Indian Institute of Science Education and Research, Kolkata

Department of Physical Sciences

Project:
PH4202
SPACE ASTRONOMY

Stellar Classification

Classifying the Celestial Objects

Submitted by :
Anurit Dey, 19MS045
Abhishek Sunamudi, 19MS098
Shivam Kumar, 19MS123

Instructor :
Dr. Dibyendu Nandi
Dr. Rajesh Kumble Nayak
Dr. Prasanta K. Panigrahi

April 16, 2023

Contents

1 Abstract

Stellar classification is of immense importance as it helps us to study the nature, behaviour and evolution of stellar entities and the universe in general. In this study, we have considered the Stellar classification dataset from the Sloan Digital Sky Survey. The data is analysed for Pearson and Spearman correlations between the different features associated with the stellar entities, that may provide many insights into the underlying physical processes that govern the nature and behaviour of such entities. To further study the behaviour and relationships of the various features, we have undertaken univariate and multivariate analysis. Finally, using this data, a machine learning model (Random Forest) is trained to classify the stellar entities into galaxy, star and quasar. Our model demonstrates that the Random Forest algorithm effectively classifies the stellar entities with high accuracy, which is mainly due to its ability to handle high-dimensional dataset, and by using SMOTE technique to solve the disadvantage of imbalanced data.

2 Introduction

The classification of different stellar structures such as stars, galaxies and quasars using spectral data is an important tool for the astronomers to study and understand the nature and behavior of the many celestial objects in space. These include:

1. Understanding stellar evolution: Studying the properties of stars at different stages can reveal information about their evolution and the processes involved.

2. Understanding the stellar properties: Stellar properties such as mass, radius, temperature and luminosity can be determined from this classification which can give insights into the physical processes inside such celestial systems and their current stage in their evolutionary process.

3. Identification of Exoplanets: By the spectral study of stars including brightness, and luminosity, astronomers can infer the presence of exoplanets orbiting such stars.

4. Analysing Galactic Structure: From stellar classification, astronomers can identify different populations of stars with different ages etc. that compose the galaxies.

5. Cosmology: Stellar classification holds immense importance for the study of the origin and evolution of the Universe.

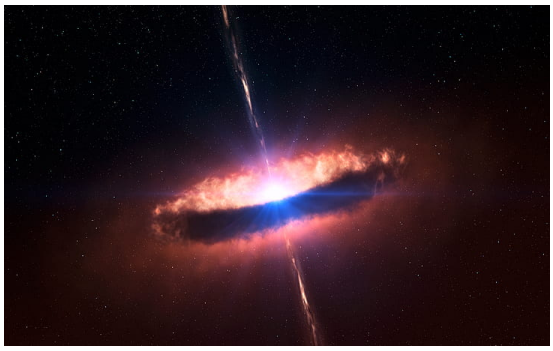
Numerous stellar classifications have been made and stellar classification systems such as Morgan-Keenan have been designed in the past. Here, we have classified stars, galaxies, and quasars based on their spectral characteristics. A brief description of these three class of stellar objects/entities is given as follows:

- Stars - They are primarily self powered, self luminous objects which emit radiation in the visible region or otherwise due to the nuclear fusion occurring in their cores.

- Galaxies - They are vast systems of stars, gas and dust that are held together by gravity. Mainly, they are classified by their shape which can be elliptical, spiral or irregular.

(a) Galaxy

(b) Star



(c) Quasar

Figure 1: Three different classes of stellar entity

- Quasars - Expanded to "Quasi-Stellar objects", which are basically subclass of Active Galactic Nucleus (AGN). They are distant objects that emit enormous amounts of energy, including radio waves and X-rays. They are powered by supermassive black holes at the centers of galaxies.

Classification using spectral data is much more useful as it is based on the physical properties of the celestial objects, it is very precise, has large sample size, consistent and is broadly applicable to classify across different distances and ages of such entities in space.

We performed data analysis on the data called the Stellar Classification Data-set - SDSS1 consists of 10,000 observations of space taken by the Sloan Digital Sky Survey (SDSS) which is a major multi-spectral imaging and spectroscopic redshift survey using a dedicated 2.5-m wide-angle optical telescope at Apache Point Observatory in New Mexico, United States. First, we carried out exploratory data analysis of the different features of the objects to determine their trends and behaviours, so as to better understand the data before doing more complex analysis. Following this, we determined the correlations between the many features in terms of Pearson and Spearman Correlation coefficients. We visualized the trends, distributions and statistics of the various features in the data-set by univariate analysis. To gain insights such as patterns and trends in the relationships between the features in the -data set, we performed multivariate analysis. Finally, we balance the class distribution using the SMOTE technique so as to avoid major misclassification, before training the Random Forest Classifier to predict the class (galaxy, star or quasar) to which the stellar object belongs.

3 Data Overview

The data set 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 which identifies it to be either a star, galaxy or quasar. The detailed overview of the data which includes the stellar object class and the features is given as follows:

Feature	**Description**
obj_ID	Object Identifier, the unique value that identifies the object in the image catalogue used by the CAS (Catalogue Address Space)
Alpha	Right Ascension angle of the celestial object (at J2000 epoch which is the January 1, GMT used as time reference)
Delta	Declination angle of the celestial object (at J2000 epoch)
u	Ultraviolet filter in the photo-metric system, used to measure the flux of light from the celestial object in the UV part of the electromagnetic spectrum.
g	Green filter in the photo-metric system,used to measure the flux of light from the celestial object in the green (visible) part of the electromagnetic spectrum.
r	Red filter in the photo-metric system, used to measure the flux of light from the celestial object in the red (visible) part of the electromagnetic spectrum.
i	Near Infrared filter in the photo-metric system, used to measure the flux of light from the celestial object in the near-infrared part of the electromagnetic spectrum.
z	Infrared filter in the photo-metric system, used to measure the flux of light from the celestial object in the IR part of the electromagnetic spectrum.
run_ID	Run Number used to identify the specific scan that a telescope makes during each phase of the survey
rerun_ID	Rerun Number to specify how the image was processed.Each rerun refers to a particular set of data processing and analysis steps that are applied to the raw telescope data in order to create scientific catalog of stars, galaxy and quasars. It is used to distinguish between the different versions of the data products produced by the SDSS pipeline. It may include updated or refined processing algorithms, changes to calibration procedures or other modifications.
cam_col	Camera column to identify the scanline within the run, this refers to a specific camera used to take images of the sky as part of the SDSS survey.
field_ID	Field number to identify each field, which refers to the part of sky imaged by the SDSS telescope, which is approximately 14 sq. degrees.
spec_obj_ID	Unique ID used for optical spectroscopic objects (Meaning that 2 different observations with the same spec_obj_ID must share the same output class).
class	object class (galaxy, star or quasar object).
redshift	redshift value based on the increase in wavelength. It refers to the shift in the wavelength of light from a distant object towards the red end of the spectrum due to the expansion of the universe.

