Using machine learning to classify and localize stellar objects

Ahmed Taha Hassina

Abstract: Mapping the universe has always been a salient endeavor in astronomy and astrophysics. Advancements in observational astronomy have generated vast amounts of data containing various features of celestial objects. Inducing a growing need for accurate and detailed classification and localization of stellar objects in the cosmos. In this paper, we present a comprehensive study that combines machine learning techniques to classify celestial objects into distinct categories and predict their precise locations in the sky. This study is divided into two parts: a classification task, where the stellar objects are classified into galaxies, stars, or quasars (quasi-stellar radio sources). The resulting model exhibits exceptional performance in differentiating these objects, as demonstrated by high classification accuracy. We extend our analysis to predict the location of stellar objects using regression techniques. By employing multi-target regression, we model the right ascension and declination coordinates, enabling accurate localization of celestial objects on the celestial sphere. The practical implications of our research lie in producing comprehensive celestial catalogs, facilitating targeted observations, and contributing to the broader field of observational astronomy. The ability to accurately classify and localize stellar objects lays the groundwork for mapping the cosmos and advancing our understanding of the universe's intricate structure.

Keywords: *galaxies, stars, Quasars, machine learning, observational astronomy, localization.*

1. Introduction

Advancements in technology have revolutionized the field of astronomy, enabling us to explore the vast depths of the universe like never before. This development within the sphere of astrophysics led to an increasing volume of astronomical data generated by advanced telescopes and observatories such as the **S**loan **D**igital **S**ky **S**urvey (SDSS), making this data more accessible. Consequently, the need for efficient and accurate data processing techniques has become paramount. Machine learning, a branch of artificial intelligence, has emerged as a powerful tool in the analysis of astronomical data, offering novel solutions to various challenges faced by astronomers. Using one of the most robust machine learning algorithms, The Random Forest, this research aims to study stellar observations released in January 2023 by the SDSS as the survey's 18th data release. The entities detected by the observatory providing this data are either galaxies, stars or Quasars (Quasi-stellar radio source). Galaxies are vast, majestic systems that serve as the building blocks of the universe. They are immense collections of stars, planets, gas, dust, and dark matter bound together by gravity. Galaxies come in a remarkable variety of shapes, sizes, and configurations, ranging from small, dwarf galaxies with a few million stars to massive elliptical galaxies harboring trillions of stars [1]. Taking us to our second object, stars serve as the fundamental building blocks of galaxies. These luminous spheres of hot, glowing gas are held together by gr-

vity and emit light and heat through nuclear fusion reactions in their cores. Stars are scattered throughout the universe, forming intricate patterns known as constellations and playing a crucial role in shaping the cosmos [2]. Quasars, on the other hand, are incredibly powerful and distant celestial objects that emit enormous amounts of energy across the electromagnetic spectrum. These enigmatic entities represent one of the most fascinating and mysterious phenomena in the universe. Quasars were first discovered in the 1960s as extremely bright, point-like sources of light that resembled stars in optical telescopes. However, further observations revealed that their spectra were highly unusual, exhibiting a characteristic redshift, indicating that they are located at extreme distances from Earth. The light emitted by quasars carries crucial information about the early universe. Due to their extreme distances, quasars provide a glimpse into the universe's distant past, allowing astronomers to study the cosmos in its infancy. By analyzing the spectra of quasars, scientists can probe the chemical composition of the intergalactic medium and investigate the conditions of the universe during its early epochs. Following this logic, the classification and localization of these celestial bodies play a crucial role in the understanding of the expansion of our universe and its composition. This study will use Python language due to its numerous applications in machine learning, the sufficiency of its libraries for this task and its simplicity and expansibility. For the classification task, the Random Forest Classifier will be used to predict the classes of the celestial objects. For the regression task, the Random Forest Regressor will be used to predict the coordinates of these entities. After each task, the accuracy of the model will be evaluated and the structure of the Random Forest studied.

