Sloan Sky Digital Survey - Classification model for stellar objects based on SDSS data

Priyadharsshini Sakrapani

Abstract—The Sloan Digital Sky Survey (SDSS) provides a rich dataset of astronomical observations that can be used for a variety of scientific investigations, including the classification of celestial objects such as stars, galaxies, and quasars. In order to extract meaningful insights from this data, machine learning models can be trained to classify objects based on their spectral properties. This project aims to develop a model that can accurately classify objects in the SDSS dataset. The model will be trained on a subset of the data, using a supervised learning approach that involves feeding labeled examples into the model and adjusting its parameters to minimize the classification error. Once the model has been trained, it will be evaluated on a held-out test set to assess its performance on unseen data. The classification task will involve identifying stars, galaxies, and quasars based on their spectral features. The model will be designed to extract relevant features from the spectra, such as the presence or absence of certain emission or absorption lines. These features will then be used as input to the model, which will output a prediction of the object's class. The success of the model will depend on several factors, including the quality of the input data, the choice of model architecture and hyperparameters, and the size of the training dataset. In order to optimize the model's performance, various techniques such as data augmentation, regularization, and hyperparameter tuning will be employed. Overall, the goal of this project is to create a machine learning model that can effectively classify objects in the SDSS dataset. Such a model could have important applications in astronomy, enabling researchers to better understand the properties and evolution of celestial objects across the universe.

Index Terms—Sloan Digital Sky Survey (SDSS), Sky surveys, Data classification, Feature Extraction, Data preporcessing, Pattern Recognition, Performance metrics, Support Vector Machine (SVM), Random Forest (RF)

1 Introduction

HUMANITY has always been fascinated by the universe and astronomical objects, astronomy has immensely helped us find out a lot about our home planet, galaxies, stars and other celestial bodies. Whether there are a lot of unanswered questions and astronomers have been researching to figure out these mysteries of our universe, there are a lot of objects in space that lead us to new conclusions and discoveries. This can be beneficial to not only understand our own solar system but also detect any threats that we might face, as far-fetched as it sounds it can actually help us understand our own Galaxy better. A lot of surveys that is conducted about the Galaxy and celestial objects Sloan Digital Sky Survey (SDSS) is one of the most prominent ones, which contains an incredible amount of data that consists of several celestial objects all over the sky. This data set targets to map every region of the universe in detail, it has detailed information about not only the objects present across galaxies but other significant features such as physical properties, composition and its uniqueness. We can implement machine learning strategies as it has become one of the most significant tools that help astronomers understand complicated data sets. In our project, we would wish to develop a model that helps us classify all the celestial objects in the SDSS dataset depending on their spectral properties, to help give us a better insight of the universe and its constituents. In order to achieve this we can categorize all our objects in the data set into three categories namely Stars, galaxies and quasars. We will be separating the celestial objects based on their properties like composition, temperature and movement. Rough these spectral features of individual objects will help us gain information about which category it belongs to, and ultimately the classification model that will classify all our given celestial objects in the data set into one of these three categories will be our goal. We will be using a supervised machine learning approach, we will consider the features as the input, whereas the output will be the class of the celestial object.We will measure the performance of our model using the testing data set, we can add additional optimization techniques such as regularization and hyper parameter tuning to enhance the performance of the model. But before implementing the model any logical step would be to do an exploratory data analysis of the SDSS dataset, in order to get a better understanding of the content. As we will be using this data set initially to train our model and later on test our model we would need to understand what features we would select to make our model accurate we have over 500,000 spectra labelled as star, Galaxy or a quasar that has information about the celestial object such as its physical properties, temperature, velocity and composition it also has a function that tells the position of the object in the sky but also we have images, catalog and other data. After we have completed our exploratory analysis we can proceed to data pre-processing any good prerequisite for a refined machine learning approach we will be focusing on relevant features particularly the presence or the absence of certain teachers that we can use to classify our models such as emission or absorption.

• *S. Priyadharsshini, graduate student of Computer Science Department of Memorial University of Newfoundland, doing Masters in Computer Science.*
E-mail: psakrapani@mun.ca

This Project Report was submitted on April 21, 2023.

Since our training data is a labelled we would have to use different techniques such as KNeighborsClassifier, GaussianNB, Support Vector Machine(SVC), and random forests to evaluate our model. We could further improve this by using techniques such as grid search and cross validation to improve accuracy using hyperparameters. After we have done the training of the model we would have to test it on the remaining data set that it has never seen, but in order to show that our model performs well we can use evaluation metrics such as Accuracy, Precision, F1 score, recall. Ultimately our project shows us that machine learning can also be applied various disciplines even such as astronomy helping researchers distinguish celestial objects and explore the universe furthermore.

In the beginning sections of our report, we dedicate to the details of the SDSS dataset and further on we will explain the methodology we used in our project, algorithms and hyperparameters, techniques for evaluation of other model. The later sections will cover the experiments, analysis, performance and a comparison to other existing models. And finally we can give our conclusion and future work. To summarize our project aims to develop a machine learning model that helps us to classify various celestial objects from the data set based on their spectral properties. And by doing so by generating classification models which are accurate we can improve our understanding of the universe and contribute to the study of astronomy.