plate	plate ID, identifies each plate in SDSS (It is the aluminum plate used to obtain spectra of celestial objects in a specific region of the sky during a particular SDSS observation run).
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

4 Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method for analysing data sets in order to summarise their primary characteristics and identify patterns, relationships, and anomalies. It is an essential step in the data analysis process. EDA typically entails producing summary statistics and visualizations, such as histograms, scatterplots, boxplots, and density plots, to gain insights into the data and identify potential problems such as missing data, outliers, or skewness. EDA aims to comprehend the data prior to employing more complex statistical or machine learning models and to guide the selection of suitable methods and techniques for data analysis.

4.1 Understanding the Dataset

Here we begin our analysis by taking a look at the dataset, we simply print a few columns to get the idea. We then look at the Data Type of the features and proceed accordingly.

	obj_ID	alpha	delta	u	g	r	i	z	run_ID	rerun_ID	cam_col	field_ID	spec_obj_ID	class	redshift	plate	MJD	fiber_ID
0	1.237661e+18	135.689107	32.494632	23.87882	22.27530	20.39501	19.16573	18.79371	3606	301	2	79	6.543777e+18	GALAXY	0.634794	5812	56354	171
1	1.237665e+18	144.826101	31.274185	24.77759	22.83188	22.58444	21.16812	21.61427	4518	301	5	119	1.176014e+19	GALAXY	0.779136	10445	58158	427
2	1.237661e+18	142.188790	35.582444	25.26307	22.66389	20.60976	19.34857	18.94827	3606	301	2	120	5.152200e+18	GALAXY	0.644195	4576	55592	299
3	1.237663e+18	338.741038	-0.402828	22.13682	23.77656	21.61162	20.50454	19.25010	4192	301	3	214	1.030107e+19	GALAXY	0.932346	9149	58039	775
4	1.237680e+18	345.282593	21.183866	19.43718	17.58028	16.49747	15.97711	15.54461	8102	301	3	137	6.891865e+18	GALAXY	0.116123	6121	56187	842
5	1.237680e+18	340.995121	20.589476	23.48827	23.33776	21.32195	20.25615	19.54544	8102	301	3	110	5.658977e+18	QSO	1.424659	5026	55855	741
6	1.237679e+18	23.234926	11.418188	21.46973	21.17624	20.92829	20.60826	20.42573	7773	301	2	462	1.246262e+19	QSO	0.586455	11069	58456	113
7	1.237679e+18	5.433176	12.065186	22.24979	22.02172	20.34126	19.48794	18.84999	7773	301	2	346	6.961443e+18	GALAXY	0.477009	6183	56210	15
8	1.237661e+18	200.290475	47.199402	24.40286	22.35669	20.61032	19.46490	18.95852	3716	301	5	108	7.459285e+18	GALAXY	0.660012	6625	56386	719
9	1.237671e+18	39.149691	28.102842	21.74669	20.03493	19.17553	18.81823	18.65422	5934	301	4	122	2.751763e+18	STAR	-0.000008	2444	54082	232

Figure 2: Visualizing the first 10 columns of the dataset

We observe that all the columns have 100000 non-null values and 0 NaN or null values, which is optimal. We won't be needing any techniques to handle null values.

From the Data Type plot we see that, out of the 18 data columns 10 are of float64 type and 7 are of int64 type and only the class column is not numerical and is an object type data. We'll have to convert the object type data field to a numerical type for our further analysis. We simply replace 'GALAXY', 'STAR' and 'QSO' with 0, 1 and 2 respectively to convert them to numerical type.

We can also see that there are a lot of columns denoting several different types of IDs, time and dates, etc *(obj_ID, run_ID, rerun_ID, cam_col, field_ID, spec_obj_ID, plate, MJD, fiber_ID)*. We can simply drop these columns for our analysis as they are only for identification and won't provide us with any new information.

```
Data columns (total 18 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   obj_ID       100000 non-null  float64
 1   alpha        100000 non-null  float64
 2   delta        100000 non-null  float64
 3   u            100000 non-null  float64
 4   g            100000 non-null  float64
 5   r            100000 non-null  float64
 6   i            100000 non-null  float64
 7   z            100000 non-null  float64
 8   run_ID       100000 non-null  int64
 9   rerun_ID     100000 non-null  int64
 10  cam_col      100000 non-null  int64
 11  field_ID     100000 non-null  int64
 12  spec_obj_ID  100000 non-null  float64
 13  class        100000 non-null  object
 14  redshift     100000 non-null  float64
 15  plate        100000 non-null  int64
 16  MJD          100000 non-null  int64
 17  fiber_ID     100000 non-null  int64
dtypes: float64(10), int64(7), object(1)
```

Figure 3: Data type of features and Non-Null count

4.2 Data Distribution

We then take a look at the data distribution. We plot the data count with respect to each class using a combination of matplotlib and seaborn packages.

