratory Data Analysis (EDA) techniques are deployed to unveil underlying patterns, trends, and relationships within the stellar dataset. Statistical measures, visualizations, and correlation analyses contribute to a holistic understanding of the dataset's characteristics.

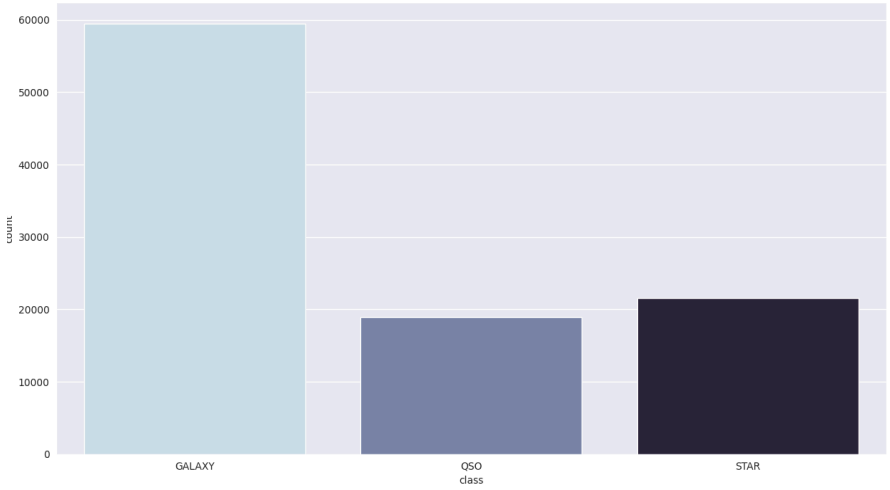
Visualization tools such as scatter plots, histograms, and heatmaps offer intuitive insights into the distribution of features, identifying potential clusters or groupings. Correlation matrices provide a glimpse into the relationships between variables, guiding subsequent feature selection and model training decisions. This dual-phase data pre-processing and exploration approach sets the stage for a robust machine-learning analysis. The synergy between preparing the dataset and gaining insights through exploratory analysis lays the foundation for accurate classification and clustering of stellar objects, propelling our understanding of the cosmic tapestry to new heights.

3.1 DATA UNDERSTANDING AND EXPLORATION

The data consists of 100,000 observations of space taken by the SDSS (Sloan Digital Sky Survey). Every observation is described by 17 feature columns and 1 class column, identifying it as a star, galaxy, or quasar.

- obj_ID = Object Identifier, the unique value that identifies the object in the image catalog used by the CAS
- alpha = Right Ascension angle (at J2000 epoch)
- delta = Declination angle (at J2000 epoch)
- u = Ultraviolet filter in the photometric system
- g = Green filter in the photometric system
- r = Red filter in the photometric system
- i = Near Infrared filter in the photometric system
- z = Infrared filter in the photometric system
- run_ID = Run Number used to identify the specific scan
- rereun_ID = Rerun Number to specify how the image was processed
- cam_col = Camera column to identify the scanline within the run
- field_ID = Field number to identify each field
- spec_obj_ID = Unique ID used for optical spectroscopic objects (this means that 2 different observations with the same spec_obj_ID must share the output class)
- class = object class (galaxy, star or quasar object)
- redshift = redshift value based on the increase in wavelength
- plate = plate ID, identifies each plate in SDSS
- MJD = Modified Julian Date, used to indicate when a given piece of SDSS data was taken
- fiber_ID = fiber ID that identifies the fiber that pointed the light at the focal plane in each observation

DISTRIBUTION OF CLASS VARIABLES



As we can see Galaxy has the highest number of samples, approximately compassing 59,445 samples, whereas Quasar has 21,594 samples Star constitutes 18,961 samples.