2. Methodology

2.1. Machine learning for astronomy and astrophysics

Machine learning is a subfield of artificial intelligence that uses data to enable computers to learn and make predictions without being explicilty programmed for the specific task, involving the development of algorithms and statistical models [3]. The core idea behind machine learning is to allow machines to improve their performance on a given task through experience and exposure to relevant data. By leveraging statistical techniques and pattern recognition, machine learning algorithms identify patterns, trends, and relationships within the data, enabling them to generalize and make accurate predictions on new, unseen data [4]. Machine learning has found diverse and impactful applications in astronomy and astrophysics and and is gaining popularity as a vaiable alternative for manual processing and computationally intensive template-based matched filtering algorithms. It has revolutionazed the way astronomers analyse and intepret vast amounts of data. Additionally, machine learning is instrumental in identifying rare and transient events in astronomical surveys, enabling the discovery of new phenomena and celestial objects. Furthermore, in the field of cosmology, machine learning techniques help analyze large-scale structures and extract valuable insights from complex datasets, leading to a deeper understanding of the universe's evolution.

2.2. The Random Forest Algorithm

Random Forest is an ensemble supervised learning algorithm used for both classification and regression tasks. It is based on the concept of decision trees and combines the predictions of multiple individual decision trees to make more accurate and robust predictions. The algorithm works by creating a multitude of decision trees

during the training phase, each using a random subset of features [5]. In classification, the final prediction is determined through a majority vote of the individual trees, while in regression, it is computed as the average of the predictions from each tree. Random Forest's ability to handle high-dimensional data, reduce overfitting, and capture complex relationships between features makes it a widely used and powerful tool in various domains, including astronomy, where it has been successfully applied to tasks such as object classification, localization of cosmic entities and redshift estimation.

2.3. Data

SDSS-V is the first facility providing multi-epoch optical & IR spectroscopy across the entire sky, as well as offering contiguous integral-field spectroscopic coverage of the Milky Way and Local Volume galaxies. This panoptic spectroscopic survey continues the strong SDSS legacy of innovative data and collaboration infrastructure [6]. This study will use the spectrar features observed by this survey as training data. The data consists of 100,000 stellar object observation: 52343 galaxy (52.3%), 37232 star (37.2%) and 10425 Quasar (10.4%) (figure 1). Each observation is described by 42 features and one class column classifying the observations. Features include the object identifiers (Objid) and (Specobjid); J2000 celestial coordinates : right ascension (ra) and declination (dec); the final redshift of the celestial object (redshift); the run and rerun numbers: the run refering to the specific period in which the survey observed a part of the sky and the rerun refering to the preprocessing of the obtained data; The camera column (camcol); the number of the sky field in which the observation was obtained (field); the number of physical glass plates mounted on the telescope (plate); the ID of the oplical fiber responsible for gathering the object's light (fiberID); Modified Julian Date represents the number of days that have passed since midnight Nov. 17, 1858. It is used in SDSS to keep track of the time of each observation (mjd); petroRad_u, petroRad_g, petroRad_r, petroRad_i, and petroRad_z are the petrosian radii for the five photometric bands u (ultraviolet), g (green), r (red), i (infrared), and z (near-infrared) respectively; petroFlux_u, petroFlux_g, petroFlux_r, petroFlux_i, and petroFlux_z are the petrosian fluxes for the five photometric bands; these features describe the total amount of light emitted from the celestial objects. petroR50_u, petroR50_g, petroR50_r, petroR50_i, and petroR50_z are the petrosian half-light radii for the five photometric bands. psfMag_u, psfMag_g, psfMag_r, psfMag_i, and psfMag_z are the magnitudes of objects measured using the Point Spread Function (PSF) in the five photometric bands. expAB_u, expAB_g, expAB_r, expAB_i, and expAB_z - axis ratio of exponential fits to the light profile of celestial objects observed in the five photometric bands.

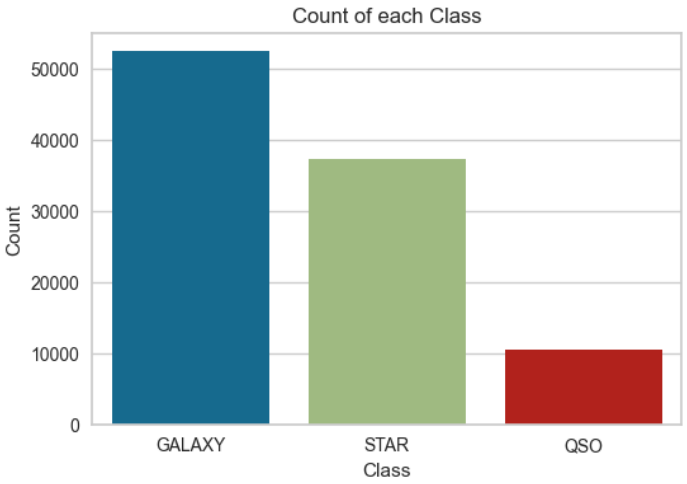

Figure 1. Class size and distribution

2.4. Stellar classification

2.4.1. Data preprocessing

Machine learning algorithms typically require numerical data as input, and converting the target labels into numeric values enables the algorithms to work effectively. In this case, assigning 0 to "GALAXY," 1 to "STAR" and 2 to "QSO" allows the algorithm to treat the three different classes as distinct numerical categories.