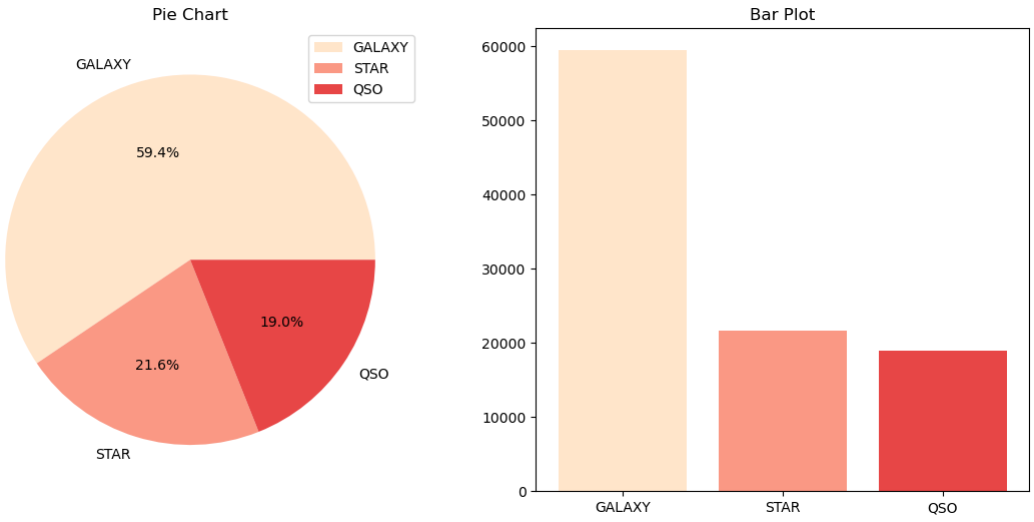


Figure 4: Pie chart and Histogram plot of the data distribution

From the plots, we can see that we have a non-uniformly distributed data. The total number of data points for galaxy is 59.4%, stars is 21.6% and for Quasars is 19.0%.

4.3 Correlation Analysis

The purpose of correlation analysis is to measure the strength and direction of the relationship between two variables. A correlation analysis yields a correlation coefficient with a range of -1 to +1. A positive correlation coefficient indicates that the two variables are positively related, i.e., an increase in one variable is associated with an increase in the other. A negative correlation coefficient indicates that the two variables are negatively related; an increase in one variable is associated with a decrease in the other, and vice-versa.

A correlation coefficient of zero indicates that the two variables have no relationship. In research and data analysis, correlation analysis is frequently used to determine whether and to what extent two variables are related. It is essential to note, however, that correlation does not imply causation, and that other variables may be involved in the relationship between the two variables.

Pearson correlation and Spearman correlation are two of the most commonly used statistical methods for correlation analysis. They both tell us the direction and the strength of the association, but they use different formulas and work better with different types of data.

4.3.1 Pearson Correlation

Pearson correlation measures the linear relationship between two continuous variables. To calculate Pearson correlation, we need to find the mean and the standard deviation of both variables. Then we standardize each observation by subtracting the mean and dividing by the standard deviation. Then we multiply the standardized values of both variables, add them up, and divide by the sample size minus one. This gives us the Pearson correlation coefficient, or r.
Mathematically it is given as :

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})2}\sqrt{\sum_{i=1} n(y_i - \bar{y})^2}}$$

where:

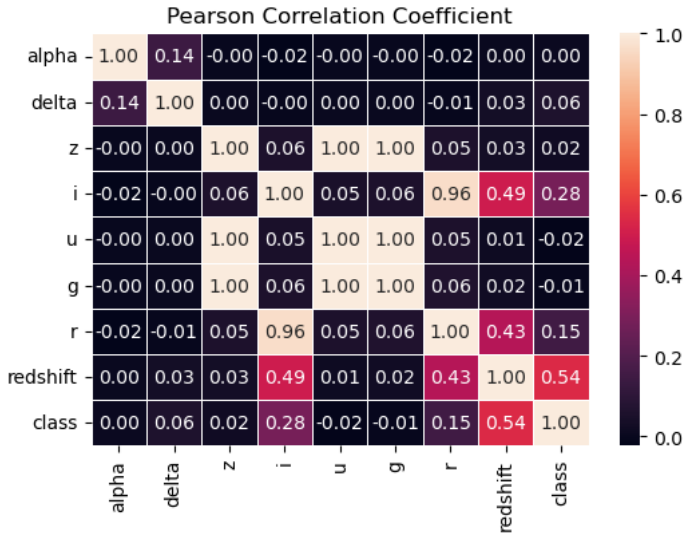
r_{xy} is the Pearson correlation coefficient between x and y.
x_i and y_i are the ith values of x and y, respectively.
$\bar{x}$ and $\bar{y}$ are the sample means of x and y, respectively.

Pearson correlation is useful when we want to study the strength of the association between two variables. Pearson correlation works well when there is a linear relationship between two variables. However, the Pearson correlation has some limitations. It is not suitable for data that is skewed or has outliers. It also assumes that the variables have a linear relationship, which may not always be true in real life.

We generated the correlation coefficients of all the features with respect to each other using the pandas library's corr function. From the figure (5a), we can clearly see the correlation coefficients of the features with respect to others. Feature **z** has a correlation coefficient of (1.00) with respect to both **u** and **g**, which implies that they are very strongly correlated. Other features like **g** and **u** (1.00), **r** and **i** (0.96), etc are also highly correlated.

However, in this case, we are most concerned with the correlation of all these features with respect to feature **class**. This would help us identify important features for the stellar object classifier. Features with some degree of correlation are chosen as features with very little correlation won't help in classifying between different class of objects. Figure (5b) shows the correlation of features with respect to **class** in descending order. Features like **redshift** (0.536), **i** (0.284) and **r** (0.151) seem to be the most correlated with **class** in this case and **delta** being the least.

Pearson's correlation only considers linear relationship, for non linear monotonic correlation we take a look at the Spearman correlation coefficients.



(a) Pearson correlation coefficient matrix

```
class       1.000000
redshift    0.536822
i           0.284396
r           0.150691
delta       0.056643
u           0.017701
z           0.017352
g           0.005915
alpha       0.004552
Name: class, dtype: float64
```

(b) Pearson correlation coefficient with respect to *class*

Figure 5

4.3.2 Spearman Correlation

Spearman correlation is a method of measuring how well two variables are related by using their ranks instead of their actual values. This is different from the Pearson correlation, which uses the numerical values and assumes a linear relationship. To calculate Spearman correlation, you need to rank each variable from lowest to highest and then find the difference between the ranks for each pair of observations. Then you square the differences and add them up. Finally, you divide the result by a number that depends on the sample size. This gives you the Spearman correlation coefficient, also called rho (ρ) or rank correlation.

$$\rho = 1 - \frac{6 \sum d_i 2}{n(n2 - 1)}$$

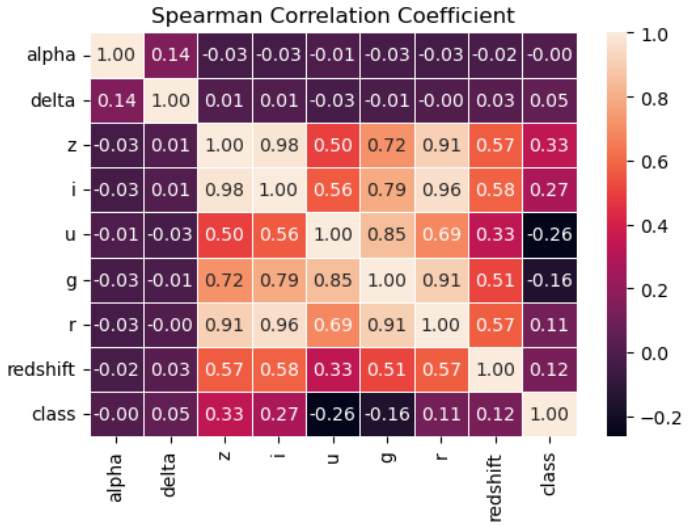
where:

ρ is the Spearman correlation coefficient.
d_i is the difference between the ranks of corresponding values of x and y.
n is the number of observations.

