**Astronomical Object Classification**

**Shlomi Zecharia, Yosef Desse**

*Ariel University, Ariel, Israel*

|  |  |
| --- | --- |
| Shlomi Zecharia,  Department of Computer Science and Mathematics,  Ariel University, Ariel, Israel shlomi.ze29@gmail.com  Yosef Desse,  Department of Computer Science and Mathematics,  Ariel University, Ariel, Israel yosefdassa@gmail.com | **Abstract:** This paper presents a classification model for categorizing astronomical objects into stars, galaxies, and quasars using data from the Sloan Digital Sky Survey DR14. The dataset consists of 10,000 observations with 17 feature columns, including key attributes like redshift, spectral type, and magnitude. The goal is to develop a model that accurately classifies objects based on these features.  **Github:** |
|

**Introduction**

Astronomical object classification plays a crucial role in modern astrophysics, as it allows researchers to categorize celestial objects based on their physical and observational properties. With advancements in space observation technology, large-scale surveys like the Sloan Digital Sky Survey DR14 have provided extensive datasets that include detailed information about stars, galaxies, and quasars. These objects can be classified using various attributes such as redshift, spectral type, and magnitude, which reflect their underlying physical properties. The classification process is essential for understanding the distribution, evolution, and properties of astronomical objects. It enables scientists to make predictions about the nature of objects in the universe, identify new phenomena, and explore relationships between different types of celestial bodies. This paper focuses on the development and application of machine learning techniques to classify astronomical objects based on the SDSS DR14 dataset. The goal is to utilize the available features to build a robust and accurate classification model that can reliably distinguish between stars, galaxies, and quasars.

**Dataset Overview**

The dataset used in this project is derived from the Sloan Digital Sky Survey DR14, which is one of the most comprehensive astronomical surveys. The data includes information on various celestial objects, including stars, galaxies, and quasars, which will be classified based on their features. The dataset consists of 10,000 observations, each described by 17 feature columns and one class column that assigns each object to one of the following categories: star, galaxy, or quasar.

The key attributes used for classification include redshift, spectral type, and magnitude across multiple wavelength bands. These features are crucial for understanding the characteristics of the objects and differentiating between the various categories. The dataset also includes spatial information (right ascension and declination), which provides insights into the positions of the objects in the sky.

**Key Features:**

* **objid (Object Identifier):** A unique identifier for each observation

Usage: Primarily used to distinguish between different observations.

* **ra (Right Ascension**): The angular distance along the celestial equator from the Sun at the March equinox to the hour circle of the point above the Earth.

Usage: Right Ascension, along with Declination, helps pinpoint the position of an object in the sky. In this context, it can provide geographical location data in the celestial coordinate system.

* **dec (Declination):** The angular distance above or below the celestial equator, indicating the location of an object on the celestial sphere.

Usage: Works in tandem with Right Ascension (RA) to pinpoint the object's location in the sky.

**u, g, r, i, z (Magnitude):** These represent the brightness of the object observed in the five filters or bands (u, g, r, i, z) of the telescope. These magnitudes are part of the Thuan-Gunn photometric system.

• u: Represents the ultraviolet magnitude.

• g: Represents the green filter.

• r: Represents the red filter.

• i: Represents the near-infrared filter.

• z: Represents the far-infrared filter.

Usage: These magnitudes are essential for determining the color and temperature of the astronomical objects. They help classify objects like stars, galaxies, and quasars based on their brightness across different wavelengths. Hotter objects like stars tend to have specific patterns in these magnitudes, which can aid in classification.

* **redshift:** The shift in the wavelength of light emitted by the object, typically towards the red end of the spectrum. This shift occurs because objects moving away from Earth stretch the light’s wavelength.

Usage: Redshift helps in estimating the distance of an astronomical object and understanding its motion relative to Earth. Higher redshift values typically indicate that the object is farther away, and this information is vital when classifying galaxies and quasars, which can have different redshift values compared to stars.

* **run (Run Number):** A rerun number specifies a reprocessing of the same observation.