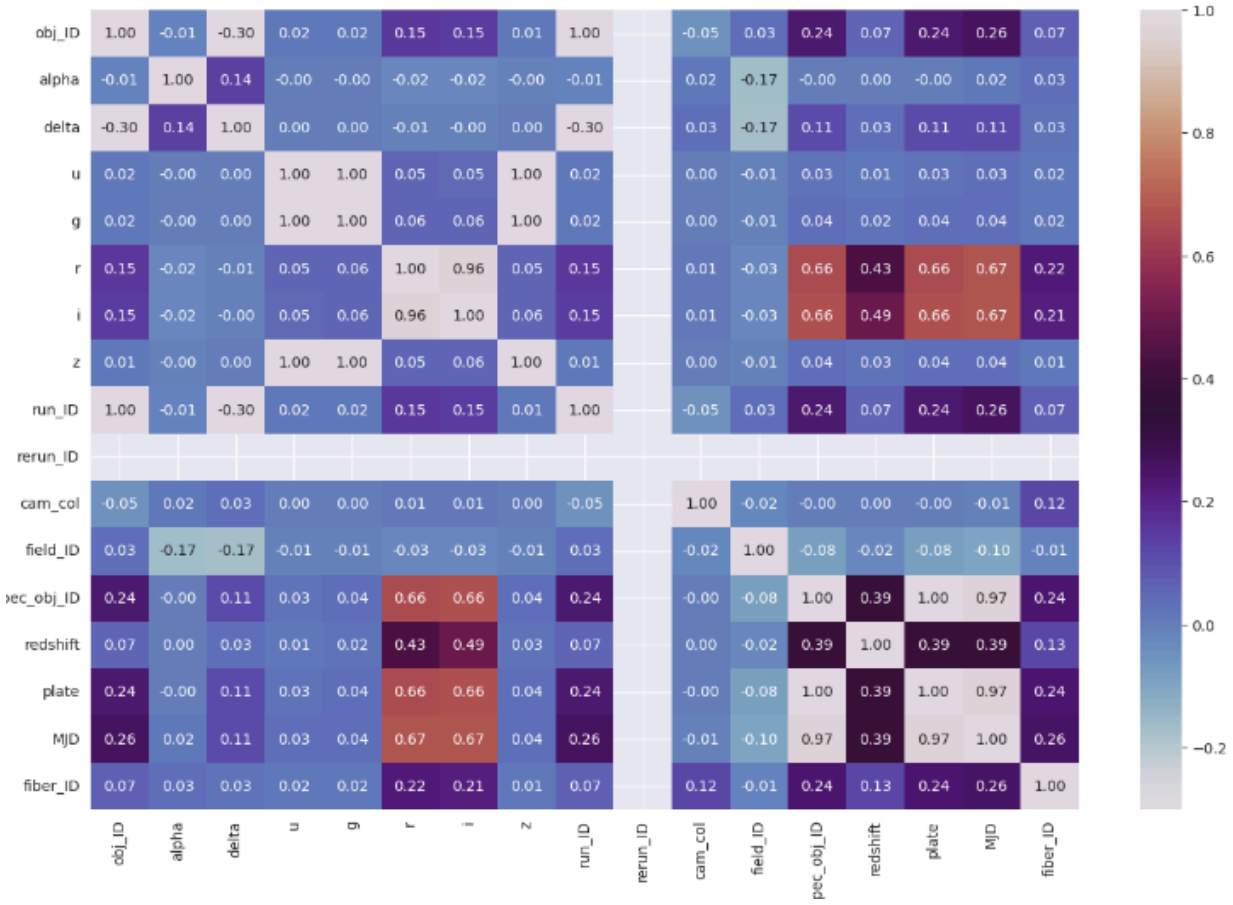
CORRELATION AMONG FEATURE VARIABLES

Understanding the interrelationships between feature variables is a pivotal aspect of data exploration in stellar object classification. Correlation analysis provides insights into how features co-vary, offering valuable information for feature selection, model interpretation, and identifying potential redundancies within the dataset. Correlation is typically measured using statistical metrics, with the Pearson correlation coefficient being a commonly employed method. The coefficient ranges from -1 to 1, where:

- 1 indicates a strong positive correlation,
- -1 indicates a strong negative correlation, and
- 0 indicates no linear correlation.

Highly correlated features may carry redundant information. Identifying and retaining only one representative from a correlated set can enhance model simplicity and interpretability. Understanding correlations helps in selecting features that contribute independently to the

models. Highly correlated features can potentially introduce multicollinearity, impacting the stability and interpretability of certain models.



Many features such as Ultraviolet Filter, Green Filter, and Infrared Filter have a perfect correlation of 1 among them. The correlation is not evenly distributed. Some variables depict a high correlation with each other, whereas some depict a very low correlation.

3.2 DATA CLEANING AND HANDLING MISSING VALUES

MISSING VALUE DETECTION

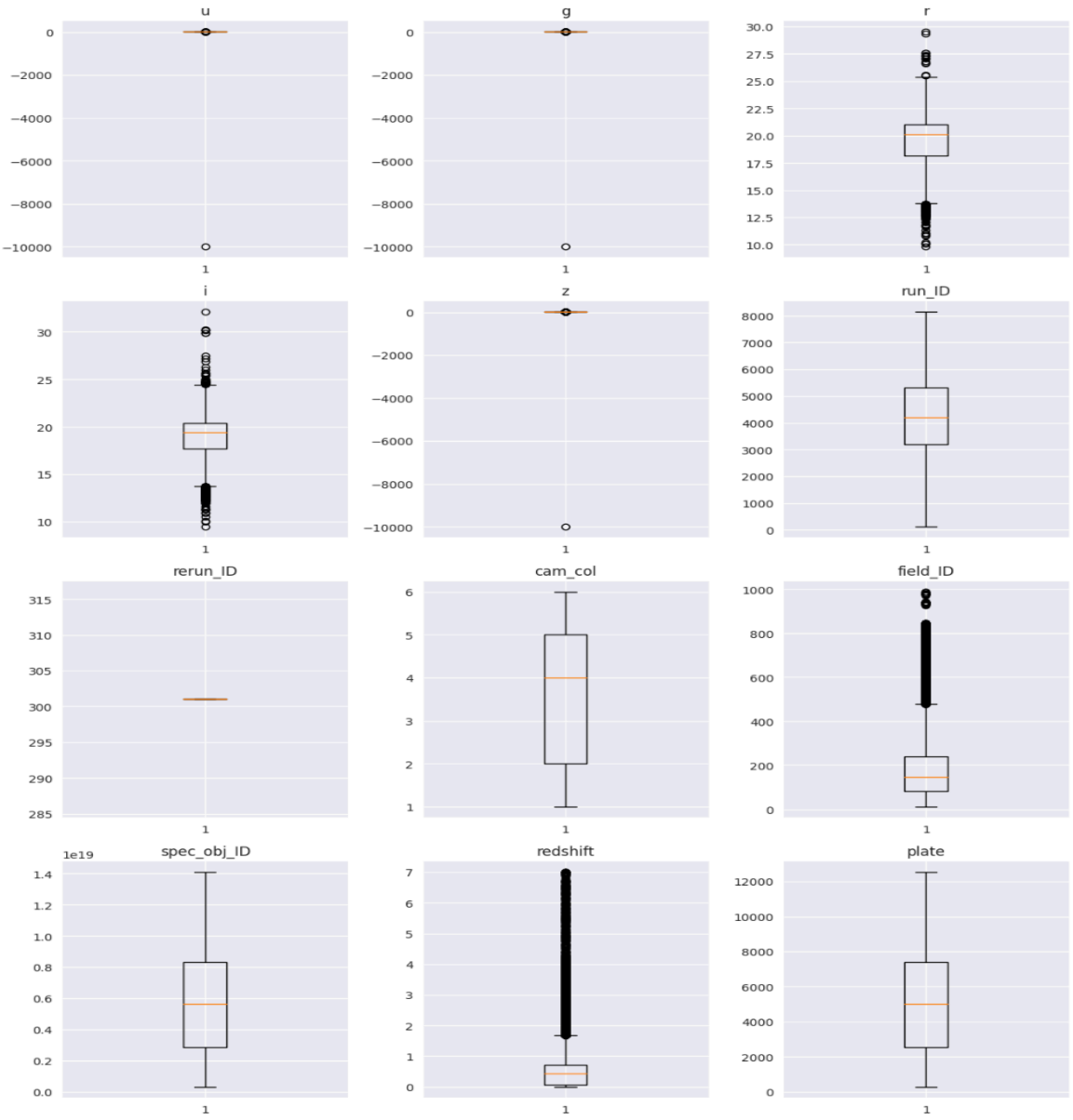
Upon careful examination, we discover that none of the features contain any missing values.