Spearman correlation is useful when the variables are not normally distributed or when the relationship is not linear. Spearman correlation assumes that the variables have a monotonic relationship, which means that their ranks do not change drastically. If the relationship is non-monotonic, Spearman correlation may not be appropriate. Also, Spearman correlation may be less sensitive to linear correlations between continuous variables than Pearson correlation.

Figure (12a) shows the Spearman correlation coefficient matrix. In this case, a lot of features are highly correlated with each other, like **i** and **z** (0.98), **r** and **g** (0.91), **z** and **r** (0.91), etc. We also see some negative correlation coefficients in this case, **u** and **class** (-0.26), and **g** and **class** (-0.16). These indicate negative correlation among them.

As said before, we are most concerned with correlation coefficients with respect to



(a) Spearman correlation coefficient matrix

```
class       1.000000
z           0.329337
i           0.267524
u           0.264597
g           0.160557
redshift    0.122859
r           0.109295
delta       0.054679
alpha       0.000375
Name: class, dtype: float64
```

(b) Spearman correlation coefficient with respect to *class*

Figure 6

feature **class**. Figure (12b), shows the correlation of features with respect to **class** in descending order. In this case, **z**, **i** and **u** are the three most correlated features with **class**. Feature **delta** and **alpha** being the least two.

From our correlation analysis, we observe that most of the blocks in Pearson correlation show a correlation very close to 0, while only a few blocks are close to 0 in the case of Spearman correlation. Thus we can infer that most of the features hold a non-linear dependency.

We can also see that both the features **alpha** and **delta** have correlation coefficients less than or equal to 0.05 for both Pearson and Spearman correlation. It is not a surprise, as alpha and delta are the ascension and declination angles of stellar objects, which give their positions on the celestial sphere.So, we can conclude that the different classes of stellar objects are randomly scattered throughout the celestial sphere. And so, alpha and beta don't contain any classification information. Simply put, we don't expect stars to be placed at a particular region and galaxies at some other on the celestial sphere.
So, on the basis of this correlation analysis, we drop **alpha** and **delta** from our dataset as they are not of much importance in our further analysis.

4.4 Univariate analysis

Using the selected features from the correlation analysis, we move on to Univariate analysis. Univariate is a statistical technique used to describe and explain the behaviour of a single variable in a data set. In exploratory data analysis (EDA), univariate analysis is frequently used to understand a variable's characteristics and spot any outliers or unusual observations. Visualising data distribution with a histogram or box plot, density plots, comparing groups or evaluating the significance of differences between variables with statistical tests like the chi-squared test are examples of univariate analysis.
Here, we will analyse our data using box plots and density plots.

4.4.1 Box Plot

Box plots are commonly used to summarize the distribution characteristics of a data set by visualizing them through quartiles. The box in the plot represents the middle 50% of the data, extending from the lower quartile (25th percentile) to the upper quartile (75th percentile). The line inside the box represents the median or the middle value of the data set. The whiskers extend from the box to represent the minimum and maximum values in the data set. Any data points outside the whiskers are considered outliers and are plotted as individual points, which can help to identify extreme or unusual values in the data. Box plots are also useful to compare multiple data sets side-by-side, for detecting skewness, symmetry or clustering in data.

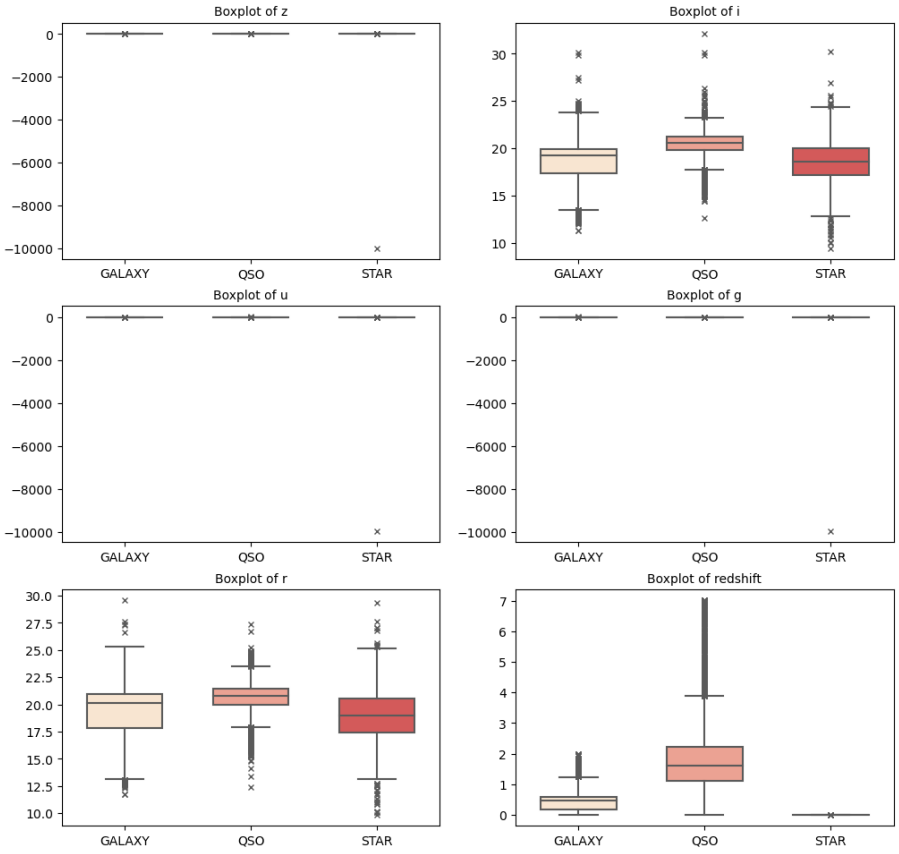


Figure 7: Box plot of data showing erroneous values in plots of z, u and g

Figure (7) shows the box plot (drawn separately for each class) of the selected features in the dataset. Just at the first glance, we can identify that there's something unusual with the plots of **z**, **u** and **g**. We can see that there's an erroneous value, which is causing the problem. To remove it we can simply find the index of those three erroneous values (by finding the minimum) and drop them from the dataset and plot it. Figure (8) shows the new plot after removing those three erroneous values.