Usage: This is useful in understanding if there were any changes in the data processing methods or if the observation data has been refined or re-analyzed.

* **camcol (Camera Column):** A camera column number identifying the scan line within the observation run.

Usage: This feature helps identify the specific area within the telescope's field of view during the observation.

* **field (Field Number):** The field number is used to identify a section of the sky observed during the scan.

Usage: Like other technical features (run, rerun, camcol), it helps pinpoint the location of the observation. While this can be useful in some spatial or time-series analysis.

* **specobjid (Spectral Object Identifier):** A unique identifier for each object in the spectral data view.

Usage: This serves primarily as a reference for matching spectral data to objects. It’s helpful for combining multiple datasets or for linking objects across different views.

* **plate:** The plate number, identifying the specific spectroscopic plate used to capture the object’s light.

Usage: It identifies which plate (in the SDSS’s catalog) the object belongs to.

* **mjd (Modified Julian Date):** The date and time the observation was taken, expressed in Modified Julian Date.

Usage: Knowing the observation date can help in temporal analysis (e.g., to check for trends over time or variations in the data).

* **fiberid:** The identifier for the fiber used to collect light from the object.

Usage: Each observation is associated with a specific optical fiber, which helps track the light's path through the telescope’s optical system.

**Tools and Techniques**

**1. SVM (Support Vector Machine)** :

Description: A supervised machine learning model used for classification tasks. SVM aims to find the optimal hyperplane that maximizes the margin between classes (in this case, star, galaxy, and quasar).

Application: SVM will be used to identify the best separating boundary between the three classes based on the provided features (e.g., redshift, magnitude, spectral type).

**2. KNN (K-Nearest Neighbors):**

Description: A non-parametric classification method where the class of a sample is determined by the majority class among its nearest neighbors in the feature space.

Application: KNN will classify objects by analyzing the distance between their feature values (e.g., redshift, magnitude) and their nearest neighbors in the dataset.

**3. Logistic Regression :**

Description: A classification model that predicts the probability of an input belonging to a specific class. It can be extended to multi-class problems using Softmax.

Application: Softmax is used to handle multi-class classification (star, galaxy, or quasar). Cross-entropy loss is applied during training to minimize prediction errors.

**4. Decision Tree Classifier:**

Description: A model that splits the data into subsets based on the most significant features, forming a tree-like structure that makes predictions by following the decision rules.

Application: The decision tree will be used to classify astronomical objects by evaluating the most important features (like spectral type or redshift) and splitting the data accordingly.

**5. Random Forest Classifier:**

Description: A model that combines multiple decision trees, where each tree is trained on a random subset of the data and features. The final prediction is made by aggregating the results of all trees (majority vote for classification).

Application: The random forest will classify astronomical objects by leveraging the collective decision of multiple trees. It provides improved accuracy and robustness compared to a single decision tree and highlights the most important features (e.g., redshift, spectral type) for classification.

**Research Questions**

• Can we classify astronomical objects (stars, galaxies, and quasars) based on their features, such as redshift, magnitude, and spectral type?

• Which combination of features provides the most accurate classification for distinguishing stars, galaxies, and quasars?

• Can we achieve a high classification accuracy using machine learning models on this dataset?

• Can PCA (Principal Component Analysis) improve classification accuracy?

• Which of the five machine learning algorithms Logistic Regression, Decision Tree, SVM, KNN and Random Forest performs best for classifying astronomical objects from the SDSS dataset?

**Exploratory Data Analysis (EDA)**

*Data Preprocessing and Cleaning*

First, Let's Examine the First Five Rows of the Data:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer code

Description automatically generatedChecking for Missing Values and Numerical Data Types:

There are no missing data values (10,000 non-null entries) and all features are numerical, except for the class (label), we'll deal with that later.

