STELLAR CLASSIFICATION USING MACHINE LEARNING

Authors: Saksham Somani, Kumud Dubey, Divya Kamma, Yen Tran
IDS 575: Machine Learning and Statistical Methods
Prof: Moontae Lee

Abstract

The heavenly bodies are objects that swim in outer space. The classification of these objects is a challenging task for astronomers. This article presents a novel methodology that enables an efficient and accurate classification of cosmic objects (3 classes) based on evolutionary optimization of classifiers. This research collected the data from Sloan Digital Sky Survey database. The aim of this project is to make a perfect classification model according to the data collected by the survey work that will help in distinguishing between stars, galaxies and Quasars using the many parameters given in the dataset. We will use different models for performing classification and will find the best one with highest accuracy. The paper has
used multiple algorithms to build prediction models to classify stars, galaxies, and quasars in the universe and make a comparison among three models. The results of the test have shown that the prediction accuracy of the Random Forest model reaches roughly 99 percent with a great computing efficiency, which performs the best.

Introduction

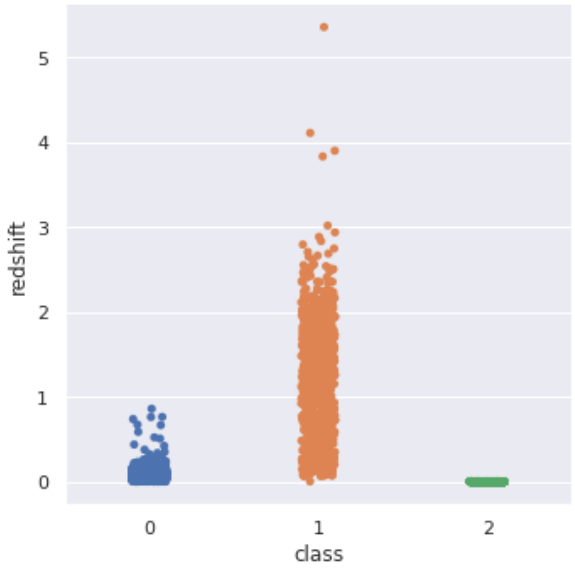
In this project, we are classifying celestial bodies observed by the Sloan Digital Sky Survey (SDSS) into Stars, Galaxies, and Quasars. The sky is filled with several celestial bodies that we observe daily. All of them have their own unique characteristics. We analyzed these characteristics to find the most important and differentiating characteristics in order to effectively classify stars, galaxies and quasars.

But before we start working on the technical aspects of the project, we first need to understand what these celestial bodies are. Starting with stars, a star is an astronomical object that contains a very luminous spheroid of plasma. It is all held together by the gravity of the star. They are the center and the reason for the formation of any galaxy. There are about 1022 to 1024 stars in the universe. Galaxies are a gravitationally bound system of stars, planets, interstellar gas, dust, and dark matter. There are more than 100 billion galaxies in the universe. Fairly lesser known to most, quasars just like stars are highly luminous active galaxies. It is very difficult to differentiate between a quasar and a star. The main difference between the two is that a quasar rotates very fast and emits enormous amounts of energy that is much more than a star. Also, a quasar is brighter than a star. There are about 750,000 quasars in the universe.

Previous Work

There is a significant increase in research works related to stellar spectra detection and classification. Many researchers focused on star quasar (Zhang et al. 2011; Jin et al. 2019; Zhang et al. 2009, 2013; Viquar et al. 2018), galaxy quasar (Bailer-Jones et al. 2019) or star-galaxy Philip et al. (2002) binary classification. Others (López et al. 2010, Becker et al. 2020) focused on multi-class classification of stars, galaxies and quasars Cabanac et al. (2002); Acharya et al. (2018). In these works, various methods have been applied to automatically classify the heavenly bodies accurately. Many authors used classical machine learning algorithms such as support vector