Now, that we have the box plots of every feature, we can quickly identify some key observations from the plots.

1. We can get an idea about the distribution's mean and variance (spread), middle 50% of data with the lower and upper quartiles.

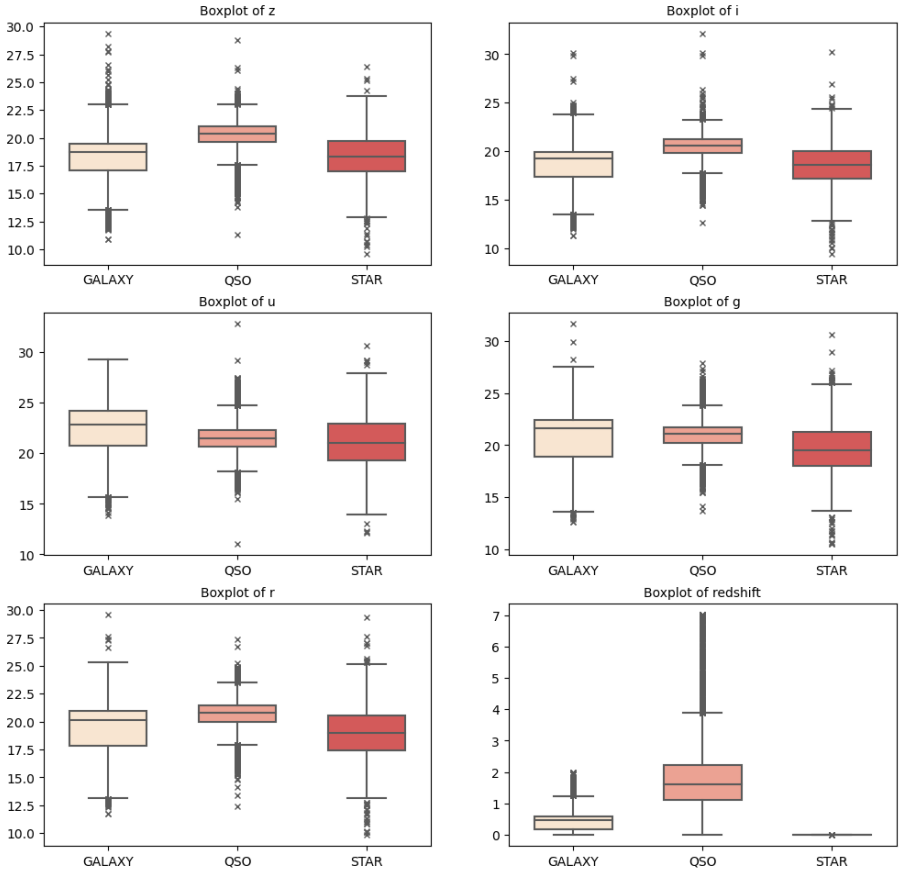


Figure 8: Box plot of data

2. The boxplot of the redshift feature for the STAR class shows that all the values are very close to 0, this can be used by the model. For the model to categorize a data point as the STAR class with maximum accuracy should have a redshift value of 0.

3. Boxplots can help in identifying outliers in the dataset. As we can see, there are a lot of outliers present in each feature.

4.4.2 Outlier Removal

Outliers are data points that are significantly different from other data points in the dataset, and they can arise due to errors in data collection, measurement errors, or simply due to natural variations in the data. Outliers can have a disproportionate effect on statistical measures such as the mean and standard deviation, which can skew the results and lead to incorrect conclusions.

From figure (8), we can see that there are a lot of outliers in the dataset for each feature. For removing the outliers, we simply identify the data points which are in the lower and upper (25 percentile) quartile. Data points below or above respectively those two limits are dropped from the dataset. Figure (9) shows the dataset after removing the outliers.

After the removal of outliers and all other previous analysis, we are left with a dataset of 93,954 (100,000 initially) observations and each observation is described by 7 feature columns and 1 class column.