	No. of Missing Values
obj_ID	0
alpha	0
delta	0
u	0
g	0
r	0
i	0
z	0
run_ID	0
rerun_ID	0
cam_col	0
field_ID	0
spec_obj_ID	0
class	0
redshift	0
plate	0
MJD	0
fiber_ID	0

So, we don’t need to apply any imputation methods to treat missing values.

OUTLIER DETECTION

In the realm of stellar object classification, outlier detection assumes a critical role in ensuring the fidelity and reliability of the dataset. Outliers, anomalies, or erroneous data points can significantly impact the performance of machine learning models and distort the accuracy of subsequent analyses. Outlier detection involves identifying and handling data points that deviate substantially from the expected or typical patterns within the dataset. Outliers may stem from measurement errors, instrument anomalies, or even genuinely unique celestial phenomena.



In our dataset, most of the features don't have any outliers except a few, namely columns like the red filter in the photometric system, near-infrared filter in the photometric system, field_id to identify each field, and redshift.

LABEL ENCODING

In the context of stellar object classification, the label encoding of the class variable is a pivotal step that transforms categorical labels such as 'stars,' 'galaxies,' and 'quasars' into numerical representations. This numerical encoding is instrumental in facilitating the application of various machine-learning algorithms that require numeric input.

Consider the following label encoding scheme:

- 'Galaxies' is encoded as 0,
- 'Stars' is encoded as 1, and
- 'Quasars' is encoded as 2.

This encoding preserves the categorical distinctions while providing a format that machine learning models can readily interpret. The choice of encoding strategy is influenced by the classification task's nature and the algorithm's characteristics.

It's important to note that while label encoding is a suitable preprocessing step, it assumes an ordinal relationship among the classes. In the context of stellar classification, this might imply an unintentional order that does not align with the intrinsic properties of celestial objects. In scenarios with inappropriate ordinal implications, one might explore alternative encoding techniques like one-hot encoding.

Nevertheless, label encoding is a pragmatic solution for efficiently preparing the class variable for training diverse machine learning models, contributing to the seamless integration of categorical data into the analytical pipeline.

3.3 DATA INTEGRATION AND FEATURE ENGINEERING

SMOTE

In a classification problem, class imbalance occurs when one or more classes have significantly fewer instances than others. This can pose challenges for machine learning models as they may become biased towards the majority class, leading to suboptimal performance, especially for the minority classes. In our dataset as well, we have faced a similar issue where the GALAXY category has the highest number of samples, compared to STARS and QUASARS.

0	59445
2	21594
1	18961

Traditional machine learning models might struggle to accurately predict minority class instances because they have fewer examples to learn from during training. This can result in models being overly influenced by the majority class, leading to lower sensitivity or recall for minority classes.

SMOTE (Synthetic Minority Oversampling Technique) is a resampling technique designed to address class imbalance by generating synthetic samples for the minority class. Rather than duplicating existing instances, SMOTE creates synthetic examples by interpolating between existing minority class instances.

SMOTE works in the following way:

- Identifying Minority Instances: SMOTE first identifies minority class instances in the dataset.
- Nearest-Neighbor Interpolation: For each minority instance, SMOTE selects its k nearest neighbors in the feature space.
- Creating Synthetic Instances: Synthetic instances are generated by interpolating between the minority instance and its nearest neighbors. This involves combining the features of the minority instance with those of its neighbors.

This is the output after resampling:

0	59445
1	59445
2	59445

While SMOTE is a powerful tool, its effectiveness depends on the dataset's characteristics. In some cases, exploring different resampling techniques or a combination of strategies might be beneficial.

FEATURE SCALING

Feature scaling is a crucial preprocessing step in machine learning that involves transforming the numerical features of a dataset to a standard scale. The goal is to ensure that all features contribute equally to the model's training process and prevent any particular feature from dominating due to differences in their scales.

We have performed ***Standardization*** on our dataset. It transforms features with a mean of 0 and a standard deviation 1. It is preferred when features have varying scales, and the presence of outliers needs to be considered. Distance-based models (e.g., k-NN) need feature scaling as these models rely heavily on distance calculations. Scaling does not affect the relationship between features but enhances model performance.

$$Z = \frac{X - \mu}{\sigma}$$

All the feature variables in our dataset have been scaled down, and the function StandardScaler() from scikit-learn has been used to standardize them. The values will range from -1 to 1.