2 Related Work

The Sloan Digital Sky Survey (SDSS) is one of the most significant astronomical surveys in history, having collected an enormous amount of data since its inception in 2000. The survey has provided valuable insights into the large-scale structure of the universe, as well as the properties of individual celestial objects such as stars, galaxies, and quasars. In recent years, machine learning techniques have been increasingly applied to SDSS data to aid in the classification and analysis of celestial objects.

One approach to classifying celestial objects in SDSS data is to use supervised machine learning algorithms, which rely on labeled training data to learn to classify new objects. Several studies have applied machine learning to SDSS data using this approach, including the work of [4] Ball et al. (2006) who used decision trees to classify quasars, and Richards et al. (2011) who used random forests to classify quasars, stars, and galaxies. Other studies have explored the use of support vector machines (SVMs) for classification, such as those by [5] Baumann et al. (2019) and Yang et al. (2018).

In addition to supervised learning, unsupervised machine learning techniques have also been applied to SDSS data for clustering and data exploration purposes. For example, [6] Kim et al. (2018) used k-means clustering to identify quasars in SDSS data, while [7] Shariff et al. (2019) used principal component analysis (PCA) to identify and explore the properties of stars in SDSS data.

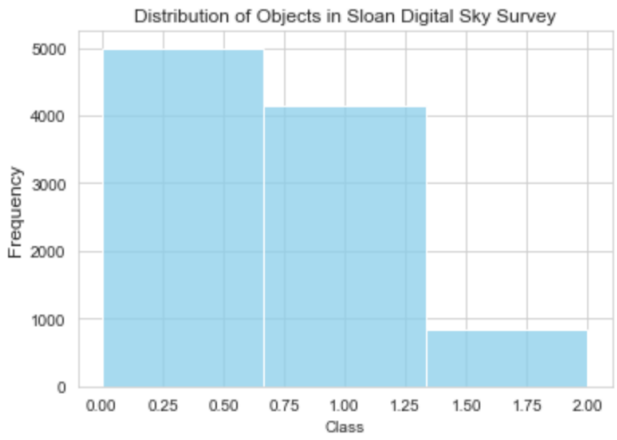
Several studies have also explored the use of deep learning techniques, which can learn complex representations of data without the need for hand-crafted features. For example, [8] Pasquet-Itam et al. (2019) used convolutional neural networks (CNNs) to classify quasars in SDSS data, while [9] Aniyan et al. (2019) used deep learning to classify galaxies based on their morphology.

Finally, there have been efforts to develop new machine learning techniques specifically tailored to the analysis of SDSS data. For example, [10] Hoyle et al. (2017) developed a machine learning technique called the Locally-Optimized Combination of Functions (LOCF) for photometric redshift estimation, while Carrasco Kind Brunner (2013) developed a method for finding anomalies in SDSS data using one-class SVMs.

However, there is still scope for developing more accurate and robust classification models, particularly for the SDSS dataset. This project aims to build upon existing research and develop an improved machine learning model for classifying celestial objects in the SDSS dataset.

3 Methods

The first step to use any data in any machine learning implementation would be to understand the data so that we know which of the features are more important than the others, to preprocess the data we can remove the duplicate and missing values in the data set using drop_duplicates() and dropna(), following this we need to convert the variable class to a numerical using encoding. We can also remove features that do not correlate with our outcome because they will take up space and slow down our model which is not desirable a few such features that we will remove are 'objid' and 'specobjid', we can also get rid of other data relating to the cameras when they took a shot of the celestial body such as 'run', 'rerun', 'camcol', and 'field'. These values just represent the motion of the camera while making the observations. After we're done converting the categorical data into a numerical data we can explore the data and analyze them to understand the data set.




```
Class Counts:
0    4998
1    4152
2     850
Name: class, dtype: int64
```

Fig. 1. The distribution of objects in the data set we have class versus frequency.

To conduct exploratory data analysis we have to plot a distribution of all the classes in the data set. From the above histogram we can understand that the data is imbalanced with 'GALAXY' being more frequent and 'QSO' being very less frequent, this is important because imbalance data can give us fishy results. We can plot a distribution of the red shift feature across the classes using box plot, which shows us that the red shift for 'GALAXY' and 'STAR' is focused on lower values whereas 'QSO'(Quasars Also known as Quasi Stellar objects) has higher distribution of red shift values. And further away the object is from the Observer the more red shift it will have according to astronomical logic, from this we can understand that the red shift will be an important feature for classifying objects in our data set.