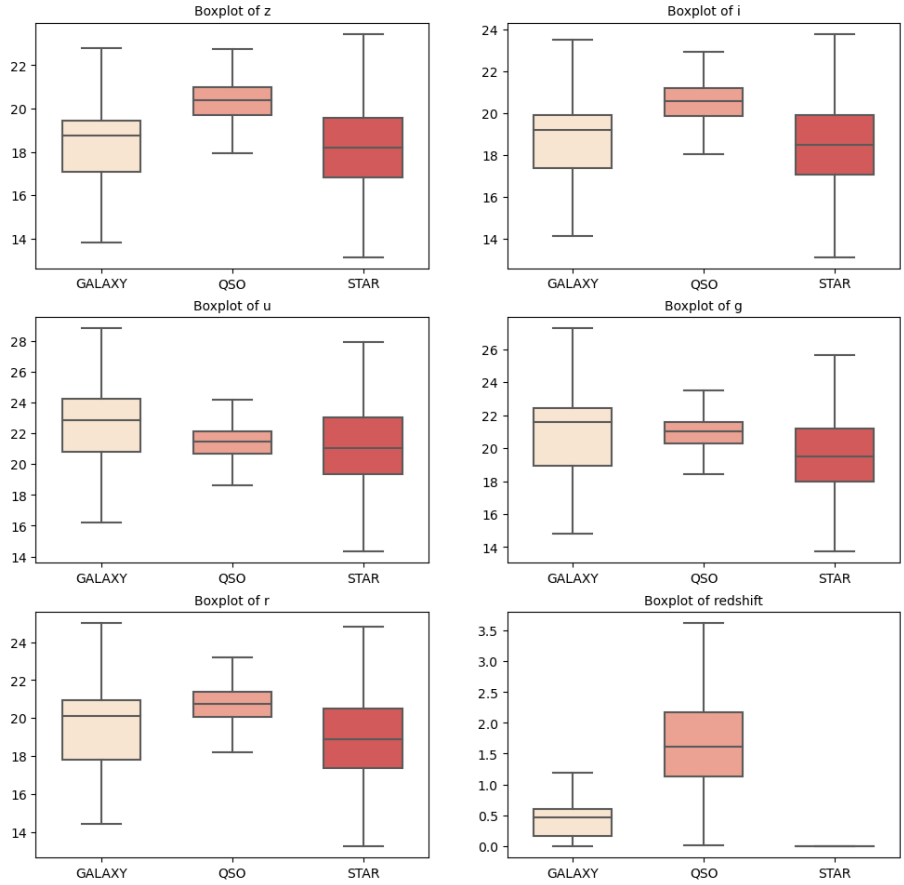


Figure 9: Box plot of data after outlier removal

4.4.3 Density Plot

From box plots, we now move to a different technique of univariate analysis i.e. density plots. A density plot is a useful graphical tool for visualizing the distribution of a data set. It creates a continuous representation of the data distribution by smoothing the histogram of the data set. The x-axis of the density plot represents the data range, while the y-axis represents the density of the data. Density plots are commonly used for exploratory data analysis (EDA). They are useful for identifying the shape of a data distribution, such as bimodal or skewed data, where the data is more or less compact around a particular value or range.

Figure (10) shows the density plot of all the features based on each class. We can make observe the following things from the density plot,

1. The classes in each density plot for a given feature are overlapping, which means, we cannot categorize the data by using simple logical statements. We need to have statistical models to categorize the classes of the data.

2. The density plot of the redshift feature for the **STAR** class conforms with the above boxplot.

3. All redshift values corresponding to class **STAR** lie very close to zero, and create a massive peak, making it difficult to see the distribution for the other class.

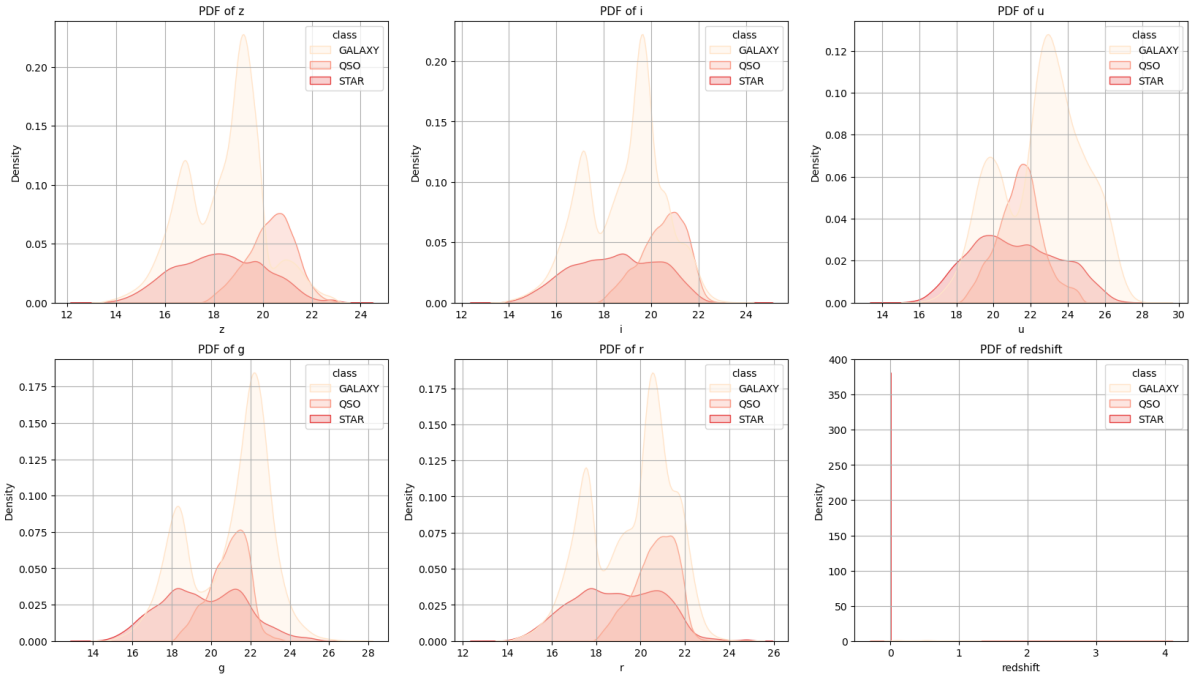


Figure 10: Density plot of the data

4. We can also notice that all density plots corresponding to **GALAXY** class are bimodal.

4.5 Multivariate analysis

Multivariate analysis is a statistical technique used to study the relationship between multiple variables in a data set. It involves analyzing and interpreting relationships between multiple derived measures or variables and can be used to investigate the data structure, identify patterns and trends, and perform clustering. It is not only useful for finding correlations and relationships between different variables but also can lead to a greater understanding of how each variable behaves with respect to others. One of the most common techniques in multivariate analysis is a pair plot.

4.5.1 Pair Plot

A pair plot, also known as a scatter plot matrix, is a graphical representation of the pairwise relationships between variables in a data set. With more than two variables in a data set, pair plot is useful to quickly identify relationships, patterns, and potential outliers among multiple variables. Figure (11) shows the pair plot of all the features with respect to each other, with different classes being colored differently.

We can clearly see, that features **z**, **i**, **u**, **g**, and **r** are positively correlated to each other. This was also observed in the correlation analysis. Whereas, in the case of **redshft**, we see some sort of clustering. **QSO** seem to have higher level of redshift, followed by **GALAXY** and then **STAR** with almost zero redshift.

