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 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 (SDSS), 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.

- **spType (Spectral Type):** The classification of the object based on its temperature and absorption lines in its spectrum.
  Usage: Spectral type is crucial for classifying stars and other objects in space based on their spectral properties. It helps determine if the object is a star, galaxy, or quasar, and provides insight into its physical characteristics, such as temperature and size.

- **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:

objid	ra	dec	u	g	r	i
1.237650e+18	183.531326	0.089693	19.47406	17.04240	15.94699	15.50342
1.237650e+18	183.598370	0.135285	18.66280	17.21449	16.67637	16.48922
1.237650e+18	183.680207	0.126185	19.38298	18.19169	17.47428	17.08732
1.237650e+18	183.870529	0.049911	17.76536	16.60272	16.16116	15.98233
1.237650e+18	183.883288	0.102557	17.55025	16.26342	16.43869	16.55492

z	run	rerun	camcol	field	specobjid	class	redshift	plate	mjd	fiberid
15.22531	752	301	4	267	3.722360e+18	STAR	-0.000009	3306	54922	491
16.39150	752	301	4	267	3.638140e+17	STAR	-0.000055	323	51615	541
16.80125	752	301	4	268	3.232740e+17	GALAXY	0.123111	287	52023	513
15.90438	752	301	4	269	3.722370e+18	STAR	-0.000111	3306	54922	510
16.61326	752	301	4	269	3.722370e+18	STAR	0.000590	3306	54922	512

Checking for Missing Values and Numerical Data Types:

```
objid      10000 non-null  float64
ra         10000 non-null  float64
dec        10000 non-null  float64
u          10000 non-null  float64
g          10000 non-null  float64
r          10000 non-null  float64
i          10000 non-null  float64
z          10000 non-null  float64
run        10000 non-null  int64
rerun      10000 non-null  int64
camcol     10000 non-null  int64
field      10000 non-null  int64
specobjid  10000 non-null  float64
class      10000 non-null  object
redshift   10000 non-null  float64
plate      10000 non-null  int64
mjd        10000 non-null  int64
fiberid    10000 non-null  int64
```

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.

Unique values for each feature:

```
ra: 10000 unique values
dec: 10000 unique values
u: 9730 unique values
g: 9817 unique values
r: 9852 unique values
i: 9890 unique values
z: 9896 unique values
run: 23 unique values
rerun: 1 unique values
camcol: 6 unique values
field: 703 unique values
redshift: 9637 unique values
plate: 487 unique values
mjd: 355 unique values
fiberid: 892 unique values
specobjid: 6349 unique values
objid: 1 unique values
```

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.