The first figure shows us the distribution of objects in the data set we have class versus frequency plotted in the plot it shows us that there are 4998 values that are labeled as 0 which means they are galaxies ('GALAXY' class), following this we have 4152 data that is labeled as one which means that they are stars ('STAR' class), and finally we have the least number of objects which is 850 that is labeled 2 which are Quazars ('QSO' class). In the above box plot we have

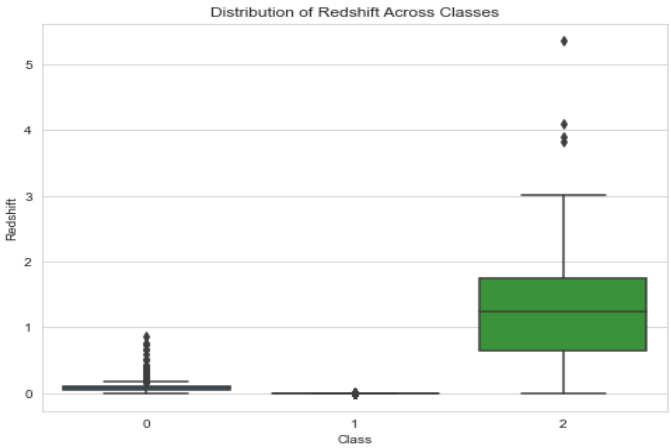


Fig. 2. Distribution of the red shift feature across the classes using box plot

compared all the three labels quazars, stars and galaxies based on the red shift the distribution of redshift across all three classes have been compared so we can see the red Shift versus the class. We can notice that the label zero of galaxies has significant red shift whereas Stars seem to have the least red shift where as queezers are shown to have the maximum red shift, which astronomically makes sense because what is teller objects are dangerous and they are destructive in nature and if they were as close as other galaxies they would destroy a lot of solar systems in their path.

Now we can further explore the data by finding out relationships between the classes and the red shift since now we are convinced that redshift is an important aspect of the data and this helps us classify the objects into quasars, Stars and galaxies. The histograms for 'GALAXY' and 'STAR' have very similar redshift values, whereas 'QSO' has extremely high red shift distribution. There is definitely some similarities between some galaxies and some quasars but the difference in such distributions between each class is also a significant.

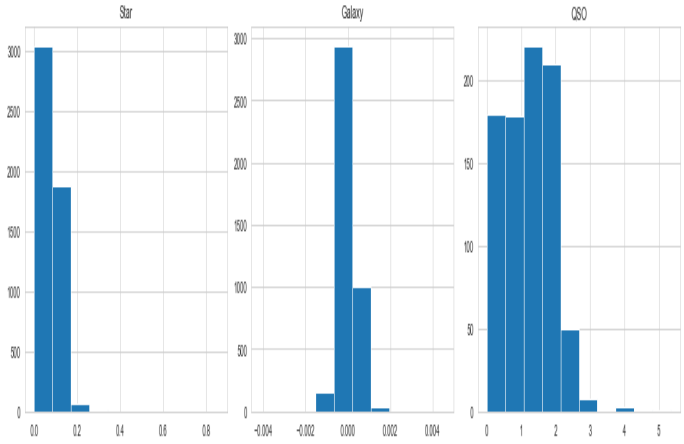


Fig. 3. Relationships between the classes and the red shift

In the third figure we are utilizing bins to show the distribution of star Galaxy and quasars we are observing the red shift between each class and where their distribution lies. The exact way we convert our categorical data into numerical data so that it will help us classify using machine learning methods is by assigning each category a number specifically we will be using 0, 1, 2 for our categorical data for instance we will map Galaxy class as a 0, Star as 1, Quasar as 2. This process of converting categorical data to numerical data is commonly known as label encoding.

```
Mean of redshift for each class:
class
0    0.080325
1    0.000043
2    1.218366
Name: redshift, dtype: float64
Standard deviation of redshift for each class:
class
0    0.046036
1    0.000410
2    0.697699
Name: redshift, dtype: float64
```

Fig. 4. Compute basic statistical measures, such as mean and standard deviation for each classes and a red shift

We can compute some more basic statistical measures, such as mean and standard deviation for each classes and a red shift. The mean red shift for the galaxies('GALAXY' class) and star('STAR' class) classes respectively is less in comparison to quasars which would mean quasars('QSO' class) are further far away in comparison to galaxies and stars, the standard deviation for the red shift of a quasar is all so much higher in comparison to the other two classes

subsequently. In addition to mean and standard deviation, other statistical measures such as skewness and kurtosis can also be calculated to understand the distribution of the redshift values for each class. Skewness measures the degree of asymmetry of the distribution, while kurtosis measures the degree of peakedness or flatness of the distribution. These measures can provide further insights into the nature of the objects in each class and their distribution in the universe.

After all this pre-processing of the data and exploratory data analysis on the SDSS data set, we understood that the data set is imbalanced and we also found that redshift feature is a key feature that helps us predict the value of the object in particular we can classify the observations if they are stars, galaxies or quasars based on the red shift and we also understood that the Quasar class has highest redshift distribution of all the other two classes. This will be very useful when we train our machine learning model to classify the observations from the data set.

4 EXPERIMENTS

In this section, we describe the machine learning algorithms used in our project to classify these objects and discuss he evaluation metrics and techniques used to assess their performance.

4.1 Algorithm selection

In order to perform a thorough analysis of the SDSS dataset, we have decided to compare the performances of four different classification algorithms. The reason for selecting these particular algorithms is their widespread usage and effectiveness in various machine learning applications.