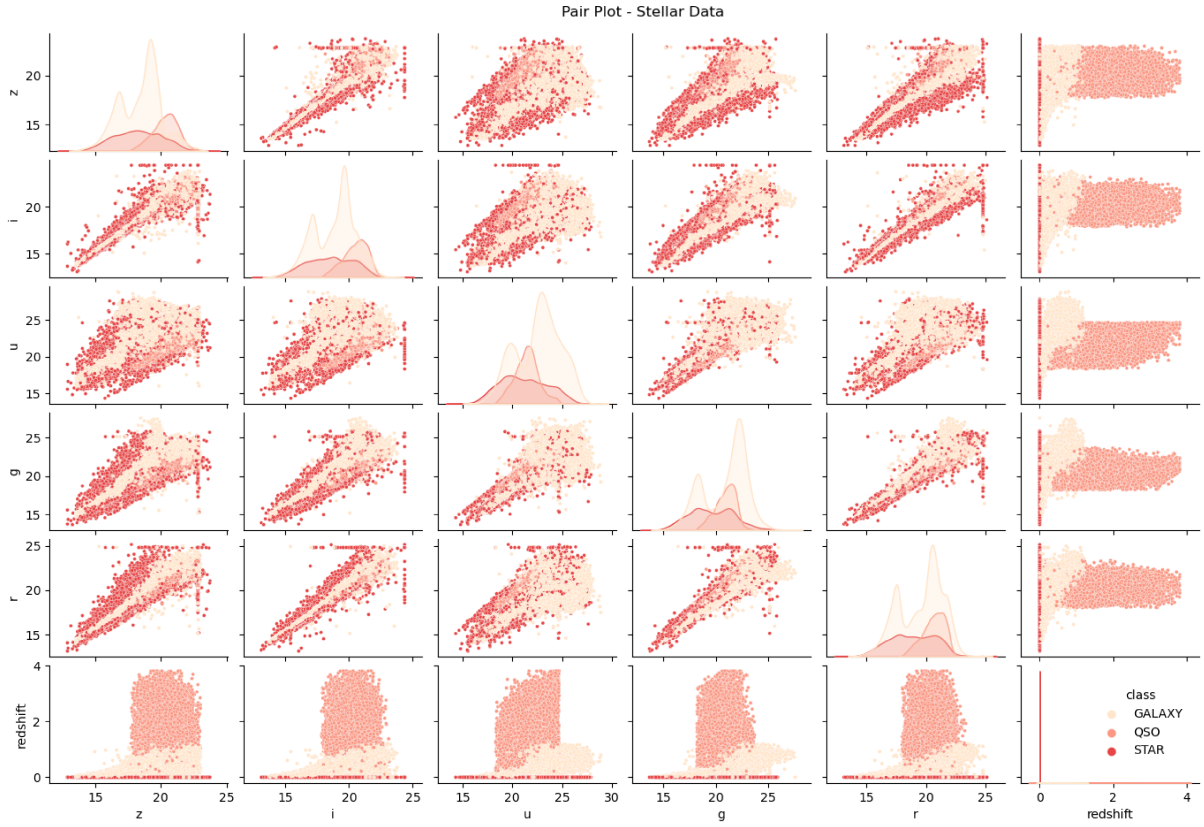


Figure 11: Pair plot of the data

4.6 Data Balancing

The major roadblock that any machine learning algorithm faces before making accurate predictions is the **Imbalanced Data Distribution**, where observations in one of the classes are much higher or lower in number than the other classes. As a result, they tend only to predict the majority class, hence, having major misclassification of the minority class in comparison with the majority class.

There are mainly two ways of handling such data imbalance cases which are oversampling the minority cases, or under-sampling the majority cases while modelling the data. SMOTE, one of the most well-known techniques for data balancing, belongs to the former type. We have addressed this problem using the SMOTE technique as mentioned below.

4.6.1 SMOTE

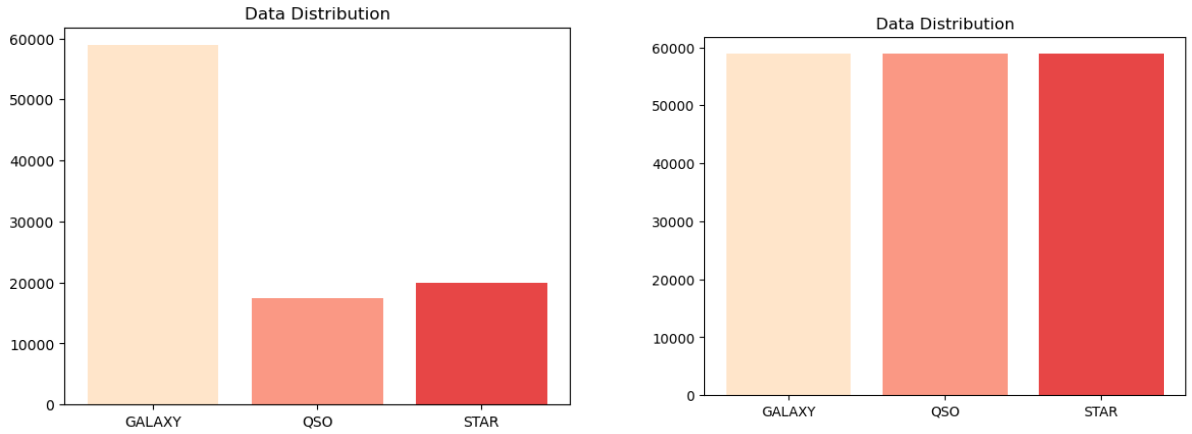
SMOTE or Synthetic Minority Oversampling Technique aims to balance class distribution by synthesizing new minority instances between existing minority instances. These synthetic minority classed are generated by randomly selecting one or more of the k-nearest neighbors for each example in the minority class. The data is reconstructed after the oversampling process to be used for classification. The following steps are followed in SMOTE:

1. *Step 1:* Given the minority class set **A**, for each x∈**A** the k-nearest neighbors of x are obtained by calculating the Euclidean distance between x and every other sample in set A.

2. *Step 2:* The sampling rate N is set according to the imbalanced proportion. For each x∈**A**, N examples are randomly selected from its k-nearest neighbors to construct the set A_1.

3. For each element x_n in A_1, new example is generated by:

$$x^{'} = x + rand(0,1) \times |x - x^{'}| \tag{1}$$



(a) Data distribution before SMOTE

(b) Data distribution after SMOTE

Figure 12

Before data balancing, the number of observations corresponding to **GALAXY**, **QSO** and **STAR** were 58869, 17371 and 19946 respectively. After SMOTE, the number of observations for all classes was made equal to 58869.