Outlier detection is performed in data preprocessing to identify and handle data points that are significantly different from the majority of the data. Outliers are data points that deviate substantially from the typical patterns in the dataset and can have a significant impact on the performance of machine learning models.Outlier detection and elimination ensures data quality, model performance and robustness, and interpretability. This study will use the Local Outlier Factor (LOF), it is a popular outlier detection algorithm that measures the local density deviation of a data point with respect to its neighbors. It assigns an anomaly score to each data point based on its local density compared to the densities of its neighbors. The LOF algorithm can be used to identify data points that are significantly less dense than their neighbors, which are likely to be outliers [7]. First, for each data point in the dataset, the distance to its k-nearest neighbors (k-distance) is computed. The value of k is a user-defined parameter and determines the number of neighbors to consider. The k-distance provides an estimate of how close or far away a data point is from its k-nearest neighbors. Second, the reachability density (RD) is calculated. The reachability density or reachability distance between Xi and Xj is defined as the maximum of the k-distance of Xj and the actual distance between Xi and Xj. In layman terms, if a point Xi lies within the K-neighbors of Xj, the reachability distance will be K-distance of Xj, else, reachability distance will be the distance between Xi and Xj .

$$RD(Xi, Xj) = max(K - dis(Xj), dis(Xi, Xj)) \quad (1)$$

Third, the local reachability density will be calculated. It is inverse of the average reachability distance of a given data point A from its neighbors. Intuitively according to LRD formula, the more the average reachability distance (i.e., neighbors are far from the point), the less density of points are present around a particular point. This tells how far a point is from the nearest cluster of points. Low values of LRD implies that the closest cluster is far from the point.

$$LRD_k = \frac{1}{\Sigma Xj \in N\square(A) \frac{RD(A, Xj)}{||N\square(A)||}} \quad (2)$$

Lastly, the Local Outlier Factor (LOF) is calculated. the LRD of each point is used to compare with the average LRD of its K neighbors. The LOF is the ratio of the average LRD of the K neighbors of A to the LRD of A. Intuitively, if the point is not an outlier (inlier), the ratio of average LRD of neighbors is approximately equal to the LRD of a point (because the density of a point and its neighbors are roughly equal). In that case, LOF is nearly equal to 1. On the other hand, if the point is an outlier, the LRD of a point is less than the average LRD of neighbors. Then LOF value will be high. However, selecting a threshold of LOF > 1 is not a rule, in a matter of fact, it is usually inconclusive [7].

$$LOF\square(A) = \frac{\Sigma Xj \in Nk(A)\ LRD\square(Xj)}{||N\square(A)||} \bullet \frac{1}{LRD\square(A)} \quad (3)$$

The main hindrance with imbalanced data is that there are too few examples of the minority class for a model to effectively learn the decision boundary. One way to solve this problem is to oversample the examples in the minority class. This can be achieved by simply duplicating examples from the minority class in the training dataset prior to fitting a model. This can balance

the class distribution but does not provide any additional information to the model. As can be seen in the histogram representing our three classes (figure 1), there is a large imbalance in our data. This study will use the built in imblearn's Synthetic Minority Oversampling TEchnique (SMOTE), The methodology of the SMOTE function is to select k examples that are close in the feature space, draw a line between the examples in the feature space and drawing a new sample at a point along that line in the prospect of creating more comprehensive samples for the minority data [8]. After oversampling the data, each class would have a count of 52343 as shown in figure 2.