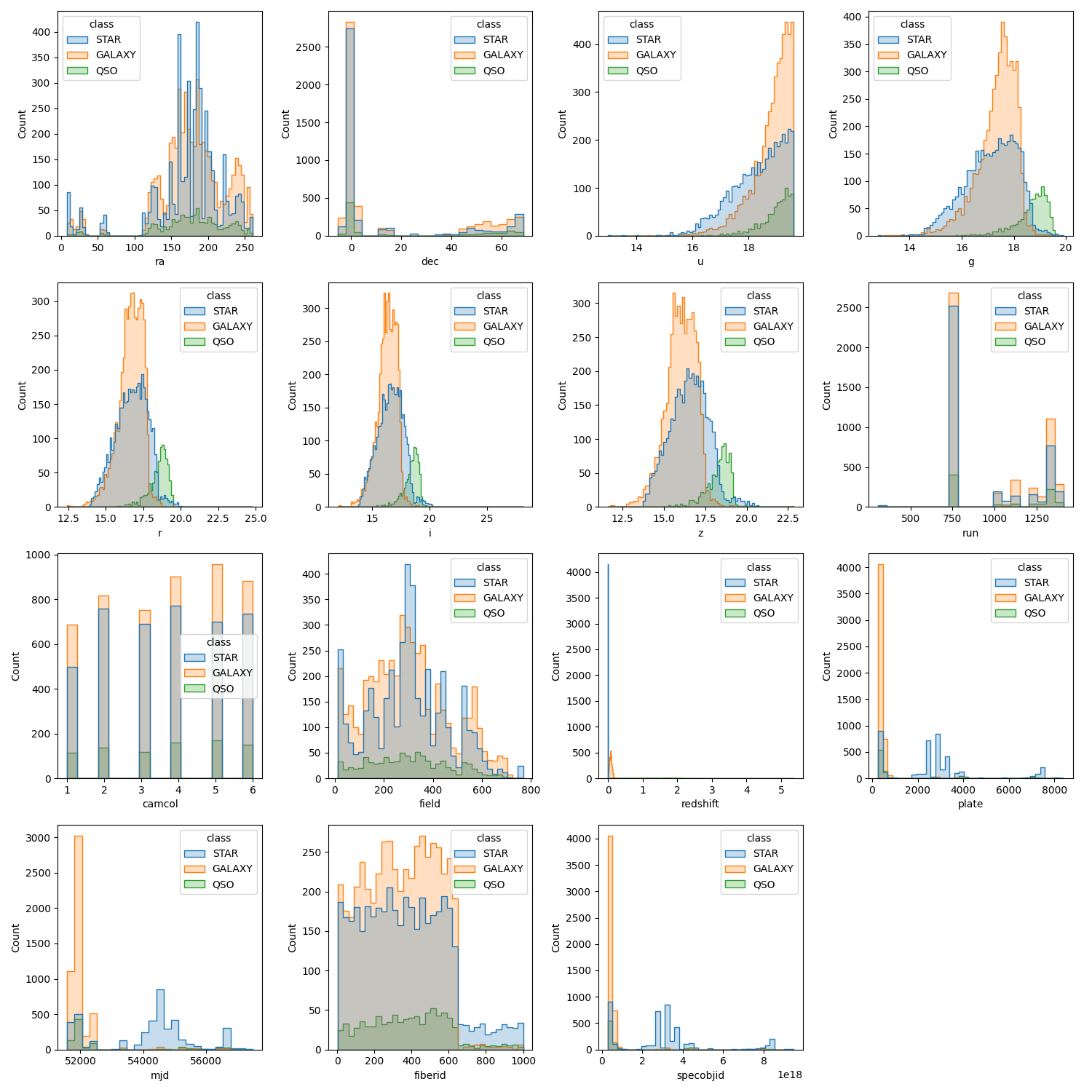
A screenshot of a computer

Description automatically generatedUnique values for each feature:

It can be observed that *rerun* and *objid* contain only a single unique value, making them non-contributory to the dataset. Therefore, we **remove** them from the dataset.

Additionally, examining *camcol* and *run*, it appears that they are likely categorical rather than numerical variables. This is suggested by the fact that *camcol* has only 6 unique values and *run* has just 23 unique values out of 10,000 samples. We will further analyze this by inspecting their histograms later to determine whether they exhibit characteristics of categorical or numerical data.