5 Modeling

5.1 Data Normalization

The first step in modelling the data is Data Normalization, it is a technique used to transform the values of a dataset to a common scale. This is due to the fact that machine learning algorithms are sensitive to scale of the input features. Here, we have used Scikit-learn's (machine learning Python library) StandardScaler for the same. It is a preprocessing step which standardizes the features of a dataset by removing the mean and scaling to unit variance. It ensures that features with larger scales do not dominate the learning process, which could lead to biased results.

5.2 Random Forest Classifier

Once the data is normalised, the next step is to select a classification algorithm. For the current task, we have decided to go with a Random Forest Classifier. Random Forest Classifier algorithm is an example of a supervised learning algorithm, where a model is trained on a labelled dataset to predict output labels based on input features. In the case of the random forest classifier, the output is a categorical variable, as in this case, whether the object belongs to class STAR, GALAXY or QSO, based on its input features.

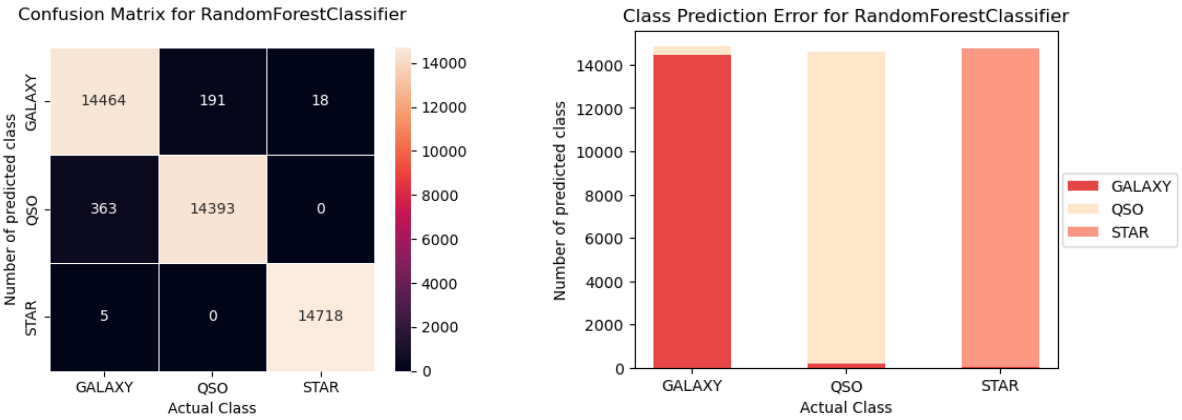
The random forest classifier algorithm selects a random subsample of data, and from this sample, a decision tree is constructed in a way that optimizes the prediction accuracy. The algorithm repeatedly does this with different samples of data and produces as many decision trees as needed. Each tree in the random forest works by partitioning the data into smaller and smaller sets, ultimately classifying each data point into one of the output classes. Random Forest Classifiers are generally known for their accuracy, ability to handle high-dimensional data, resistance to overfitting and robustness.

Before training the Random Forest Classifier, the dataset was randomly divided into training and test sets using the train_test_split function from the scikit-learn library. The Training Set contained 75% of the data and the Test Set had the remaining 25%. The classifier was trained on the Training Set. The training was completed in under a minute. The trained model was then tested on Test Set and the model obtained an accuracy of **98.693 %**.

The classification test reports along with the confusion matrix and class prediction error plots are shown below,

	Precision	**Recall**	**F1-score**	**Support**
GALAXY	0.9752	0.9858	0.9804	14673
QSO	0.9869	0.9754	0.9811	14756
STAR	0.9988	0.9997	0.9992	14723
Macro Avg	0.9870	0.9869	0.9869	44152
Weighted Avg	0.9870	0.9869	0.9869	44152
Accuracy			**0.9869**	44152

Table 2: Classification report with Accuracy, Precision, Recall and F1-scores



(a) Confusion Matrix

(b) Class Prediction Error plot

Figure 13: Random Forest evaluation matrix and plot

6 Conclusion

In conclusion, our analysis of the stellar classification dataset has provided us with insights into the relationship between the spectral emissions of the stellar objects in the different regions of the emission spectrum. We calculated the correlation coefficients of the features and observed key correlations among them. We performed univariate and multivariate analyses to understand the behaviour and relationships amongst the different features and removed outliers that can lead to errors during classification. To avoid the misclassification of minor classes, we performed data balancing via the SMOTE technique, followed by data normalization as classification models are sensitive to scale. Lastly, we trained the Random Forest Classifier to predict the class to which the object belongs. Our model was able to predict the class of stellar object with a high accuracy of 98.693%, and we got the precision, recall and F1 score for the three classes.

7 Acknowledgement

We acknowledge the use of data from the Sloan Digital Sky Survey (SDSS). The SDSS is a joint project of the University of Chicago, Fermilab, the Institute for Advanced Study, the Japan Participation Group, Johns Hopkins University, the Max-Planck-Institute for Astronomy (MPIA), the Max-Planck-Institute for Astrophysics (MPA), New Mexico State University, Princeton University, the United States Naval Observatory, and the University of Washington. Funding for the SDSS and SDSS-II has been provided by the Alfred P. Sloan Foundation, the Participating Institutions, the National Science Foundation, the U.S. Department of Energy, the National Aeronautics and Space Administration, the Japanese Monbukagakusho, the Max Planck Society, and the Higher Education Funding Council for England. To visit the website, click on the link: Sloan Digital Sky Survey

We also acknowledge our professors Prof. Dibyendu Nandi, Prof. Rajesh Kumble Nayak, Prof. Prasanta Kumar Panigraphi for giving us an excellent opportunity to learn Astronomical Data Science and execute this project under their guidance.

8 Data Resource

Data for Stellar Classification:
`https://www.kaggle.com/datasets/fedesoriano/stellar-classification-dataset-sdss17`

9 References

- The Seventeenth Data Release of the Sloan Digital Sky Surveys: Complete Release of MaNGA, MaStar, and APOGEE-2 Data, Abdurro'uf et al 2022 ApJS 259 35
- Archival Data Analysis Project - Astronomy Group, Stony Brook University
- Understanding Redshift | Astronomy | Fuseschool
- Correlation Analysis | Pearson and Spearman Correlation

- ML | Data Normalization
- ML | Outlier Removal
- ML | Handling Imbalanced Data with SMOTE and Near Miss Algorithm in Python
- ML | Random Forest Classifier
- Evaluation Matrices | Confusion Matrix

10 Source Code

The source code for the data analysis and classification model can be found by visiting the: Source Code