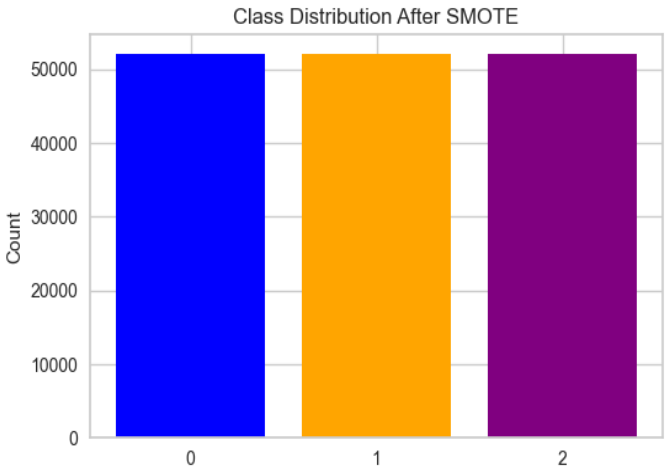

Figure 2. Class count and distribution after oversampling

Feature selection is a crucial step in improving the accuracy of a predictive model. Feature selection methods can be used to identify and remove unneeded, irrelevant and redundant attributes from data that do not contribute to the accuracy of a predictive model or may in fact decrease its accuracy. Fewer attributes is desirable because it reduces the complexity of the model, and a simpler model is simpler to understand and explain. In this study, the correlation of the 42 features and the object's class was carefully analyzed in order to select which features have important roles in the prediction process. To ensure careful feature selection, the built in corr() Python function is used to display the correlation between all the features on the class column. This code returns numbers ranging from -1 to 1. this range represents whether a feature's correlation with the class column is negative, positive or equal to zero as shown in figure 3.

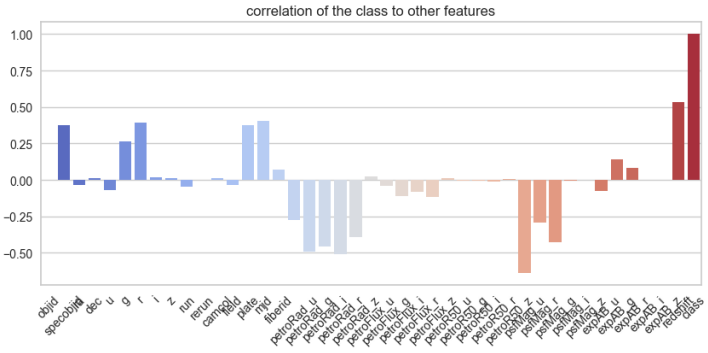

Figure 3. Correlation of different features with the 'class' column

2.4.2. Model training

To reduce the risk of overfitting, the dataset is divided into a training set, which the model uses to learn patterns and relationships, and a test set or validation set, which is kept separate for unbiased evaluation. For our model, the train set represents 70% of all the data and the evaluation set of 30%. This split is performed randomly to ensure an unbiased and generalized prediction. After careful consideration, it turned out that the optimal number of trees (n_estimators) in our model is 100. An illustration of the first decision tree in the model (index 0) is shown is figure 4. Each tree in this model uses a random subset of features making, not only, every tree unique but every random forest generated after running the program. Therefore, we could have different accuracy scores after running the code many times.