*Univariate & Bivariate Analysis*



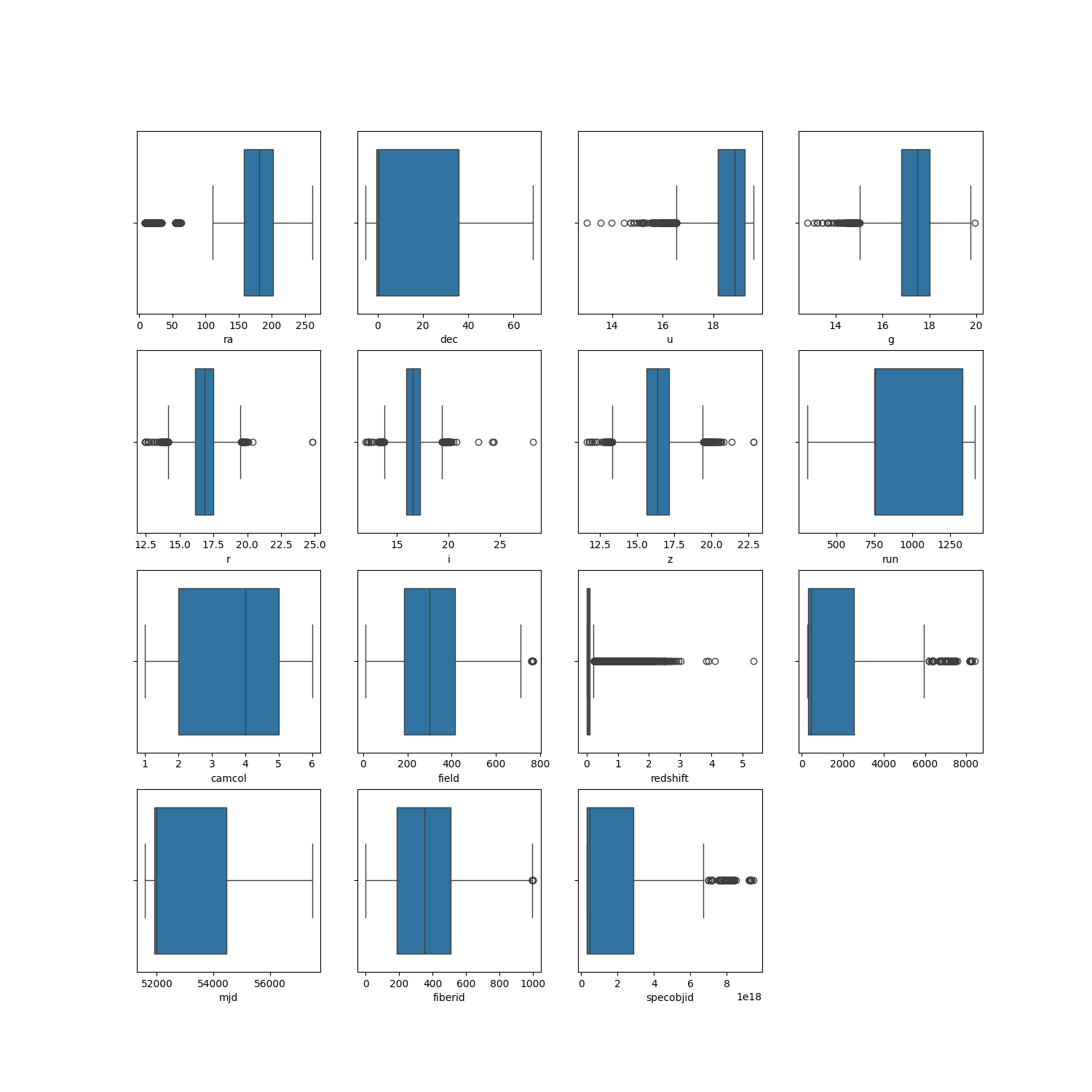
Fig. 1.Histograms - plots the distribution of a numeric variable's values as a series of bars

Fig. 2. Box plots - The data shows varying levels of dispersion across features.