Univariate & Bivariate Analysis

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

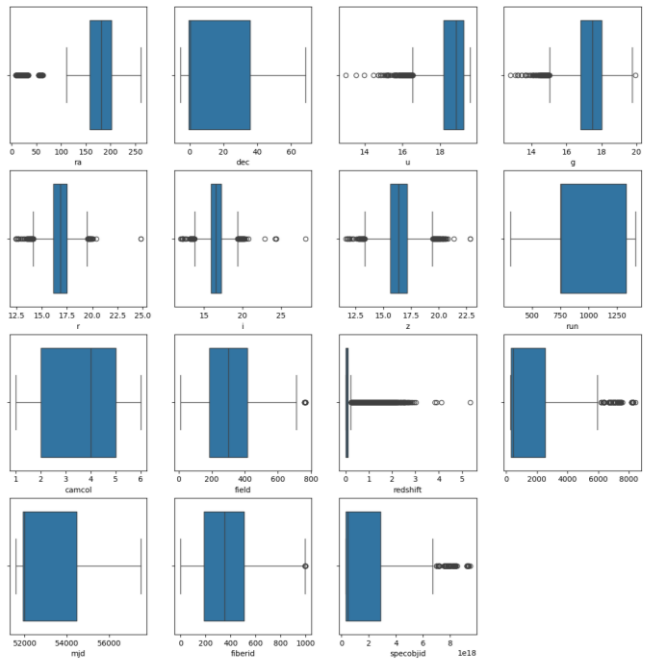

Fig. 2. Box plots - The data shows varying levels of dispersion across features

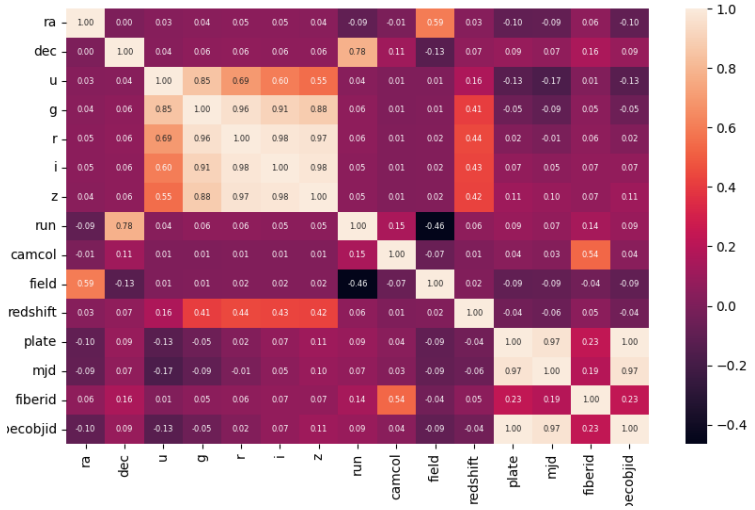

Fig. 3.Heatmap - represents a correlation matrix between various variables. Correlation measures the strength and direction of the linear relationship between two variables. Warm colors (red, orange): Strong positive correlation, Cool colors (blue): Strong negative correlation, Neutral colors (white, yellow): Weak or no correlation.

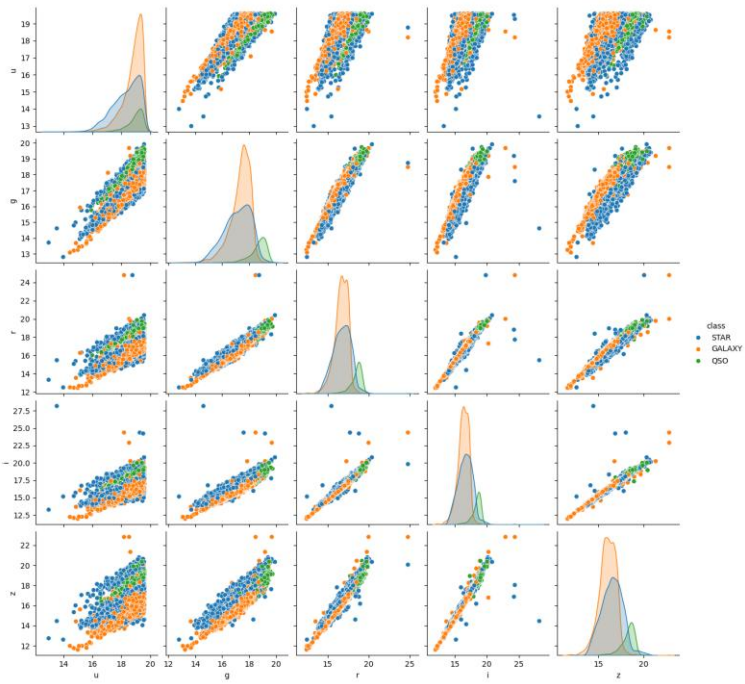

Fig. 4. Scatter Plots - Used to compare pairs of features (u, g, r, i, z). The analysis shows strong linear correlations between the features u, g, r, i, and z, with significant overlap between the classes STAR, GALAXY, and QSO.

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.

Fig .2 : With some (like u, g, r, i, z) having relatively low dispersion, while others (such as m, dec, run, field, plate, mjd, fiberid) exhibit higher dispersion. Certain features, like redshift and specobjid, contain many outliers, which should be carefully examined and handled

Fig .3 : Correlations between features:
Strong positive correlations:

- There is a very strong correlation between u, g, r, i, 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.
- A moderate positive correlation exists between run and field.

Negative correlations:

- A weak negative correlation exists between dec and ra.

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 among u, g, r, i, and z suggest that these variables might carry redundant information. This is a common scenario in astronomy, where these bands represent measurements of the same object under different wavelengths. Techniques like PCA or feature selection could reduce dimensionality while retaining the most relevant information.

Fig .4 : The analysis shows strong linear correlations between the features u, g, r, i, and z, with significant overlap between the classes STAR, GALAXY, and QSO.

- **u vs. g**: A strong positive linear correlation is observed with considerable overlap between the classes. STARs tend to have lower values for both u and g.
- **u vs. r**: Similar to u vs. g, there is a positive linear correlation and significant class overlap.
- **u vs. i**: The correlation remains linear and positive, but the class overlap appears slightly reduced.
- **u vs. z**: The correlation is still positive and linear, with even less overlap between the classes.
- **g vs. r**: A strong positive linear correlation with noticeable class overlap.
- **g vs. i**: Similar to g vs. r, with slightly less overlap between the classes.
- **g vs. z**: Positive linear correlation with further reduced overlap.
- **r vs. i**: A positive linear relationship with significant class overlap.
- **r vs. z**: Similar to r vs. i, with reduced overlap.
- **i vs. z**: Positive linear correlation, with the least overlap among all the pairs.

Despite the correlations, the significant overlap between the STAR, GALAXY, and QSO classes suggests that these features alone may not provide enough information to distinguish between these classes clearly. Dimensionality reduction methods like PCA could simplify the model without significant performance loss.