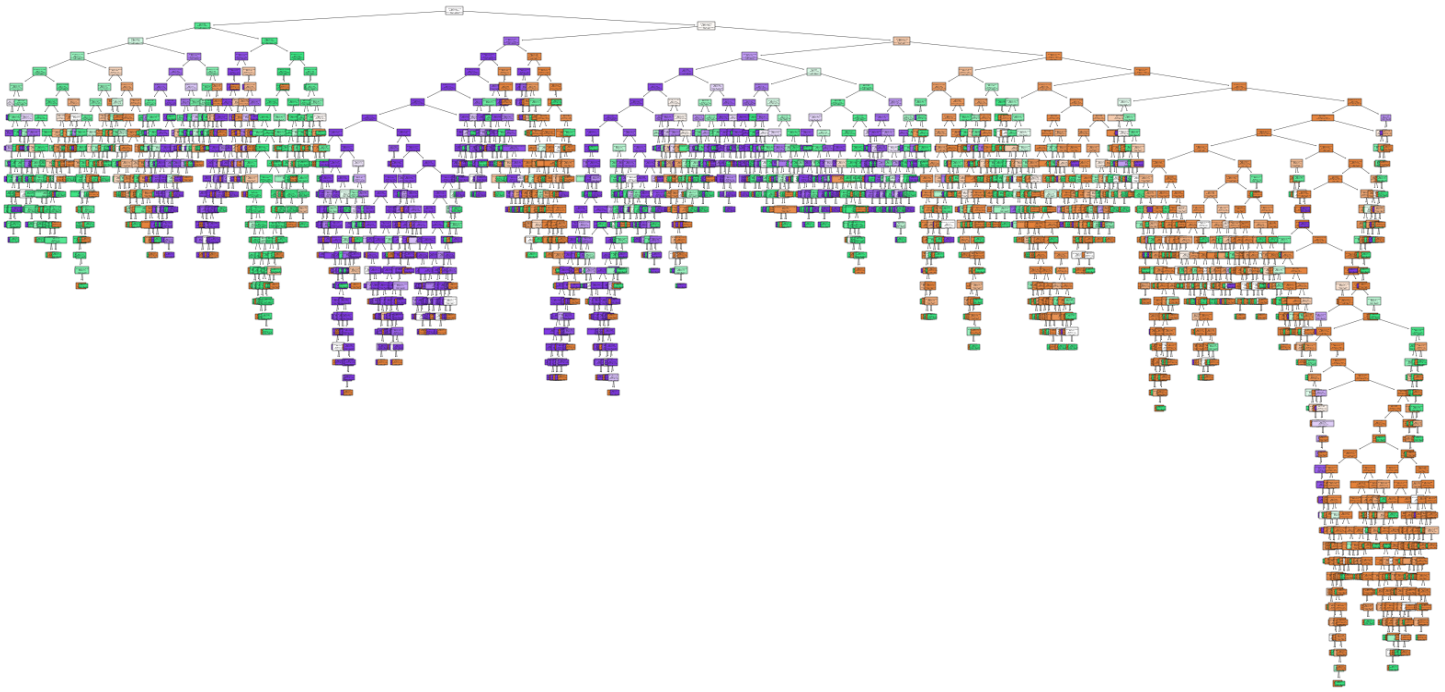

Figure 4. Visualization of the first decision tree of the Random Forest classifier

2.4.3 Model evaluation

With 100 trees, the Random Forest classifier model achieved a nearly impeccable accuracy of 99.3% in the classification of the celestial bodies.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	15630
1	1.00	1.00	1.00	15586
2	0.99	0.99	0.99	15651
accuracy			0.99	46867
macro avg	0.99	0.99	0.99	46867
weighted avg	0.99	0.99	0.99	46867

Figure 5. Classification report

As shown in figure 5, the precision, F1-score and recall for galaxies and quasars were 99%. As for stars the model attained a 100% score in precision, Recall and F1-score. The support represents the number of samples for each class in the test dataset. The classification overall achieved an average accuracy of 99% for all classes and the training process takes 32 seconds.

The scores' definitions are listed in the following equations:

$$precision = \frac{TP}{TP + FP} \quad (4)$$

$$recall = \frac{TP}{TP + FN} \quad (5)$$

$$F1\ Score = \frac{2 \times (precision \times recall)}{precision + recall} \quad (6)$$

In each of the equations above, the TP represents the true positive; FP, the false positive; and FN the false negative. The confusion matrix shown in figure 9 depicts the counts of TP, TN (true negative), FP and FN in detail giving an interesting insight into the model. The confusion matrix provides the count of the predicted labels and actual labels, by analyzing the intersections

between them, we can gain insight into the performance of our model.