FEATURE SELECTION

Feature selection is a crucial step in the machine learning pipeline to identify and retain the most relevant features for model training. Correlation analysis is a widely used technique for selecting features based on their relationships with the target variable and each other. It measures the strength and direction of a linear relationship between two variables. In the context of feature selection, the focus is on the correlation between features and, if applicable, between features and the target variable. Features with high correlation with the target variable are more informative for predictive modeling.

$$r = \frac{n(\Sigma xy) - (\Sigma x)(\Sigma y)}{\sqrt{[\, n\Sigma x^2 - (\Sigma x)^2 \,][\, n\Sigma y^2 - (\Sigma y)^2 \,]}}$$

So, we have chosen the ultraviolet filter, green filter, red filter, near-infrared filter, and infrared filter in the photometric system, redshift value, and plate ID. These features were selected solely based on their high Pearson correlation coefficient, which is said to have high predictive power in classification algorithms.

4. ALGORITHM IMPLEMENTATION

4.1 ALGORITHMS IMPLEMENTED

Machine learning algorithms advance our understanding of celestial objects, particularly in stellar classification and clustering. These algorithms leverage data-driven techniques to discern patterns, extract features, and uncover relationships within vast datasets originating from astronomical observations. Stellar classification often involves categorizing objects into predefined classes (e.g., stars, galaxies, quasars). Supervised learning algorithms, like Random Forests or Support Vector Machines, can be trained on labeled datasets to learn the relationships between observational features and their represented classes. Clustering algorithms like k-means, hierarchical clustering, or DBSCAN can identify natural groupings within a dataset without predefined labels. In astronomy, this can reveal inherent structures in stellar populations. Machine learning algorithms are powerful tools in stellar classification and clustering, offering the potential to automate and enhance the analysis of vast astronomical datasets. These algorithms contribute to uncovering hidden patterns, identifying distinct classes, and providing deeper insights into celestial objects' diverse nature. Following are some of the algorithms that we built, trained, and predicted on our dataset.

4.1.1 RANDOM FOREST CLASSIFIER

Random Forest is an ensemble of Decision Trees. It builds multiple trees on a random subset of features and training data. The final prediction is an average or vote of the individual trees. Random Forest mitigates overfitting, improves accuracy, and provides insights into feature importance. Random Forest addresses overfitting by aggregating predictions from multiple Decision Trees. This ensemble approach boosts robustness and generalization, making it well-suited for the complexity inherent in astronomical datasets. The Random Forest algorithm measures feature importance, aiding astronomers in prioritizing observational attributes crucial for stellar classification. This insight enhances the interpretability of the classification model. In stellar classification, it can handle diverse observational features and enhance robustness.

4.1.2 XGBOOST CLASSIFIER

XGBoost is a gradient-boosting algorithm that sequentially adds weak learners (typically decision trees) to correct errors made by previous models. It uses gradient descent optimization. XGBoost employs a boosting technique, sequentially improving model performance by focusing on instances where previous models made errors. This precision-oriented approach is beneficial when dealing with complex and nuanced astrophysical patterns. Handling Class Imbalance: XGBoost's ability to handle class imbalance is crucial for stellar classification, where certain classes (stars, galaxies, quasars) may be underrepresented in observational datasets. XGBoost is known for its high performance, efficiency, and ability to handle complex relationships. In stellar classification, it can adapt to intricate patterns and contribute to accurate predictions.

4.1.3 DECISION TREE CLASSIFIER

Decision Trees recursively split the dataset based on feature conditions, forming a tree-like structure. Each leaf node represents a class label. Decision Trees allow astronomers to interpret the rules and conditions leading to specific classifications. This transparency is valuable for understanding the astrophysical implications of features influencing the classification of stars, galaxies, and quasars. Decision Trees are fundamental in understanding the importance of features and capturing complex relationships within astronomical data. However, they can be prone to overfitting.

4.1.4 BAGGING CLASSIFIER

Bagging (Bootstrap Aggregating) combines multiple models trained on different subsets of the training data. It averages their predictions for better generalization. Bagging reduces variance and enhances the stability of models. Bagging Classifier excels in creating an ensemble of models that individually may be sensitive to particular subsets of data. Aggregating predictions achieve stability, advantageous in scenarios with diverse and dynamic astronomical phenomena. Bagging Classifier is well-suited for classifying variable stars, whose brightness changes over

time. Ensemble methods can handle the intricacies of time-series data, contributing to accurate classifications. Classifying stellar objects can contribute to more reliable predictions by aggregating information from diverse subsets.

4.1.5 K-MEANS CLUSTERING

K-Means is an unsupervised clustering algorithm that partitions a dataset into 'k' distinct, non-overlapping subgroups or clusters. The algorithm iteratively assigns data points to the nearest cluster centroid and updates the centroids based on the mean of the assigned points. K-Means can be employed to identify stellar associations or clusters within a galaxy. These associations may represent groups of stars with a common origin or are part of a star-forming region. Features such as red filter, green filter, redshift, infrared filter, and near-infrared filter to help cluster and identify which samples are stars, quasars, and galaxies. As astronomical datasets continue to grow, K-Means offers a scalable clustering solution. Its efficiency makes it suitable for processing vast amounts of observational data to unveil inherent structures. While K-Means provides clear cluster assignments, the interpretability of the clusters may require additional domain knowledge to understand the astrophysical significance of the groupings. In summary, the K-Means algorithm serves as a valuable tool in stellar clustering, enabling astronomers to uncover patterns, identify associations, and gain insights into the organization of celestial objects. Its adaptability, efficiency, and ability to handle multidimensional data make it well-suited for the challenges of large and dynamic astronomical datasets.

4.1.6 NAIVE-BAYES CLASSIFIER

The Naive Bayes classifier is a probabilistic machine learning algorithm based on Bayes' theorem, which calculates the probability of a certain event occurring given the occurrence of another event. It is particularly well-suited for classification tasks, especially in natural language processing and document classification. The "naive" assumption in Naive Bayes is that features are conditionally independent given the class label, simplifying the computation of probabilities. Despite this simplification, Naive Bayes often performs surprisingly well in practice and is computationally efficient. It's widely used for spam filtering, sentiment analysis, and various

other text categorization tasks, where its simplicity, speed, and effectiveness make it an attractive choice for certain applications.