A screenshot of a computer

Description automatically generated

Fig. 3.Heatmap - represents a correlation matrix between various variables. Correlation measures the strength and direction of the linear relationship between two variables.

A collage of graphs showing different colored dots

Description automatically generated with medium confidence

Fig. 4. scatter Plots - Used to compare pairs of features (u, g, r, i, z).

*Conclusions of the analysis*

Fig .1 : Most numerical variables (u, g, r, i, z) exhibit a normal-like distribution, indicating that their values are centered around a mean. This is particularly useful when preprocessing, as features with normal distributions can benefit more from standardization or scaling. The other variables, like plate, fiberid, and mjd, show relatively uniform distributions, suggesting that these features might not have as much predictive power on their own.

Additionally, the histograms reveal that *camcol* and *run* are categorical variables. This is evident from the clear gaps between the bars in their histograms, indicating a lack of continuity and confirming that they represent distinct categories rather than continuous numerical values. A similar pattern can be observed for *plate, mjd,* and *specobjid*.

For instance, specobjid contains close to 6,500 unique values. However, its range spans from approximately 2.995780e+17 to 9.468830e+18, which is an extremely large interval given the limited number of unique values. Despite this wide range, its histogram exhibits characteristics of a categorical variable rather than a continuous one. This suggests that specobjid may not be useful for learning, as categorical features with a high number of unique values (65% of the dataset in this case) provide little to no meaningful information for modeling. In fact, they can negatively impact learning by introducing unnecessary patterns, as each category would have an average of only 1.54 samples, making it ineffective for generalization. Therefore, specobjid is likely to be removed from the dataset.

Fig .2 : Features dispersion:

u, g, r, i, z: These exhibit relatively low dispersion. Their small IQR (Interquartile Range) indicates clustered data and low variability.

m, dec, run, field, plate, mjd, fiberid: These show higher dispersion. Their wider boxes reflect a larger range of values and greater variability.

redshift, specobjid: These contain numerous outliers. Points outside the box plot whiskers denote extreme values, which may require further scrutiny.

Fig .3: Correlations between features

Strong Positive Correlation:

Range: 0.7 to 1

High values of one variable are strongly associated with high values of the other. When one increases, the other tends to increase as well.

Moderate Positive Correlation:

Range: 0.3 to 0.7

There is a noticeable relationship between the variables, but it is not particularly strong. An increase in one variable is somewhat associated with an increase in the other.

Weak Positive Correlation:

Range: 0 to 0.3

There is a slight positive relationship, but it is weak and not consistent.

No Correlation:

Range: -0.3 to 0.3

There is no significant linear relationship between the variables. Changes in one variable do not consistently affect the other.

Weak Negative Correlation:

Range: -0.3 to 0

There is a slight inverse relationship, but it is weak.

Moderate Negative Correlation:

Range: -0.7 to -0.3

An increase in one variable is somewhat associated with a decrease in the other.

Strong Negative Correlation:

Range: -1 to -0.7

High values of one variable are strongly associated with low values of the other, and vice versa.

Strong positive correlations:

* There is a very strong correlation between (u, g), (g, r), (g, i), (g, z), (i, r), (i, z), and (r, z), indicating a strong linear relationship between these wavelengths.
* A strong correlation exists between plate and mjd, as well as between plate and specobjid, which is logical due to their technical interconnection. Our analysis indicates that mjd*,* plate*,* and specobjid are categorical rather than numerical variables. When categorical features exhibit a high correlation, such as the perfect correlation (1.0) between plate*,* and specobjid, it suggests that one is a deterministic function of the other, meaning that knowing the value of one fully determines the value of the other.

This redundancy indicates that both features encode the same underlying information, providing little additional predictive power. Keeping both may introduce unnecessary complexity and multicollinearity, which can negatively impact the model. Therefore, one of these features should likely be removed.

Strong Negative correlations:

* There are no strongly negative correlations between the features.