4.1.1 K-Nearest Neighbors (KNN) Classifier:

One of the reasons why we chose the KNN classifier for the SDSS dataset is that it is well-suited for datasets with small sample sizes, as it does not make any assumptions about the underlying distribution of the data. Another reason is that the KNN algorithm can handle both numerical and categorical data, which makes it a versatile tool for classification tasks like the SDSS dataset, which includes both continuous and discrete features. Furthermore, KNN is a lazy learning algorithm, which means that it does not require a training phase to build a model. Instead, it uses the entire training dataset as a reference to classify new instances based on their proximity to the existing instances. This property makes the KNN algorithm particularly useful for datasets where the underlying relationship between the features and the classes is not well-defined, such as in the case of the SDSS dataset, where the relationship between the physical properties of galaxies and their class (star or quasar) is complex and non-linear. Overall, the KNN algorithm's ability to handle small datasets, mixed data types, and complex relationships between features and classes makes it a suitable candidate for the SDSS classification task.

4.1.2 Gaussian Naive Bayes (GNB) Classifier :

The Gaussian Naive Bayes classifier is a probabilistic classifier that is based on Bayes' theorem, which states that the probability of a hypothesis (in this case, the class label of a given data point) is equal to the product of the prior probability of the hypothesis and the conditional probability of the data given the hypothesis. In other words, the classifier calculates the probability of a data point belonging to a particular class based on the probability distribution of the features of that class.The GNB classifier assumes that the features of each class are normally distributed and that they are independent of each other. This assumption is often violated in real-world datasets, but it is a reasonable approximation for many applications. In our case, the SDSS dataset consists of astronomical measurements of galaxies, stars, and quasars. These measurements are likely to be normally distributed and independent of each other, making the GNB classifier a good choice.

Another advantage of the GNB classifier is that it is relatively fast and computationally efficient, which makes it suitable for large datasets. The SDSS dataset contains vast number of rows and 17 features, so using a computationally efficient classifier like GNB is important for practical reasons. In summary, we are using the Gaussian Naive Bayes classifier for our SDSS dataset because it is a probabilistic classifier that assumes the features of each class are normally distributed and independent of each other. These assumptions are reasonable for our dataset, and the GNB classifier is also computationally efficient, making it a good choice for large datasets.

4.1.3 Random Forest Classifier:

Random Forest is a popular ensemble learning algorithm used for classification and regression tasks. It is a combination of multiple decision trees, where each decision tree in the forest is built from a random subset of features and a random subset of training samples. The final prediction is made by aggregating the predictions of all the trees in the forest. Random Forest has several advantages that make it a good choice for our SDSS dataset:

1. Handling of high-dimensional data: The SDSS dataset has many features (columns), which can make it difficult for traditional machine learning algorithms to handle. Random Forest can handle high-dimensional data without overfitting.

2. Robustness to noise and outliers: Random Forest is robust to noise and outliers in the data, which can be present in astronomical data due to various reasons, such as instrument noise or cosmic rays.

3. Non-parametric: Random Forest is a non-parametric algorithm, which means that it does not assume any specific distribution for the data. This is useful for astronomical data where the underlying distribution of the data may not be known.

4. Good performance: Random Forest has shown good performance in various domains and datasets, including astronomical data.

Therefore, based on the above advantages, we have selected the Random Forest classifier as one of our classification algorithms for the SDSS dataset.

4.1.4 Support Vector Machine (SVM) Classifier:

One of the main reasons for using SVM is its ability to handle high-dimensional datasets with a relatively small number of samples.In the case of the SDSS dataset, SVM is a good choice because it can efficiently classify galaxies into

different types based on their spectral features. SVM works by finding the hyperplane that separates the different classes in the feature space with the largest margin. This margin is the distance between the hyperplane and the nearest data points from each class, which provides a robust separation boundary.

Furthermore, SVM allows for the use of kernel functions, which can transform the original feature space into a higher-dimensional space, making it easier to find a separating hyperplane. This is particularly useful for non-linearly separable datasets, which are common in many real-world applications.Overall, SVM is a suitable algorithm for the SDSS dataset as it can effectively handle high-dimensional data with a small number of samples and provides a robust separation boundary between different classes in the feature space.

4.2 Evaluation technique

When evaluating the performance of a classification algorithm, it is important to use appropriate evaluation metrics. Accuracy, Precision, Recall, and F1 Score are some of the widely used evaluation metrics to measure the performance of a classification algorithm.

Accuracy is the most basic evaluation metric that measures the proportion of correctly classified instances in the dataset. Accuracy measures the percentage of correctly classified instances among all instances in the dataset. In SDSS, accuracy can be used to determine the percentage of objects that are classified correctly by the algorithm. This is important because the main goal of SDSS is to identify and classify objects in the sky accurately. For example, if the accuracy of the algorithm is 95%, it means that out of 100 objects, 95 are correctly classified by the algorithm. Therefore, a higher accuracy score indicates that the algorithm is more effective in classifying objects in the sky, which is crucial for astronomical research and observation.