4.1.7 LIGHT GRADIENT BOOSTING MACHINE

LightGBM is an open-source, distributed, high-performance gradient boosting framework developed by Microsoft. It is designed for efficiency, speed, and scalability, making it well-suited for large datasets and computationally intensive tasks. LightGBM is part of the gradient boosting family of machine learning algorithms and is particularly popular for its ability to handle categorical features efficiently. While it's a versatile tool for various machine learning tasks, including classification, regression, and ranking, its application to stellar classification depends on the nature of the data and the specific requirements of the task.

4.2 CORRECT PARAMETER TUNING

Parameter tuning, also known as hyperparameter tuning, is a crucial step in optimizing the performance of machine learning models. Correct parameter tuning involves systematically searching for the best combination of hyperparameters to enhance the model's accuracy and generalization capabilities. Hyperparameters are configuration settings external to the model that impact its learning process. Hyperparameter tuning for decision trees involves optimizing the settings external to the model's training process and can impact performance. Decision trees have several hyperparameters that can be tuned to improve the model's accuracy, prevent overfitting, and enhance its generalization capabilities. Key Hyperparameters:

- Maximum Depth (max_depth): The maximum depth of the tree. A deeper tree can capture more complex relationships but may lead to overfitting.
- Minimum Samples Split (min_samples_split): The minimum number of samples required to split an internal node. Increasing this parameter can prevent the model from creating nodes that only capture noise.
- Minimum Samples Leaf (min_samples_leaf): The minimum number of samples required to be in a leaf node. It enforces a constraint on the size of leaf nodes, preventing small clusters of data points.

- Maximum Features (max_features): The maximum number of features for splitting a node. It can help control the diversity of the trees.

We have given two criteria: gini and entropy. Maximum depth ranges between 5 and 15. The minimum sample split is 2, 5 and 10. The minimum sample leaf is given as 1, 2, and 4. Pre-pruning in decision trees prevents them from overfitting and helps in better visualization of decision trees without it being too complex.

4.3 EFFICIENT CODING AND ALGORITHM EXECUTION

Efficient coding and algorithm execution are crucial aspects of software development, especially in scenarios where performance is a priority.

- Algorithmic Complexity: Analyze algorithms' time and space complexity using Big-O notation. This provides a theoretical foundation for assessing algorithmic efficiency.

- Optimize Loops and Iterations: Reduce the number of nested loops whenever possible. Nested loops contribute to higher time complexity, and alternatives such as memoization or dynamic programming may be considered.

- Memory Management: Be mindful of memory management. Avoid memory leaks by deallocating resources properly, especially in languages without automatic garbage collection.
- Use Built-in Functions and Libraries: Utilize built-in functions and libraries provided by programming languages. These libraries are often optimized for performance and have undergone extensive testing.

5. MODEL EVALUATION AND PERFORMANCE ANALYSIS

When evaluating the performance of a machine learning model for classification tasks, various metrics are used to assess how well the model is performing. When choosing evaluation metrics, consider the classification problem's specific characteristics and your application's goals. For example, metrics like precision, recall, and F1 score may be more informative than accuracy in imbalanced datasets. The choice of metrics should align with the priorities of the problem at hand.

5.1 EVALUATION METRICS AND PERFORMANCE ASSESSMENT

Here are some commonly used classification metrics:

1. ***Accuracy:*** The ratio of correctly predicted instances (both true positives and negatives) to the total number of instances.

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

2. ***Precision:*** The ratio of correctly predicted positive observations to the total predicted positives. Precision emphasizes the model's ability to avoid false positives.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

3. ***Recall*** (Sensitivity or True Positive Rate)***:*** The ratio of correctly predicted positive observations to the total actual positives. Recall emphasizes the model's ability to capture all relevant instances.

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

4. ***F1 Score:*** The harmonic mean of precision and recall. It provides a balanced assessment of a model's performance, especially when an imbalance exists between classes.

$$\text{F1 Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}$$

5. ***Confusion Matrix:*** A table that summarizes the model's predictions, showing the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

6. ***Silhouette Score:*** Silhouette score is a metric for evaluating clusters' quality in a clustering algorithm. It measures how similar an object is to its cluster (cohesion) compared to other clusters (separation). The silhouette score ranges from -1 to 1, where a higher score indicates better-defined clusters

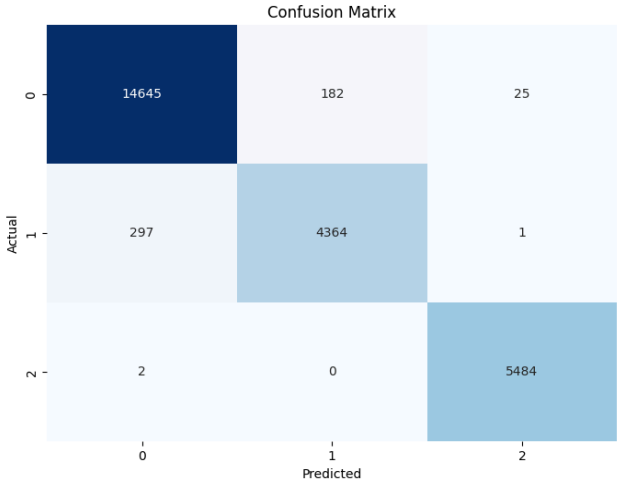
5.2 COMPARATIVE ANALYSIS OF DIFFERENT MODELS

5.2.1 RANDOM FOREST CLASSIFIER

The following images represent the confusion matrix and classification report of Random Forest. Random Forest gave us a good accuracy of 98%. The precision scores were 0.98 on average. The recall scores were 0.95, and the f1 scores were 0.96 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 24,493 class labels and 507 labels that have been incorrectly predicted.