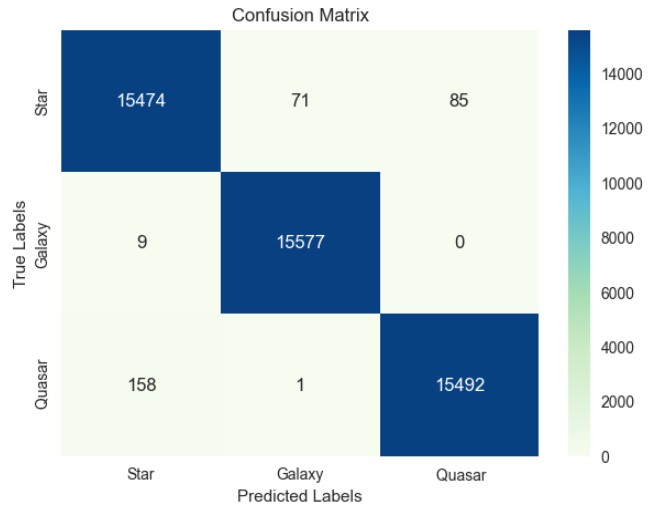

Figure 6. Confusion matrix

To check the performance of the model, another technique is using the AUC (area under the curve) ROC (Receiver Operating Characteristics) curve. It is one of the most important evaluation metrics for checking any classification model's performance. It is also written as AUROC (Area Under the Receiver Operating Characteristics). In short, the RUROC curve tells how much the model is capable of distinguishing between classes. The closer the AUC is to 1, the better the prediction is. The ROC curves for each class are represented in figure 7 [9].

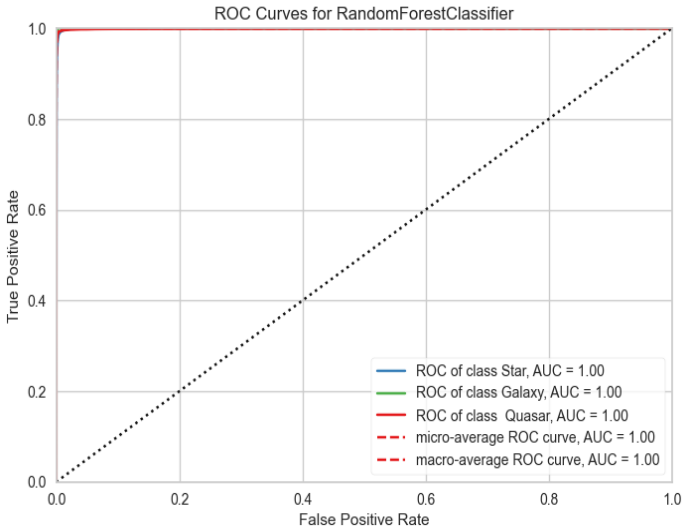

Figure 7. ROC curves and their corresponding AUC

2.5. Object localization

For the regression task, the model uses the same data (SDSS 18th data release). However, the predicted label will differ. In order to localize celestial objects in the sky, two coordinates are used: the right ascension (ra) and declination (dec). The multi-target regression algorithm will therefore predict these algorithms who represent two distinct columns of the dataset.

2.5.1. Coordinates

Like cities, every object in the sky has two numbers that fix its location called *right ascension* and *declination*, more generally referred to as the object's *celestial coordinates*. Declination corresponds to latitude and right ascension to longitude. Just like the latitude-longitude grid on earth, the cosmos is mapped as a sphere with the earth as its center The equator, which marks the 0° latitude line, circles the sky as the *celestial equator*, while the *north* and *south celestial poles* hover over either end of the planet's polar axes. Viewed from Earth's equator, the celestial equator begins at the eastern horizon, passes directly overhead and drops down to the western horizon. Since we're inside a sphere, it would continue around the backside of the Earth as well. From mid-latitudes, the celestial equator stands midway between the horizon and overhead point, while from the poles the celestial equator encircles the horizon. Anything north of the celestial equator has a northerly declination, marked with a positive sign. Anything south of the equator has a negative declination written with a negative sign. While we use a physical location on Earth as our reference for longitude, it is quite similar in the cosmic level. Astronomers use the spot the Sun arrives at on the first day of spring, called the **vernal equinox** as shown in figure 11. Presently, it's located in the constellation of Pisces, the Fish.

The sky can be treated as a clock, since it wheels by as Earth rotates, so the zero point of right ascension is called "0h" for "zero hours." Unlike longitude, right ascension is measured in just one direction east. Because there are 24 hours in a day, each hour of right ascension measured along the equator equals 1/24th of a circle (360° divided by 24) or 15°. That's a little more than one-half the width of the W-shaped constellation Cassiopeia. Unlike Earth coordinates, celestial coordinates *change* due to the slow wobble of Earth's axis called precession. Precession causes the equinox points to drift westward at a rate of 50.3 arcseconds annually. As the equinox shifts, it drags the coordinate grid with it. That's why star catalogs and software programs have to be updated regularly to the latest "epoch." This is done every 50 years. In our data, the ra and dec are referenced as the Epoch J2000.0 coordinates which stands from the year 2000 to 2050 [10].