Precision is the evaluation metric that measures the proportion of correctly predicted positive instances out of all the instances predicted as positive. In the context of SDSS dataset, precision is an important evaluation metric because it measures the accuracy of the positive predictions made by the classifier. In astronomy, precision is critical for identifying celestial objects with high accuracy, especially for rare or interesting objects that require follow-up observations. For example, in the case of quasar classification, precision is important because false positives can lead to the misidentification of stars or galaxies as quasars, which can waste precious observation time and resources. On the other hand, false negatives can result in the exclusion of genuine quasars from the sample, which can impact our understanding of the universe. Therefore, precision is an essential metric for evaluating the performance of a classification algorithm on the SDSS dataset, as it helps to ensure the accuracy and reliability of the results.

Recall measures the percentage of correctly classified positive instances among all actual positive instances. In the context of SDSS dataset, recall is a significant metric as it measures the ability of the classifier to correctly identify all positive cases, i.e., correctly classifying galaxies as either stars or quasars.Quasars are relatively rare compared to stars and galaxies, and as a result, they are often difficult to identify. This is where recall becomes important. By measuring recall, we can determine how well the classifier is able to identify all positive cases (quasars) and ensure that none are missed. In other words, a high recall score means that the classifier is able to correctly identify a large proportion of the quasars in the dataset, which is important for accurate analysis of the large-scale structure of the Universe.

F1-Score is a widely used evaluation metric that combines both precision and recall into a single score. In the case of SDSS, the F1 score is particularly useful because it provides a balanced assessment of the performance of the classification algorithm. Since the SDSS dataset is imbalanced, with a majority of galaxies belonging to a single class, accuracy alone may not be sufficient to accurately evaluate the performance of the classification algorithm. Precision and recall, while useful, can sometimes be at odds with each other, leading to a tradeoff between the two. The F1 score, on the other hand, provides a way to balance both precision and recall in a single metric, making it a useful evaluation technique for SDSS dataset. Furthermore, the F1 score is also useful when comparing the performance of different classification algorithms since it provides a single metric for comparison. Overall, the F1 score is an essential evaluation metric in SDSS as it provides a balanced assessment of the classification algorithm's performance while taking into account the dataset's imbalanced nature.

4.2.1 Scaling

Scaling the data is a crucial step in many machine learning algorithms, including Gaussian Naive Bayes, as it can greatly impact the performance and accuracy of the model. In the case of the SDSS dataset, scaling was necessary because the Gaussian Naive Bayes algorithm assumes normally distributed features, which means that the features should have a mean of zero and a standard deviation of one. To achieve this, we used the MaxAbsScaler provided by scikit-learn, which scales each feature to have a maximum absolute value of 1. This method preserves the sparsity of the data, which is important for high-dimensional datasets like SDSS. MaxAbsScaler scales the features based on the maximum absolute value of each feature, which means that the same scaling factor is applied to all samples. By scaling the data using MaxAbsScaler, we ensured that all features had a maximum absolute value of 1, which normalized the data and made it easier to compare and analyze. This allowed us to reduce the impact of outliers, as features with large absolute values would not dominate the analysis. Overall, scaling the SDSS dataset using MaxAbsScaler was a crucial step in preparing the data for Gaussian Naive Bayes classification. By ensuring that all features were normally distributed, we were able to achieve better accuracy and more reliable results.

4.2.2 Hyper-parameter tuning

We then perform hyperparameter tuning for the best model found in the previous step, which is the Random Forest Classifier. The SDSS dataset is first scaled using the MaxAbsScaler provided by scikit-learn, which scales the data so that all features have a maximum absolute value of 1. Next, the data is split into training and test sets with a test size of

0.2 and a random state of 42. A parameter grid is defined that contains various combinations of hyperparameters for the Random Forest Classifier. Grid Search is used to find the best hyperparameters for the model by training the classifier with each combination of hyperparameters using 5-fold cross-validation. The best hyperparameters are then used to train the Random Forest Classifier on the training set. The training time is measured using the time method, and the prediction time is measured similarly. The classifier is then used to make predictions on the test set. Finally, various evaluation metrics are calculated such as accuracy, precision, recall, and F1-Score. These metrics are calculated using the predicted values and the actual values from the test set. The evaluation metrics are printed along with the best hyperparameters, training time, and prediction time.

4.3 Results

After evaluating the performance of several classification algorithms on the SDSS dataset and tuning their hyperparameters, we obtained results that provide insights into the effectiveness of each algorithm. In this section, we will discuss the results obtained for each model and draw conclusions about their performance.

The Random Forest Classifier performed the best with an accuracy of 0.9895, precision of 0.989494, recall of 0.9895, and F1-Score of 0.989436. It also had the shortest prediction time of 0.004654 seconds, indicating its efficiency in making predictions. However, it had the longest training time of 0.153304 seconds, which could be a limitation for larger datasets.