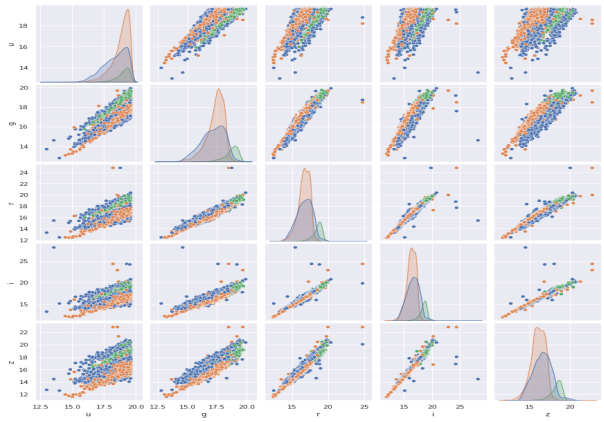
machines (SVM) or k-nearest neighbors (kNN) (Zhang et al. 2011, 2009, 2013; Tu et al. 2015,[12,15],Jin et al. 2019; Viquar et al. 2018). Authors of Zhang et al. (2013) used the SVM classifier for the same purpose.

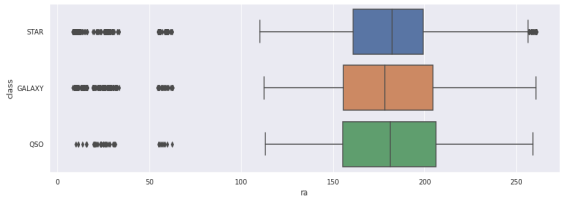


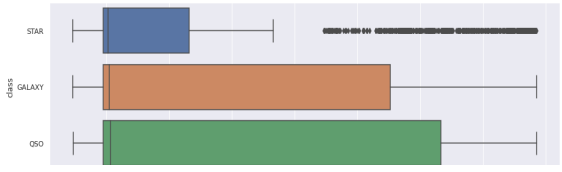
Dataset

The data consists of 100,00 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 list might get extended/reduced when we add more data to our dataset. The variable are as follow:

- ➔ obj_ID = Object Identifier, the unique value that identifies the object in the image catalog used by the CAS
- ➔ alpha = Right Ascension angle
- ➔ delta = Declination angle (at J2000 epoch)
- ➔ u = Ultraviolet filter in the photometric system
- ➔ g = Green filter in the photometric system
- ➔ r = Red filter in the photometric system
- ➔ i = Near Infrared filter in the photometric system
- ➔ z = Infrared filter in the photometric system
- ➔ run_ID = Run Number used to identify the specific scan
- ➔ rereun_ID = Rerun Number to specify how the image was processed
- ➔ cam_col = Camera column to identify the scanline within the run
- ➔ field_ID = Field number to identify each field
- ➔ spec_obj_ID = Unique ID used for optical spectroscopic objects (this means that 2 different observations with the same spec_obj_ID must share the output class)
- ➔ class = object class (galaxy, star, or quasar object)
- ➔ redshift = redshift value based on the increase in wavelength
- ➔ plate = plate ID, identifies each plate in SDSS • 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







Testing & Performance Measure

A. Testing

Our basic testing protocol was to withhold 30% of the data (hereafter referred to as the 'test set') for method comparison after our various learners have been trained, cross validated (if necessary) and undergone preliminary testing using the other 70% of the data. This 70/30 split was done prior to commencing building our various learning machines and the test set was sequestered until we were ready to perform our final testing. Some of the classifiers have parameters that can be optimized using cross-validation. In general, 30%of the remaining 75% of the data was kept back to perform the optimization. This 30% will be referred to as the 'validation set', while the remaining 70% will be called the 'training set'. We stress here that the sequestering of the true test data is essential for the unbiased comparison of our different classifiers. If one uses the same data to optimize an algorithm as to test there is a strong risk of 'training to the test set', a phenomenon that will almost certainly lead to poor real world results.