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:

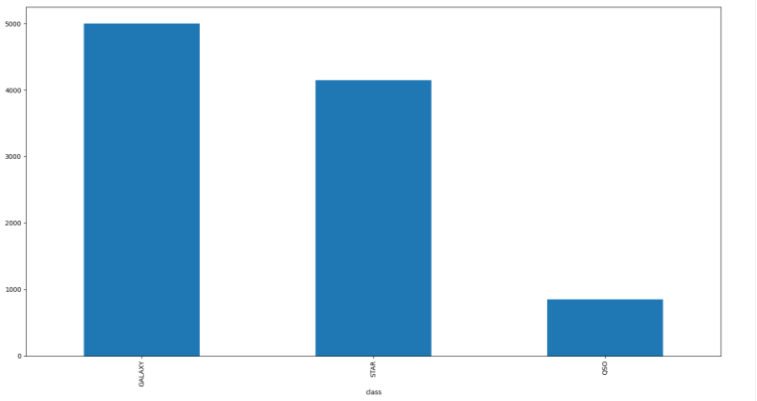

Fig. 5. Bar Plot of Class Distribution - This graph visually represents the number of samples in each class (**Galaxy**, **Star**, and **QSO**) using a bar chart.

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 and test:
We will divide the data into training 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 **80% training** and **20% testing**, and during training we use cross validation and split the train 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 :

Train Accuracy: 0.988125

Train classification report:

	precision	recall	f1-score	support
STAR	0.98	1.00	0.99	2667
GALAXY	0.99	0.98	0.99	3181
QSO	0.98	0.97	0.97	552
accuracy			0.99	6400
macro avg	0.99	0.98	0.98	6400
weighted avg	0.99	0.99	0.99	6400

```
Confusion Matrix
 [[2667    0    0]
 [  47 3123   11]
 [   1   17  534]]
```

Test Accuracy: 0.9855

Test classification report:

	precision	recall	f1-score	support
STAR	0.98	1.00	0.99	830
GALAXY	0.99	0.98	0.99	1000
QSO	0.96	0.97	0.97	170
accuracy			0.99	2000
macro avg	0.98	0.98	0.98	2000
weighted avg	0.99	0.99	0.99	2000

```
Confusion Matrix
 [[830   0   0]
 [ 18 976   6]
 [  0   5 165]]
```

SVM Model Performance Analysis:

The SVM model delivers excellent results with high accuracy (98.8% train, 98.55% test) and strong precision, recall, and F1-scores across all classes.

- STAR class achieves perfect recall, while GALAXY and QSO maintain high classification performance.
- Despite having significantly fewer samples, QSO achieves a 96% precision, suggesting that its feature distribution is well-separated from the other classes.
- Minimal misclassifications, mainly between GALAXY and QSO, as seen in the confusion matrix.
- Consistent training and test accuracy indicate minimal overfitting.

This suggests that QSO instances are inherently distinguishable based on the given features, potentially indicating clear decision boundaries in the feature space. Overall, the model effectively classifies the classes with strong generalization.

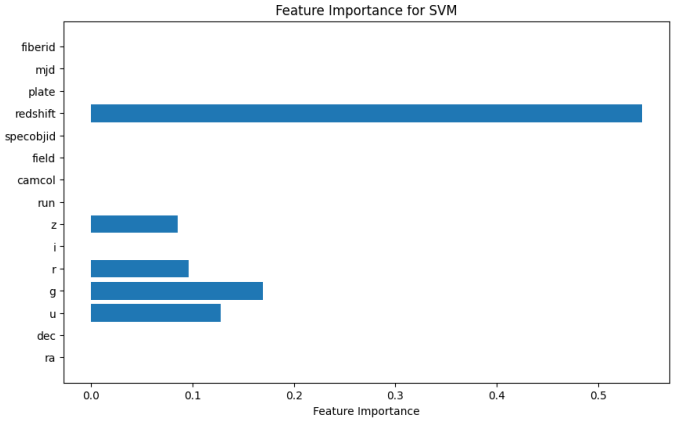

Fig. 6. The importance features in SVM classification

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 :

Train Accuracy: 0.95359375
Train classification report :

	precision	recall	f1-score	support
STAR	0.96	0.94	0.95	2667
GALAXY	0.94	0.97	0.96	3181
QSO	0.98	0.90	0.94	552
accuracy			0.95	6400
macro avg	0.96	0.94	0.95	6400
weighted avg	0.95	0.95	0.95	6400

Confusion Matrix

```
[[2518  147    2]
 [  85 3090    6]
 [  23   34  495]]
```

Test Accuracy: 0.91
Test classification report:

	precision	recall	f1-score	support
STAR	0.93	0.88	0.90	830
GALAXY	0.89	0.95	0.92	1000
QSO	0.95	0.82	0.88	170
accuracy			0.91	2000
macro avg	0.92	0.88	0.90	2000
weighted avg	0.91	0.91	0.91	2000

Confusion Matrix

```
[[729  97   4]
 [ 45 951   4]
 [ 14  16 140]]
```

KNN Model Performance Analysis:

The KNN model shows strong performance with high accuracy (95.35% train, 91% test), but there's a noticeable drop in test accuracy, indicating potential overfitting.