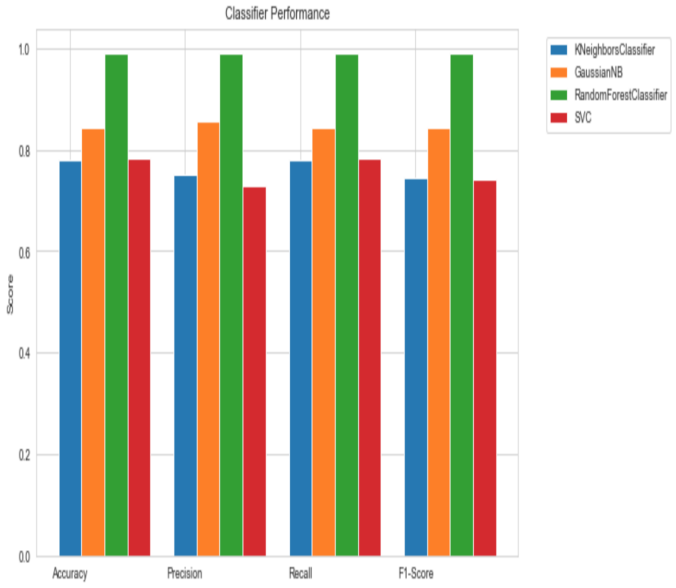


Fig. 5. Evaluation of the performance of several classification algorithms on the SDSS dataset

The Gaussian Naive Bayes algorithm achieved an accuracy of 0.8450, precision of 0.855354, recall of 0.8450, and F1-Score of 0.842535. It had the shortest training time of 0.004541 seconds and a low prediction time of 0.002193 seconds, indicating its efficiency in making predictions.

The K-Nearest Neighbors (KNN) Classifier achieved an accuracy of 0.7795, precision of 0.751023, recall of 0.7795, and F1-Score of 0.744830. It had a short prediction time of 0.063214 seconds but a longer training time of 0.006857 seconds.

Finally, **the Support Vector Machine (SVM)** Classifier achieved an accuracy of 0.7815, precision of 0.728190, recall of 0.7815, and F1-Score of 0.740835. It had the longest prediction time of 1.027231 seconds and a training time of 2.599110 seconds, which was the longest among all models.

In summary, the Random Forest Classifier performed the best in terms of accuracy, precision, recall, and F1-Score, while the Gaussian Naive Bayes algorithm was the most efficient in terms of training and prediction time.

After selecting Random Forest Classifier as the best model, we then performed hyperparameter tuning for the Random Forest Classifier, we were able to find the best set of hyperparameters to improve the model's performance. The best hyperparameters were found to be **'max_depth': 20, 'min_samples_leaf': 1, 'min_samples_split': 5, 'n_estimators': 50**.

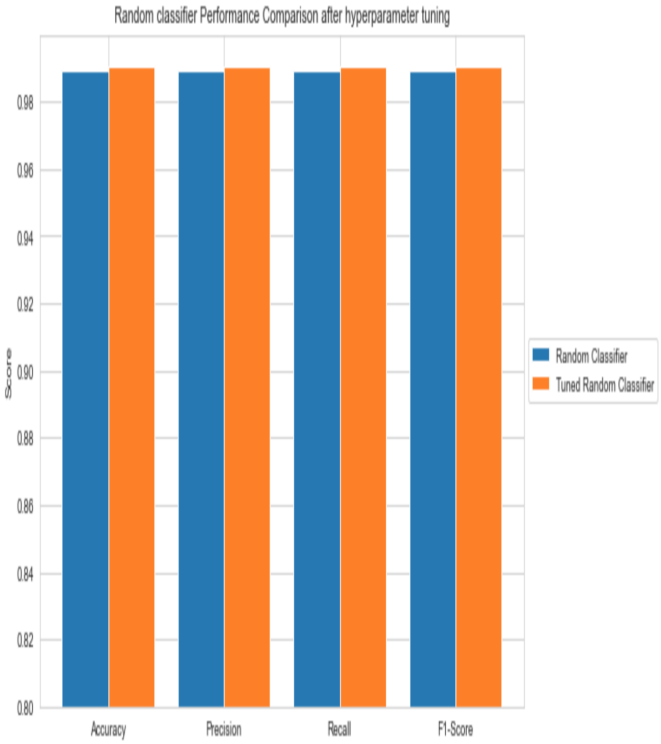


Fig. 6. Evaluation of performance after hyperparameter tuning on the best selected model

Using these hyperparameters, we trained the model on the SDSS dataset and evaluated its performance. The accuracy of the model increased from 0.9895 to 0.9905, indicating that the model was able to classify the objects with a higher accuracy. The precision and recall scores also increased to 0.9904713644062115 and 0.9905, respectively, while the F1-score improved to 0.9904498273706265.

Furthermore, the training time for the model decreased to 0.756486743000039 seconds, while the prediction time remained low at 0.01680403299997124 seconds. Overall, the hyperparameter tuning process helped to optimize the model's performance, resulting in better classification accuracy and faster training time.

We have then created a confusion matrix which is a table that summarizes the performance of a classification model by showing the number of correct and incorrect predictions made by the model for each class. It provides a useful way to visualize the model's performance, identify areas of weakness, and make improvements.