B. Performance Measures

For both testing and for optimization, we have used four measures commonly applied to classification problems. These are the accuracy (A), the precision (P), the recall (R) and the F1 score. They are defined in terms of the number of true/false positives/ negatives (tp, tn, fp and fn) as follows:

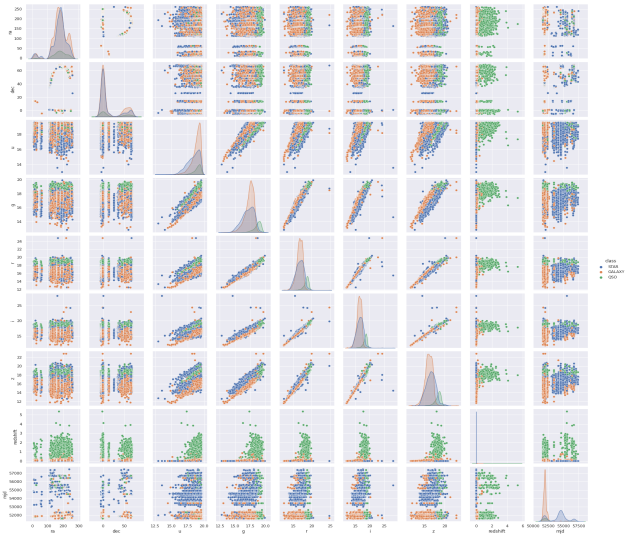
$$A = \frac{t_p + t_n}{t_p + t_n + f_p + f_n} \quad (1)$$

$$P = \frac{t_p}{t_p + f_p} \quad (2)$$

$$R = \frac{t_p}{t_p + f_n} \quad (3)$$

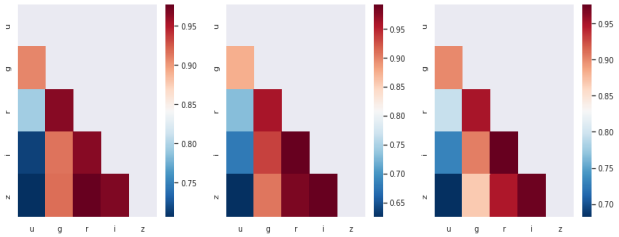
$$F_1 = \frac{2PR}{P + R}. \quad (4)$$

For our problem, we define the positive classes to be those corresponding to real objects. The choice of which of these measures should be used to measure success is rather dependent on the problem at hand.



Feature Engineering Using PCA

During data analysis, high correlation in the light band variables (u', g', r', i', z') is observed. The correlation found between magnitudes is to be expected, since the magnitudes contain information about the total brightness of an object and its spectral shape. Those 5 light-bands are substituted by lower the number of variables produced by PCA algorithm. The number of principal components set for this is 3. This helped us to keep over 99% of explained variance. Hence, the training and testing time is significantly reduced.



Model Introduction

KNN Algorithm: K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

Logistic Regression: Logistic Regression is a statistical method for analyzing a dataset in which there are one or more independent variables that determine an outcome. The intention behind using logistic regression is to find the best fitting model to describe the relationship between the dependent and the independent variable.

SGD Classifier: This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule.

Support Vector Machine Algorithms: Support Vector Machine (SVM) is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

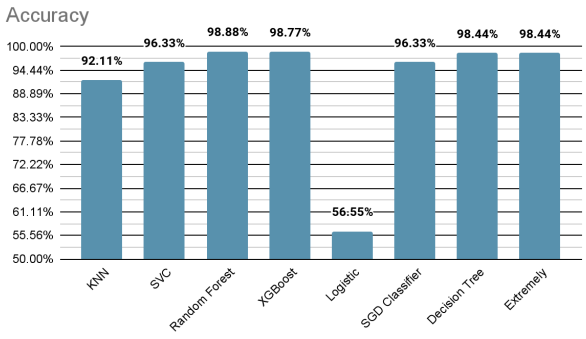
Decision Tree Classifier: Decision tree classifiers are regarded to be a standout of the most well-known methods to data classification representation of classifiers.

Random Forest Classifier: Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.

XgBoost Classifier Algorithm: XgBoost provides a wrapper class to allow models to be treated like classifiers or regressors in the scikit-learn framework. The XgBoost model for classification is called XGBClassifier. We can create and fit it to our training dataset. Models are fit using the scikit-learn API and the model.

Extra Tree Classifier: Extremely Randomized Trees Classifier is a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a forest to output its classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest.

Results



Based on the accuracy, **Random forest classifier** performed the best with an accuracy of **98.88%.** This was expected as all ensemble models tend to perform better than other models. The logistic regression model gave the least accuracy of only 56.55%. Other ensemble models including XGBoost, Extremely randomized trees (Extra Tree Classifier) and Decision trees also gave very high accuracy. K Nearest Neighbour and SCV gave a high accuracy but were comparatively lower than the tree based models.