CLASSIFICATION REPORT:

The Classification Report gives us a comprehensive view of our evaluation metric for all three classes.

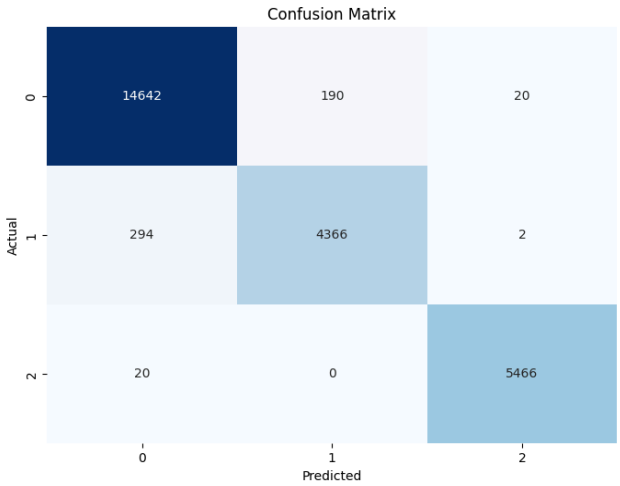
	precision	recall	f1-score	support
0	0.98	0.99	0.98	14852
1	0.96	0.94	0.95	4662
2	1.00	1.00	1.00	5486
accuracy			0.98	25000
macro avg	0.98	0.97	0.98	25000
weighted avg	0.98	0.98	0.98	25000

5.2.2 XGBOOST CLASSIFIER

The Following images represent the confusion matrix and classification report of the XGBoost Classifier. XGBoost Classifier gave us a good accuracy of 98%. The precision scores were 0.97 on average. The recall scores were 0.935, and the f1 scores were 0.96 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 24,474 class labels and 526 labels that have been incorrectly predicted.



CLASSIFICATION REPORT:

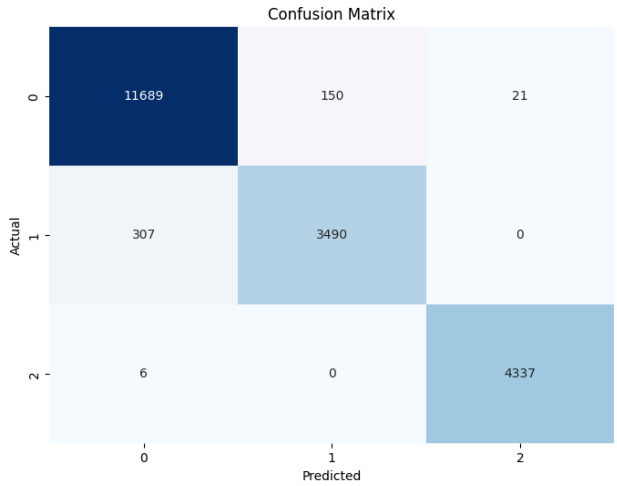
	precision	recall	f1-score	support
0	0.98	0.99	0.98	14852
1	0.96	0.94	0.95	4662
2	1.00	1.00	1.00	5486
accuracy			0.98	25000
macro avg	0.98	0.97	0.98	25000
weighted avg	0.98	0.98	0.98	25000

5.2.3 DECISION TREE CLASSIFIER

The Following images represent the confusion matrix and classification report of the Decision Tree Classifier. Decision Tree also gave us a good accuracy of 98%. The precision scores were 0.98 on average. The recall scores were 0.946, and the f1 scores were 0.97 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 19,516 class labels and 484 labels that have been incorrectly predicted.



CLASSIFICATION REPORT:

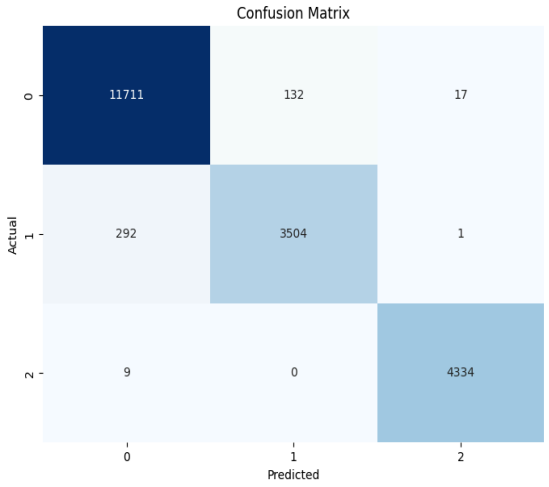
	precision	recall	f1-score	support
0	0.97	0.99	0.98	11860
1	0.96	0.92	0.94	3797
2	1.00	1.00	1.00	4343
accuracy			0.98	20000
macro avg	0.98	0.97	0.97	20000
weighted avg	0.98	0.98	0.98	20000

5.2.4 BAGGING CLASSIFIER

The Following images represent the confusion matrix and classification report of the Bagging Classifier. Bagging also gave us a good accuracy of 98%. The precision scores were 0.97 on average. The recall scores were 0.94, and the f1 scores were 0.96 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 19,549 class labels and 451 labels that have been incorrectly predicted.