Weak or no correlations:

* Most other correlations are weak or non-existent, meaning there is no significant linear relationship between these variables.

The strong positive correlations suggest that these variables might carry redundant information. Techniques like PCA or feature selection could help reduce dimensionality while retaining the most relevant information.

Fig .4 : Analysis of Scatter Plots for u, g, r, i, z features:

As observed in the heatmap (Fig. 3), a strong positive correlation exists among the u, g, r, i, and z features. To further investigate these relationships, we generated scatter plots for each pair of these features.

The scatter plots reveal varying degrees of correlation strength and differences in data spread. For instance, the i vs. r plot demonstrates a high degree of correlation, as evidenced by tightly clustered points forming a clear linear pattern. This suggests that i and r values change together consistently, with relatively low dispersion around the trend line.

Conversely, the u vs. z plot exhibits a weaker correlation. The points are more widely dispersed, indicating that while there might be a general positive trend, the relationship is less pronounced and more variable.

By examining the scatter plots, we gain a deeper understanding of the relationships between the u, g, r, i, and z features. The strong correlations suggest that some of these features may carry redundant information, potentially leading to high dimensionality without adding significant discriminatory power. Applying dimensionality reduction techniques like PCA could help simplify the model while retaining most of the relevant information.

*Prepare Data & Feature Engineering*

**Normalization**: Data normalization is an essential preprocessing step in machine learning that aims to standardize the range of independent variables or features of the dataset. In many cases, raw data may come in varying scales, and features with larger numeric ranges could dominate the learning process, leading to biased results or slower convergence during model training.

We will use the Standardization (Standard Scaling) method, this technique rescales the data to have a mean of 0 and a standard deviation of 1 by subtracting the mean and dividing by the standard deviation of each feature.

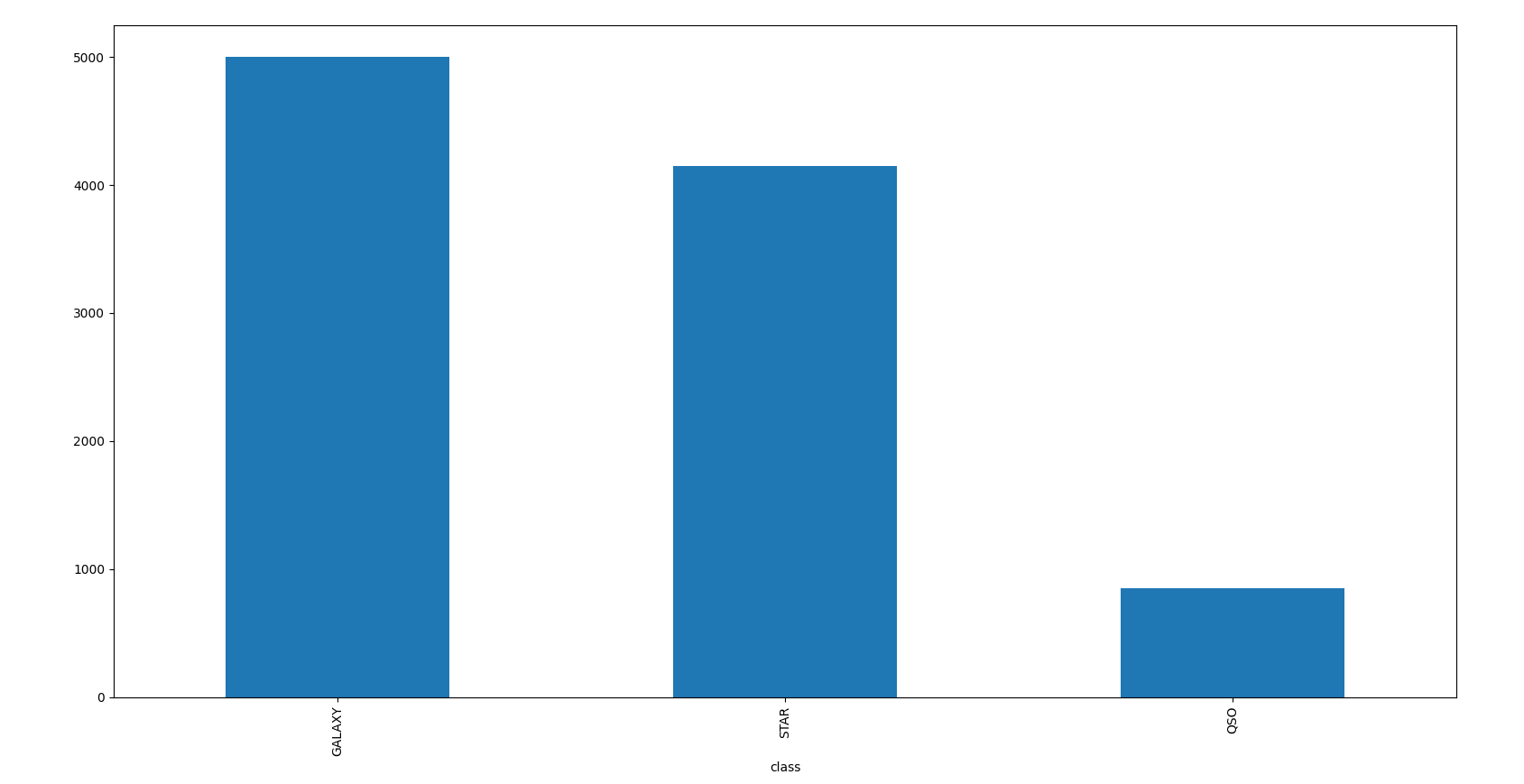
Let's look at the number of samples from each class:

Fig. 5. Bar Plot of Class Distribution - This graph visually represents the number of samples in each class (**Galaxy**, **Star**, and **QSO**) using a bar chart.

Fig. 5: The dataset demonstrates a significant imbalance between the classes. Specifically, we have nearly 5,000 samples for the **Galaxy** class, slightly over 4,000 samples for the **Star** class, and fewer than 1,000 samples for the **QSO** class. We will first attempt to classify the data using the original distribution as is. If the classification results for the **QSO** class are unsatisfactory, we will consider applying one of the following techniques to address the class imbalance: Oversampling, Data Augmentation or Weighted Loss Function.

Split data to train, validation, test:

We will divide the data into training, validation and test so that the data is balanced.

We will convert the class values ​​to numeric values:

STAR:0 ,GALAXY:1 ,QSO:2

**Model Training**

For each model we used **grid search**:

Grid search is used for hyperparameter tuning by exhaustively searching through a specified parameter grid to find the optimal combination that maximizes model performance.

We split the data into **60% training**, **20% validation** and **20% testing**, and during training we use cross validation into **five folds**.

*SVM (Support Vector Machine)*

Best model from grid search :

Model Kernel – Linear:

The linear kernel is used when the data is approximately linearly separable. It finds a straight hyperplane that best separates the classes, making it computationally efficient and interpretable.

Regularization Parameter – C = 10:

The regularization parameter **C** controls the trade-off between maximizing the margin and minimizing classification errors. A higher value (C=10) prioritizes correct classification over a wider margin, making the model more sensitive to training data but potentially increasing the risk of overfitting.

Result :

(image here)

SVM Model Performance Analysis:

The SVM model delivers outstanding classification results with high accuracy across all datasets.

Key Observations:

* STAR class achieves perfect recall, meaning all STAR instances were correctly identified across all datasets.
* GALAXY and QSO maintain strong classification performance, with QSO achieving a 96% precision in the test set, indicating well-separated feature distributions.
* Misclassifications are minimal, primarily between GALAXY and QSO, as shown in the confusion matrices.
* Consistent accuracy across training, validation, and test sets suggests minimal overfitting, indicating that the model generalizes well to unseen data.
* Validation performance aligns closely with training and test results, further reinforcing the model’s robustness.

Overall, the SVM model effectively distinguishes between classes, demonstrating clear decision boundaries in the feature space.