Cross Validation

Four of the best performing models were selected for the cross validation. These were:

- Extra Trees Classifier
- Gradient Boosting Classifier
- Decision Tree Classifier
- Random Forest Classifier

We implemented **10 fold cross validation** for each of these models and found out the mean

and standard deviation of the cross validation scores. There was an improvement in the accuracy scores of the Random Forest classifier with a **mean of 98.86 and SD of 0.0045** and the Gradient Boosting Classifier with a mean of 99.04 and SD of 0.0049

HyperParameter Tuning

We selected the Random Forest classifier for Hyperparameter Tuning. First, we did a grid search based on the following parameters:

- Criterion: Gini or Entropy
- Max Features: 0.5, 0.75, 0.9, Auto
- Min_sample_leaf: 1,2,3,4
- N_estimators: 5,10,20,50,75,100

Fitting 5 folds for each of the 192 candidates meant a total of 960 fits.

The best parameters came out to be the following:

- Criterion: **Entropy**
- Max Features: **0.75**
- Min_sample_leaf: **1**
- N_estimators: **50**

Tuning the Random forest Classifier with these parameters gave an accuracy score of **99.098%**.

Precision, Recall & F1.

The final model gave a precision score of **99.39%, recall score of 99.4% and an F1 score of 99.39%.**

For our project we needed the F1 score to be high, hence we achieved the result we wanted. A high F1 score meant that there was a balance between precision and recall.

Conclusion

Based on the spectral data obtained from the Sloan Digital Sky Survey, this study trained the model with multiple machine learning classifiers and classified the galaxies, stars and quasars in the universe. From the training results of the models, it can be seen that the random forest algorithm has the best performance in the dataset, which not only has the highest accuracy rate of 98%, but also has a high computing efficiency. Compared with it, other classifiers like Gradient Boost and Extra Tree Classifier also had a high accuracy rate but the logistic regression model had very low accuracy and the support vector machine had the longest computational time.

As for the best performance in the star classification, it may be caused by the obvious difference between stellar and the other two stars in nature, resulting in the large differences in data. For the training, there are also some shortcomings that need to be improved. For example, in data processing, the combination of under- sampling and oversampling can better avoid over-fitting and improve the training accuracy.

References

1. Acharya V, Bora P, Karri N, Nazareth A, Anusha S, Rao S (2018) Classification of sdss photometric data using machine learning on a cloud. Curr Sci 115:249
2. Bailer-Jones C, Fouesneau M, Andrae R (2019) Quasar and galaxy classification in gaia data release 2. Mon Notices R Astron Soc https://doi.org/10.1093/mnras/stz2947
3. Cabanac R, De Lapparent V, Hickson P (2002) Classification and redshift estimation by principal component analysis. Astron Astrophys. https://doi.org/10.1051/0004-6361:20020665
4. Ho, Tin Kam. "The random subspace method for constructing decision forests." IEEE transactions on pattern analysis and machine intelligence 20, no. 8 (1998): 832-844.
5. Zhang, Y., Zhao, Y., Zheng, H., & Wu, X. (2012). Classification of Quasars and Stars by Supervised and Unsupervised Methods. Proceedings of the International Astronomical Union, 8(S288), 333-334. doi:10.1017/S1743921312017176
6. Wang Yao, Zheng lie, A New Tentative SMOTE Algorithm Based on Clustering. Journal of Chongqing University of Technology (NATURAL SCIENCE), 2021
7. Zhang Ge. Research on data preprocessing in course recommendation prediction Model, China New Telecommunications, 2019, 21(19)

8. Zhou Zhihua. Machine learning Beijing: Tsinghua University Press, 2016
9. Zhao Mingmei, Jin Yangyang, Wang Yujia, Zeng Mengjia. Application `Research of Random Forest Algorithm in Decision Making. Computer & Network, 2021, 22
10. Wang Xia, Dong Yongquan, Yu Qiao, GENG Na. Review of Structural Support Vector Machines Computer Engineering and Applications, 2020, 17
11. Shi Haosu. Comparative research of the ROC curve drawing based on case and MATLAB, Electronic Design Engineering, 2010, 9