CLASSIFICATION REPORT:

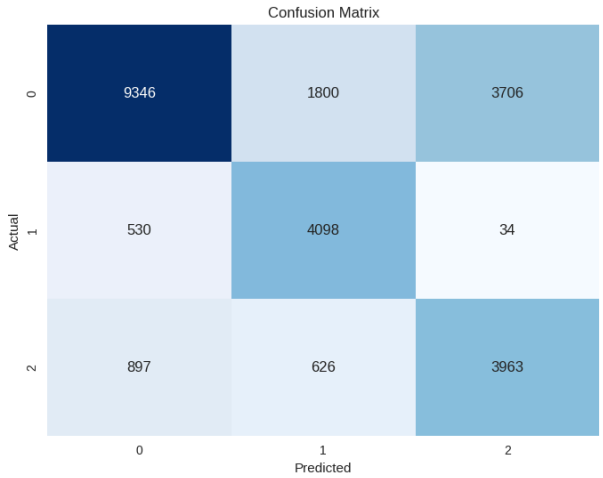
	precision	recall	f1-score	support
0	0.97	0.99	0.98	11860
1	0.96	0.92	0.94	3797
2	1.00	1.00	1.00	4343
accuracy			0.98	20000
macro avg	0.98	0.97	0.97	20000
weighted avg	0.98	0.98	0.98	20000

5.2.5 NAIVE BAYES

The Following images represent the confusion matrix and classification report of the Naive Bayes. Naive Bayes gave us a moderate accuracy of 70%. The precision scores were 0.64 on average. The recall scores were 0.74, and the f1 scores were 0.68 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 17,407 class labels and 7595 labels that have been incorrectly predicted.



CLASSIFICATION REPORT:

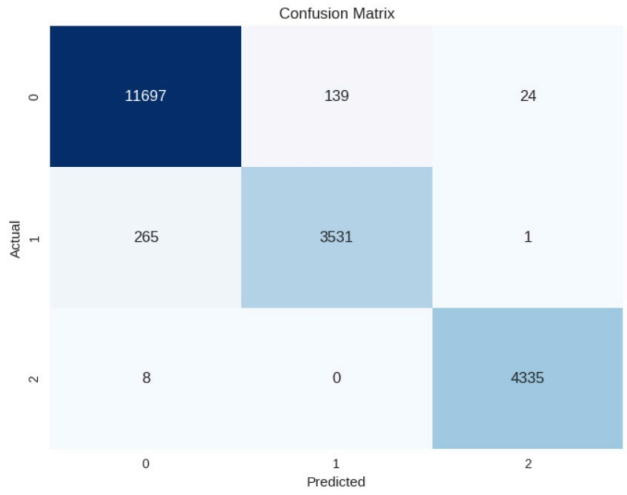
```
Accuracy: 0.6963
Classification Report:
              precision    recall  f1-score   support

           0       0.87      0.63      0.73     14852
           1       0.63      0.88      0.73      4662
           2       0.51      0.72      0.60      5486

    accuracy                           0.70     25000
   macro avg       0.67      0.74      0.69     25000
weighted avg       0.75      0.70      0.70     25000

Confusion Matrix:
[[9346 1800 3706]
 [ 530 4098   34]
 [ 897  626 3963]]
```

5.2.6 LIGHT GRADIENT BOOSTING MACHINE

The Following images represent the confusion matrix and classification report of the Light GBM. Light GBM gave us a moderate accuracy of 97.8%. The precision scores were 0.97 on average. The recall scores were 0.92, and the f1 scores were 0.97 on average. Our model performed well on the given dataset.

CONFUSION MATRIX:

The Confusion Matrix evaluates our model's performance on how many classes it has correctly predicted. It has been able to correctly predict 19,663 class labels and 437 labels that have been incorrectly predicted.



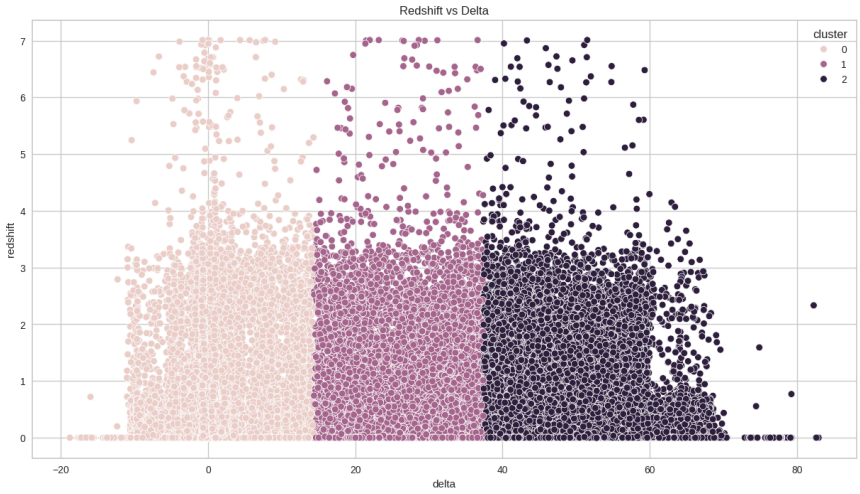
CLASSIFICATION REPORT

```
Accuracy: 0.9781
Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.99      0.98     11860
           1       0.96      0.93      0.95      3797
           2       0.99      1.00      1.00      4343

    accuracy                           0.98     20000
   macro avg       0.98      0.97      0.97     20000
weighted avg       0.98      0.98      0.98     20000
```

5.2.5 K-MEANS CLUSTERING

K-means clustering is a popular unsupervised machine learning algorithm used for partitioning a dataset into distinct groups or clusters. The primary objective of K-means is to minimize the sum of squared distances between data points and their assigned cluster centroids.