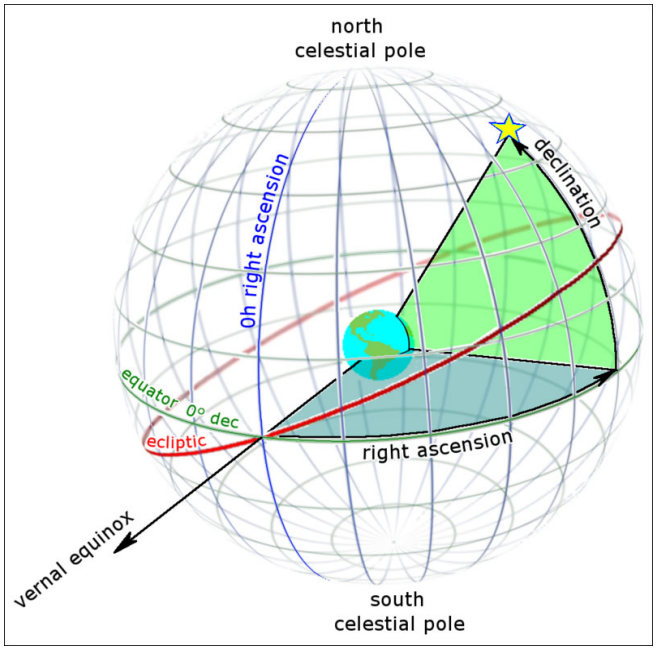

Figure 8. the cosmic sphere map. Declination (green) is measured in degrees north and south of the celestial equator. Right ascension, akin to longitude, is measured east from the equinox. The red circle is the Sun's apparent path around the sky, which defines the ecliptic [10].

2.5.2. Data preprocessing

For this regression task, the label encoding and outlier detection process will be the same since it is mainly related to the data than it is to the model. As for the feature selection, after calculating the correlation of all features with the dec and class columns, it was revealed that most features were related either negatively or positively to the coordinates. Therefore, after careful consideration, it was most safe to make use of all features during the training process knowing that the RandomForest is very resistant to that. In regression models, the target variable is continuous rather than categorical which eliminates the concept of class imbalance. Consequently, there is no need to perform oversampling.

In order to avoid potential errors or inconsistencies during model training and evaluation, the target variables (y_ra and y_dec) must be reshaped. Most machine learning libraries, including scikit-learn, expect the target variables to be in a two-dimensional array-like format. The first dimension represents the number of samples (data points), and the second dimension represents the number of target variables (output dimensions). In the context of predicting celestial coordinates, each data point represents an astronomical object, and we have two target variables (y_ra and y_dec) to predict for each object. We will use the NumPy 'reshape' function. The specific shape of the target variables after reshaping would be (num_samples, num_dimensions), where num_samples is the total number of data points, and num_dimensions is the number of target variables (2 in this case).

2.5.3. Model training

For this regression task, the study will use the random forest and the multi output regressor

since it is trying to predict the coordinates of a cosmic body which includes two variables. The data is split into training (80%) and testing (20%) datasets. During the training process, the model learns to map the features to the corresponding celestial coordinates. The model is then fed the testing dataset to evaluate its efficiency.

2.5.4. Model evaluation

For the evaluation of the regression model, several metrics will be used including the mean squared error (MSE), the mean absolute error (MAE), the relative squared error (RSE), and the R2 score (R2), the latter will be most significant in the evaluation. The R2 score is an evaluation metric that ranges from 0 to 1, the closer it is to 1, the better the model's performance . These metrics are defined by the following equations where ŷ will refer to the predicted output and y the actual output:

$$MAE = \frac{1}{n}\Sigma \left| y - \hat{y} \right| \quad (7)$$

$$MSE = \frac{1}{n}\Sigma (y - \hat{y})^2 \quad (8)$$

$$RSE = \frac{MAE}{MSE} \quad (9)$$

$$R^2 = 1 - RSE \quad (10)$$

Since we are dealing with multi target regression, each of these metrics will be calculated for the *ra* and *dec* separately.

```
Mean Absolute Error (MAE) for RA: 0.39743490657605796
Mean Absolute Error (MAE) for Dec: 0.10924619753565473
Residual Sum of Squares (RSE) for RA: 5595157668753.495
Residual Sum of Squares (RSE) for Dec: 554774501245.2527
R-squared (R^2) for RA: 0.99705350007736
R-squared (R^2) for Dec: 0.9994421009350394
```

Figure 9. Regression evaluation metrics scores

As shown in figure 9, the model performed slightly better in predicting the declination than it did with the right ascension. However, the R2 scores of ra and dec of 0.997 and 0.999 respectively show that the model has an overall excellent accuracy.