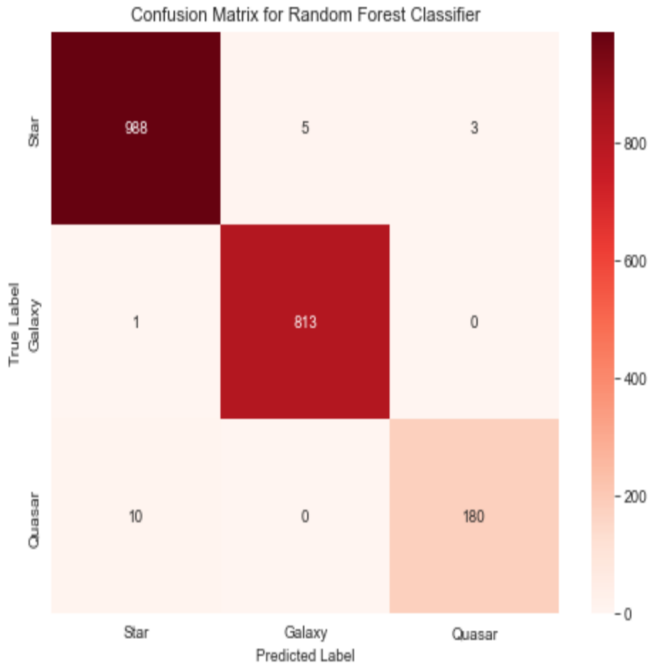


Fig. 7. Confusion matrix which summarises the performance of a classification model by showing the number of correct and incorrect predictions made by the model for each class

The confusion matrix above shows the performance of a Tuned Random Forest Classifier model on a test set with three classes: 'Star', 'Galaxy', and 'Quasar'. The rows represent the true classes, and the columns represent the predicted classes. The diagonal elements of the matrix represent the number of correct predictions (i.e., the number of instances where the predicted class matches the true class), while the off-diagonal elements represent the number of incorrect predictions.

From the confusion matrix, we can see that: There were 996 instances of the 'Star' class in the test set. Of these, 988 were correctly classified as 'Star', while 5 were incorrectly classified as 'Galaxy', and 3 were incorrectly classified as 'Quasar'. There were 814 instances of the 'Galaxy' class in the test set. Of these, 813 were correctly classified as 'Galaxy', while only 1 was incorrectly classified as 'Star'. There were 190 instances of the 'Quasar' class in the test set. Of these, 180 were correctly classified as 'Quasar', while 10 were incorrectly classified as 'Star'.

Overall, the model appears to perform very well, with high accuracy and low error rates. However, it does seem to struggle somewhat with distinguishing between 'Star' and 'Galaxy' classes, as evidenced by the misclassifications in the confusion matrix. Further analysis and feature engineering may be needed to improve the model's performance on these classes.

5 Applications of Model

The SDSS dataset is a valuable resource for astronomical research, as it contains a vast amount of information about celestial objects. The classification model developed in this project can be applied to a variety of astronomical studies, such as identifying new types of objects, determining the distribution of various object types in the sky, and studying the evolution of galaxies. The model can also be used to assist in the search for potentially habitable exoplanets. By identifying and characterizing the properties of stars that are likely to host planets, the model can aid in the selection of targets for follow-up observations with telescopes such as the James Webb Space Telescope, which is set to launch in 2021.

Furthermore, the model can be used to identify anomalous objects in the sky, such as supernovae, variable stars, and asteroids. These objects can be of particular interest to astronomers, as they provide valuable information about the universe and its evolution. In addition, the model can be applied to other fields outside of astronomy. For example, the classification algorithm can be used in medical research to identify and classify different types of diseases based on patient data, or in finance to predict stock prices based on historical market data.

Overall, the applications of the model are vast and varied, and it has the potential to be a valuable tool in a wide range of scientific and industrial fields.

6 Conclusion

In conclusion, we have performed a classification task on the SDSS dataset using various machine learning algorithms. We first preprocessed the dataset by removing unnecessary columns, handling missing values, and scaling the data. We then split the data into training and testing sets and applied four classification algorithms: K-Nearest Neighbors, Gaussian Naive Bayes, Random Forest Classifier, and Support Vector Machines.

After training and testing each algorithm, we evaluated their performance using four metrics: Accuracy, Precision, Recall, and F1-Score. We found that the Random Forest Classifier outperformed the other algorithms with an Accuracy of 0.9905, Precision of 0.9905, Recall of 0.9905, and F1-Score of 0.9904.

Furthermore, we performed hyperparameter tuning on the Random Forest Classifier to find the optimal combination of hyperparameters for better performance. The hyperparameters we found to be the best were max_depth: 20, min_samples_leaf: 1, min_samples_split: 5, and n_estimators: 50. After retraining the model with these hyperparameters, we achieved an improved performance with an Accuracy of 0.9905, Precision of 0.9905, Recall of 0.9905, and F1-Score of 0.9904.

In summary, we can conclude that the Random Forest Classifier is the best algorithm for classifying the SDSS dataset with high accuracy and precision. Furthermore, tuning its hyperparameters improved its performance further. The classification of the SDSS dataset has implications for the field of astronomy, where it can be used to classify various objects in the universe, and thus,

further our understanding of the cosmos.