*KNN (K-Nearest Neighbors)*

Best model from grid search :

Number of nearest neighbors = 3:

This parameter specifies how many of the closest data points (neighbors) are considered when making predictions. A lower value (e.g. 3) means that the classifier will be more sensitive to local patterns in the data, while higher values might smooth out decisions but could lose local detail.

P\_ Distance = 1:

The p parameter defines the distance metric used to calculate the proximity between points. When p = 1, the model uses the **Manhattan distance** (also known as L1 distance), which sums the absolute differences between the coordinates of the points.

Result :

(image here)

KNN Model Performance Analysis:

The KNN model shows solid performance but exhibits signs of overfitting.

Key Observations:

* STAR class recall is lower (87% in validation, 88% in test), suggesting that some STAR instances are being misclassified.
* QSO class exhibits the most variance, with recall dropping from 90% (train) to 82% (validation & test), indicating instability in classification.
* Confusion matrices show misclassifications primarily between STAR and GALAXY.
* A notable drop in accuracy from training to validation/test indicates overfitting, likely caused by an overly specific model fitting the training set too closely.

Potential Overfitting Explanation:

* The KNN model was optimized using Grid Search, which selected K=3 as the best parameter.
* With only 3 neighbors, the decision boundary is highly sensitive to local variations and noise in the training data.
* This can lead to overfitting, as the model may classify based on very few data points, making it less generalizable to unseen samples.
* A higher K value (e.g., 5 or 7) could help reduce sensitivity to noise and improve generalization.

Overall, while KNN performs well, the choice of K=3 suggests excessive reliance on individual data points, leading to reduced stability in validation and test performance.

Let's change to K=7 to reduce sensitivity to noise:

*Logistic Regression*

Best model from grid search :

Regularization Type – Lasso:

Lasso (L1 Regularization) adds a penalty based on the absolute values of the coefficients. It can set some coefficients to zero, effectively removing those features from the model. This helps with feature selection and prevents overfitting by making the model simpler.

Regularization Parameter – C = 100:

C in Lasso regularization controls the strength of the penalty. A larger C means less regularization (the model allows larger coefficients), while a smaller C increases the penalty, encouraging more coefficients to shrunk to zero, leading to a simpler model with fewer features.

(C = 100 means coefficient = 1/100)

Result :

(image here)

Logistic Regression Model Performance Analysis:

The Logistic Regression model also demonstrates strong classification ability.

Key Observations:

* All classes achieve high precision, recall, and F1-scores, suggesting strong decision boundaries.
* QSO recall is slightly lower (92% in train, 91% in validation, 94% in test), indicating some difficulty distinguishing QSO samples from other classes.
* The confusion matrices show misclassifications mainly between GALAXY and QSO.
* Validation results closely match training and test results, suggesting excellent generalization.

Overall, the Logistic Regression model provides a strong balance between performance and generalization.

*Decision Tree*

Best model from grid search :

Criterion – Entropy:

In a Decision Tree algorithm, the criterion defines the method for splitting nodes. Using entropy as the criterion means the tree will select splits based on information gain, aiming to reduce disorder in the dataset. A split that maximizes entropy reduction leads to more pure groups, improving the accuracy of the model by ensuring that the data within each branch is as homogeneous as possible.

Max Depth – 6:

The max depth parameter controls how deep the tree can grow. A depth of 6 prevents the tree from becoming too complex and overfitting the data. Limiting the depth ensures the tree generalizes well by capturing essential patterns without fitting noise. This makes the model simpler and more interpretable while avoiding overfitting to the training data.

Min Samples Leaf – 7:

The min samples leaf parameter ensures that each leaf node in the tree contains at least 7 samples. This requirement helps the model avoid creating overly specific branches that may fit only a small portion of the training data, thus reducing the risk of overfitting. It forces the tree to make splits that are meaningful across a larger portion of the data, leading to better generalization.

Fig. 6. The importance features in SVM classification

Fig. 7. The importance features in KNN classification

Fig. 8. The importance features in Logistic Regression