3. Discussion and conclusion

We have attempted to predict the class and coordinates of stellar objects with the Python's Random Forest algorithm. In the process, we explored the performance of our model for both classification and regression tasks, the model achieved an excellent accuracy of 99.3% and approximately 99% respectively. The data is free of null values so the preprocessing process will consist of oversampling, feature selection, label encoding and target value reshaping for the regression model. On one hand, the classification consisted of predicting the type of celestial body based on spectral features as either a galaxy, a star or a quasar. After evaluating the model, we can analyze the trends in classification. For instance, we can perceive that the model has very little confusion between predicting quasars and galaxies, this may be due to the fact that quasars are 10 to 100 times brighter than galaxies and their emission lines shift far to the red wavelength reaching up to 96% the speed of light, while galaxies have both blue and red shifts [10]. Like that, different trends in prediction are mainly a result of some vast differences in data. On the other hand, we trained our regression model to predict the right ascension and declination representing the coordinates of the stellar object. We can see that the model performed overall better in predicting the dec over the ra. Predicting ra may be a more complex task compared to predicting dec due to various factors, such as variations in the celestial sphere, seasonal changes, and different tracking systems [9]. As a result, the model might struggle more in capturing these intricate patterns accurately. This

opens the door to intriguing suggestions to improve this model.

Acknowledgments

I would like to thank Hajar Hassina for helping me structure this paper and for her guidance in this project. I also thank my mentor Salah Habachi, my advisor, for helping me find the Data I needed and assisting me in the early stages of the model building process.

References

[1] Galaxies- NASA Science universe exploration Basics | Galaxies – NASA Universe Exploration.
[2] Stars - NASA Science universe exploration Stars | Science Mission Directorate (nasa.gov)
[3] IBM - What is machine learning What is Machine Learning? | IBM
[4] MIT Management Sloan School - Machine learning explained. Machine learning, explained | MIT Sloan
[5] IBM - What is a Random Forest. What is Random Forest? | IBM
[6] SDSS - SDSS-V, pioneering panoptic spectroscopy Sloan Digital Sky Survey-V: Pioneering Panoptic Spectroscopy - SDSS-V
[7] Vaibhab Jayaswal - tds- Local Outlier Factor (LOF) — Algorithm for outlier identification Local Outlier Factor (LOF) — Algorithm for outlier identification | by Vaibhav Jayaswal | Towards Data Science
[8] Jason Brownlee - SMOTE for imbalance classification with python SMOTE for Imbalanced Classification with Python - MachineLearningMastery.com
[9] Sarang Narkhede - tds - understanding ROC-AUC curves Understanding AUC - ROC Curve | by Sarang Narkhede | Towards Data Science
[10] Sky & Telescope - Right ascension and declination: celestial coordinates for beginners. Celestial Coordinates for Beginners - Sky & Telescope - Sky & Telescope (skyandtelescope.org)
[11] Bartleby - describe some differences between Quasars and normal galaxies. Describe some differences between quasars and normal galaxies. | bartleby

Appendix

Here is a section of the classification and regression training algorithms:

classification:

```
x_train, x_test, y_train, y_test =
train_test_split(x, y, test_size = 0.3, random_state
= 42)
r_forest = RandomForestClassifier()
r_forest.fit(x_train,y_train)
predicted = r_forest.predict(x_test)
```

Regression:

```
y_ra = y_ra.values.reshape(-1, 1)
y_dec = y_dec.values.reshape(-1, 1)

X_train, X_test, y_ra_train, y_ra_test,
y_dec_train, y_dec_test = train_test_split(X,
y_ra, y_dec, test_size=0.2, random_state=42)
rf_regressor =
RandomForestRegressor(random_state=42)
multioutput_regressor =
MultiOutputRegressor(rf_regressor)
multioutput_regressor.fit(X_train,
np.hstack((y_ra_train, y_dec_train)))

predicted_ra =
multioutput_regressor.predict(X_test)[:, 0]
predicted_dec =
multioutput_regressor.predict(X_test)[:, 1]
```