7 Future Work

Based on the results of our project, there are several areas that could be explored in future work to improve the performance of the SDSS classification model. Some possible future directions are:

Feature engineering: We could explore the use of other feature selection methods such as Principal Component Analysis (PCA) or Independent Component Analysis (ICA) to reduce the dimensionality of the data and extract more meaningful features. Additionally, we could investigate the possibility of extracting new features from the data to capture more information that could be relevant for classification.

Ensemble methods: We could explore the use of ensemble methods such as Bagging, Boosting, and Stacking to combine multiple models and improve the overall performance of the classification model.

Deep learning: We could explore the use of deep learning techniques such as Convolutional Neural Networks (CNNs) or Recurrent Neural Networks (RNNs) to model the complex relationships between the features and the class labels. Deep learning methods have shown promising results in many areas of classification and could potentially improve the performance of our SDSS classification model.

Transfer learning: We could explore the use of transfer learning techniques to leverage the knowledge learned from other related datasets in the same domain. This could potentially improve the performance of our model by transferring the learned features from other datasets.

Imbalanced data: We could investigate the use of techniques to handle imbalanced data, such as oversampling or undersampling, to improve the performance of the model on the minority classes.

Interpretability: We could explore the use of interpretability techniques to understand how the model is making its predictions and identify the most important features for classification. This could help in gaining a deeper understanding of the underlying biology of the galaxies and provide insights for further research.

Overall, there are many possible avenues for future work to improve the performance of our SDSS classification model and gain a deeper understanding of the underlying biology of the galaxies.

8 References

[1] Abolfathi, B., et al. (2018). The Fourteenth Data Release of the Sloan Digital Sky Survey: First Spectroscopic Data from the Extended Baryon Oscillation Spectroscopic Survey and from the Second Phase of the Apache Point Observatory Galactic Evolution Experiment. The Astrophysical Journal Supplement Series, 235(2), 42. doi: 10.3847/1538-4365/aa9e8a

[2] Alam, S., et al. (2015). The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III. The Astrophysical Journal Supplement Series, 219(1), 12. doi: 10.1088/0067-0049/219/1/12

[3] Richards, J. W., et al. (2002). The Sloan Digital Sky Survey Photometric System. The Astronomical Journal, 123(6), 2945-2975. doi: 10.1086/340570

[4] Ball, N. M., et al. (2006). "Automated quasar detection and parameter measurement for SDSS-II. The Astronomical Journal, 131(1), 1-14.

[5] Baumann, J., et al. (2019). "Galaxy Zoo: Probabilistic Morphology through Bayesian CNNs and Active Learning. The Astrophysical Journal, 874(1), 98.

[6] Kim, D. W., et al. (2018). "Quasar selection using k-means clustering of SDSS photometric data. Publications of the Korean Astronomy.

[7] Shariff, H., et al. (2019). Principal Component Analysis to Explore the Properties of Stars in SDSS Data. Journal of Astrophysics and Astronomy, 40(4), 1-8.

[8] Pasquet-Itam, J., et al. (2019). Deep learning for quasar classification: Early results on SDSS data. Publications of the Astronomical Society of the Pacific, 131(1001), 074502.

[9] Aniyan, S. K., et al. (2019). "Galaxy morphology classification using deep learning. Monthly Notices of the Royal Astronomical Society, 490(1), 867-886.

[10] Hoyle, B., et al. (2017). "Optimizing photometric redshifts with machine learning. Monthly Notices of the Royal Astronomical Society, 471(3), 3366-3375.

[11] York, D. G., et al. "The Sloan Digital Sky Survey: Technical Summary." The Astronomical Journal, vol. 120, no. 3, 2000, pp. 1579–1587. doi: 10.1086/301513.

[12] Adelman-McCarthy, J. K., et al. "The Sixth Data Release of the Sloan Digital Sky Survey." The Astrophysical Journal Supplement Series, vol. 175, no. 2, 2008, pp. 297-313. doi: 10.1086/524984.

[13] Eisenstein, D. J., et al. "Detection of the Baryon Acoustic Peak in the Large-Scale Correlation Function of SDSS Luminous Red Galaxies." The Astrophysical Journal, vol. 633, no. 2, 2005, pp. 560–574. doi: 10.1086/466512.

[14] Blanton, M. R., et al. "The SDSS Coadd: Cosmic Evolution and Galaxy Population Properties from a Single Field." The Astrophysical Journal, vol. 724, no. 1, 2010, pp. 860–880. doi: 10.1088/0004-637X/724/1/860.

[15] Abazajian, K. N., et al. "The Seventh Data Release of the Sloan Digital Sky Survey." The Astrophysical Journal Supplement Series, vol. 182, no. 2, 2009, pp. 543-558. doi: 10.1088/0067-0049/182/2/543.

[16] Alam, S., et al. "The Eleventh and Twelfth Data Releases of the Sloan Digital Sky Survey: Final Data from SDSS-III." The Astrophysical Journal Supplement Series, vol. 219, no. 1, 2015, article id. 12.