CLUSTERS DRAWN:



SILHOUETTE SCORE:

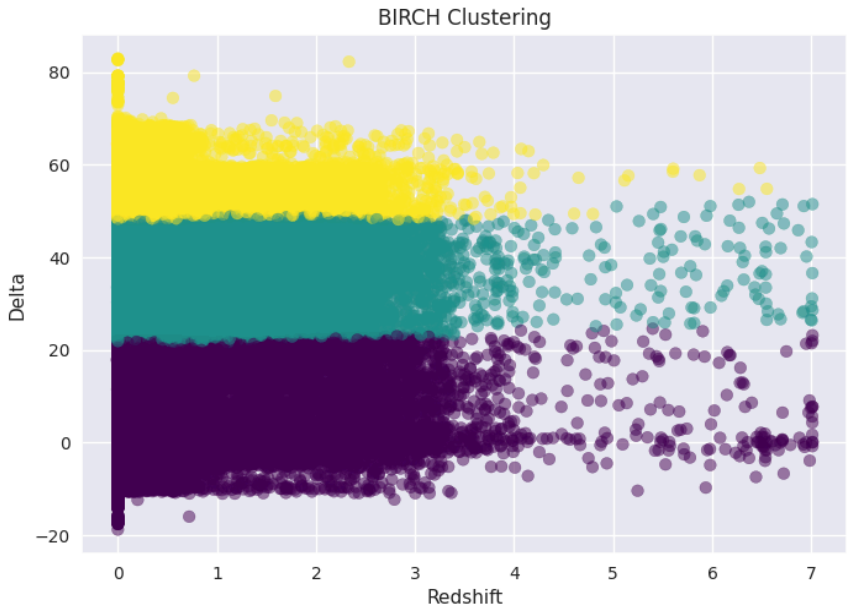
We got a silhouette score of 0.59. A silhouette score of 0.59 is generally considered quite good. The silhouette score is a metric used to evaluate the quality of clusters in a clustering algorithm. It ranges from -1 to 1, where a higher score indicates better-defined clusters.

5.2.6 BIRCH

BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is a clustering algorithm designed for large-scale datasets with a focus on efficiency and scalability. Developed to overcome some limitations of traditional clustering algorithms like K-means, BIRCH uses a hierarchical approach to create a tree-like data structure called the CF-tree (Clustering Feature tree).

CLUSTERS DRAWN

The clusters drawn are well-separated.

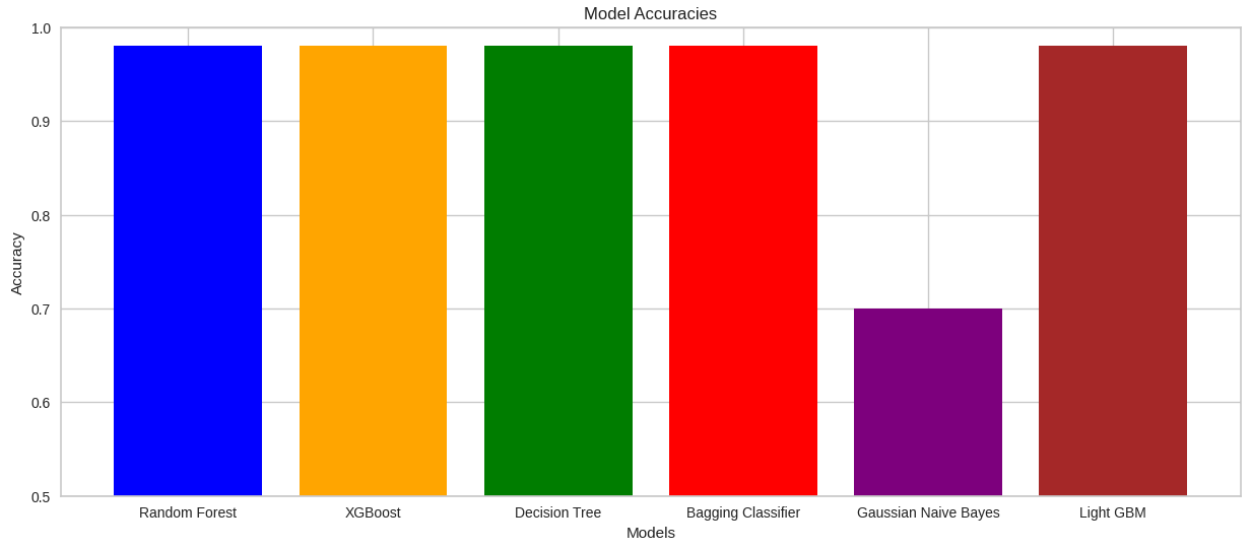


SILHOUETTE SCORE:

We got a silhouette score of 0.5498.A silhouette score of 0.55 is generally considered quite good. The silhouette score is a metric used to evaluate the quality of clusters in a clustering algorithm. It ranges from -1 to 1, where a higher score indicates better-defined clusters.

5.3 INSIGHTFUL INTERPRETATION OF RESULTS

Classification models play a pivotal role in machine learning, encompassing a variety of algorithms such as logistic regression, decision trees, random forests, support vector machines, and neural networks. These models are designed for predicting categorical outcomes based on input features. Logistic regression provides a straightforward interpretation with coefficients representing feature influence, while decision trees offer a hierarchical approach to decision-making. Random forests enhance performance through ensemble learning, and support vector machines focus on maximizing class separation. Each model has its unique strengths and interpretability, making them versatile tools for addressing diverse classification challenges.



Model Comparison: The graph illustrates the comparative accuracy scores of different classification models such as Decision Tree, XGBoost, and Random Forest for the task of stellar classification.

Model Performance: It provides a visual understanding of how well each model performs in accurately classifying the astronomical objects. This can help in identifying the most effective model for this specific classification task.

Decision Making: The accuracy scores can help in decision-making processes regarding which model to choose for the classification of stars, quasars, and galaxies.

Optimization: It provides insight into potential areas for improvement or optimization in the classification models. Models with lower accuracy scores indicate areas for further refinement.

6. REFERENCES

Books:

- "An Introduction to Modern Astrophysics" by Bradley W. Carroll and Dale A. Ostlie
- "Stellar Spectral Classification" by Richard O. Gray and Christopher J. Corbally

Websites and Platforms:

- Sloan Digital Sky Survey (SDSS): https://www.sdss.org/