- STAR class: High precision (96%) and recall (94%) in training, but recall drops to 88% in test, suggesting some misclassifications.
- GALAXY class: Strong precision (94%) in training, but precision drops to 89% in test, showing more misclassifications between STAR and GALAXY.
- QSO class: High precision (98%) in training, but recall decreases to 82% in test, indicating difficulty identifying QSO instances in the test set.

Overall, the model performs well but might benefit from further tuning to improve generalization and reduce overfitting. We will try different techniques to improve the model later.

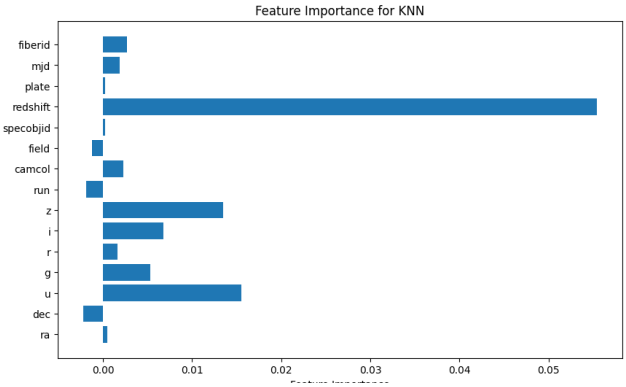

Fig. 7. The importance features in KNN classification

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 be shrunk to zero, leading to a simpler model with fewer features.

Result :

Train Accuracy: 0.984375

Train classification report:

	precision	recall	f1-score	support
STAR	0.99	1.00	0.99	2667
GALAXY	0.98	0.99	0.98	3181
QSO	0.97	0.92	0.95	552
accuracy			0.98	6400
macro avg	0.98	0.97	0.97	6400
weighted avg	0.98	0.98	0.98	6400

```
Confusion Matrix
 [[2655   12    0]
 [  30 3136   15]
 [   1   42  509]]
```

Test Accuracy: 0.984

Test classification report:

	precision	recall	f1-score	support
STAR	0.98	1.00	0.99	830
GALAXY	0.99	0.98	0.98	1000
QSO	0.96	0.94	0.95	170
accuracy			0.98	2000
macro avg	0.98	0.97	0.98	2000
weighted avg	0.98	0.98	0.98	2000

```
Confusion Matrix
 [[827   3   0]
 [ 13 981   6]
 [  0  10 160]]
```

Logistic Regression Model Performance Analysis:

The Logistic Regression model shows excellent performance with very high accuracy (98.4% train, 98.4% test) and strong precision, recall, and F1-scores across all classes.

- STAR class: Achieves perfect recall (100%) in both train and test sets, showing almost no misclassification for this class.
- GALAXY class: Strong precision (98% train, 99% test) and recall (99% train, 98% test), indicating consistent and reliable classification.
- QSO class: Precision of 97% (train) and 96% (test), with recall of 92% (train) and 94% (test), demonstrating good performance despite fewer samples.

The confusion matrices show minimal misclassifications, mainly between GALAXY and QSO. The model performs particularly well in the test set, with slight misclassifications between STAR and GALAXY. Overall, the model generalizes very well with consistent performance on both training and test data, showing minimal overfitting. This indicates that the model is well-regularized and the classes are distinguishable based on the provided features.

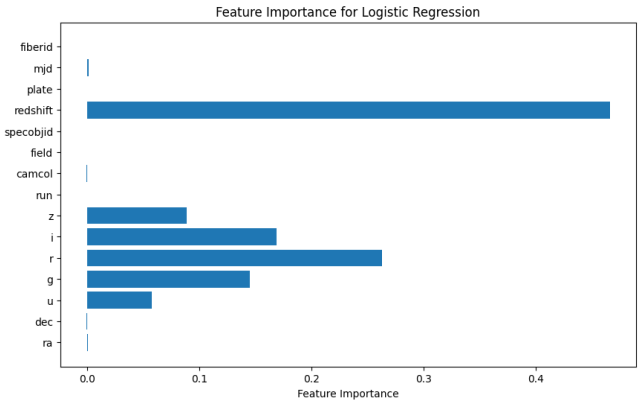

Fig. 8. The importance features in Logistic Regression