Star-Galaxy Classification Using Decision Trees and Other Machine Learning Techniques

Machine Learning CS7267
Kennesaw State University, Georgia

Kelly Lavertu
klavertu@students.kennesaw.edu

Abstract

Astronomical object classification is an important issue that could aid us in our search for extraterrestrial life. As individual images of stars and galaxies can be grainy and are often difficult to discern different features by the naked eye alone, it is fundamental to the classification of these objects to train algorithms to tell the difference. Since an image of a star compared to a galaxy can seem so similar to us, these algorithms must be programmed to classify and predict as accurately as possible. Doing so could mean the difference between finding an exoplanet comparable to Earth and skipping over it. The objective of this project is to actualize a classification recognize the difference between these celestial objects. The eventual outcomes of the foremost broadly utilized Machine Learning algorithms like SVM, KNN and RFC and with Profound Learning calculations like multilayer CNN utilizing Keras with Theano and Tensorflow.

Introduction

Classification of celestial objects is fundamental to the characterization of stars and galaxies and up until recently, this was a cumbersome process. However, years of research have led to new developments in machine learning algorithms that have made it possible for us to classify these objects faster and computers can now learn and classify these objects themselves.

As part of this project, there will be an implementation of a classifier for the purposes of characterizing different types of stars and galaxies based on certain defining features. The idea for this classifier was originally implemented by Divyansh Agrawal. Image segmentation was used to identify sources in the image and the center coordinates of each of the detected sources were queried against the Sloan Digital Sky Survey database. From there, these images were reduced to 64x64 cutouts and given the appropriate label and saved into different directories based on the suggested label from the SDSS query. The output of the algorithm will give insights into how accurate the classifiers were at classifying different stars and galaxies. The final project will describe:

- Project's feature set and classifier design: what type of classifier was chosen and why
- Project's training and test data
- Implementation of classifier including the design and code development
- Classifier performance and accuracy compared to previous methods
- Strengths and weaknesses of the classification approach
- Future work including any improvements that could be made

Literature Review

In the field of machine learning there is the ubiquitous challenge of star-galaxy classification. There have been many attempts to build a successful system to classify celestial objects. One of the most challenging aspects of creating such a system is that the features of a particular celestial objects can be muddled since images can be grainy and identifiable features can potentially overlap. In this research, both the Decision Tree and Convolutional Neural Network were used, but the focus is on the CNN in the Literature Review because the CNN proved to be more accurate in more complex cases. The results will be compared to other research papers.

In this project 3,986 images were used, 997 for testing and 2,989 for training. The first model of the Decision Tree was rather disappointing with an average accuracy of 70%. The accuracy did go up when finetuning some hyperparameters of the model, such as the maximum depth of the tree and the minimum number of samples required at a leaf node. Ball et. al was able to get an accuracy of over 95% with their decision tree algorithm, but they used the entire SDSS data release 3 database.

Dataset

There is one dataset used throughout this project. This was the dataset retrieved from Kaggle and was published by Divyansh Agrawal [5].

This dataset was created by Agrawal as part of a research project at the Aryabhatta Research Institute of Observational Sciences in Nainital, India. The images were originally captured by a 1.3m telescope in the observatory 2kx2k in size and were reduced to 64x64 cutouts. Image segmentation was then used to identify sources in the image and label them based on the SDSS database. The number of features for each image is 64x64=4,096.

Agrawal made star and galaxy images accessible in a way that other sources did not—the images were already pre-processed into 64x64 cutouts. This allowed me to focus on building my algorithms and finetuning parameters to make my algorithms more efficient.

Methodology

There are 2 different algorithms used for this project.
I. Decision Tree
II. Convolutional Neural Network (CNN)

I. Decision Tree

The Decision Tree algorithm works by creating a model of decisions based on a tree-like structure. In this structure, the features of the image are used as nodes and the labels of the images are the leaf nodes. For the purposes in this project, each image from the testing set is compared pixel by pixel to each image in the training set based on the most significant features of the images. This process is repeated for each subset until the tree is fully grown. Once the tree is trained, it can be used to classify new images based on the rules of the tree and then assign the image to the appropriate class.

This type of algorithm was chosen because I wanted to compare the results of a supervised algorithm to those of an unsupervised algorithm. Often times, data preparation is effortless for the decision tree algorithm. It can also handle both numerical and categorical data, as well as multi-output problems. Finally, it is easy to understand and visualize.

II. Convolutional Neural Network

The algorithm used for classification of stars and galaxies is called a Convolutional Neural Network. This type of algorithm was chosen because it can handle large amounts of data. Since this is an unsupervised algorithm, analyzing data builds the CNN's intuition, thus increasing the learning capabilities over time.

The inputs are the set of pixels from the n by n grid the images originate from, in this case 64x64. The architecture of the algorithm, beginning with a starting node for each input pixel, also contains two hidden layers and three activation functions. The first hidden layer of the neural network is a linear layer. The input, the previously mentioned nodes, goes into 128 nodes. The next step is the first ReLU function. Following the ReLU function is the second linear hidden layer that starts with 128 nodes as input and scales down to 64 nodes. From here, there is another ReLU function to help train the model. Following the second ReLU activation function there is another linear hidden layer with an input size of 64 and an output size of 10. Finally, there is one more activation function, called Softmax.

ReLU is an activation function used for neural networks to facilitate the learning and training of the network. A ReLU activation function is essentially a node that will take a numeric input, x and will use the input if x is nonnegative or will set that input value to 0 if x has a value less than zero. This leads to faster training as the ReLU activation function leads to a cutoff for limited unspecified inputs. Softmax is another activation, and it differs from the previously mentioned ReLU in that SoftMax is used when dealing with probabilities.

A major contribution to this algorithm is the Adam optimizer. The Adam optimizer is an extension of the classical Stochastic Gradient Descent and is used to optimize the neural network by updating the network weights iterative based on training data. The entire algorithm can be thought of as a function, with parameters and inputs. The output of the function is the answer the algorithm gives based on its parameters and input. The input is given but the parameters must be tweaked to give an acceptable answer. The purpose of the Adam optimizer is to properly tune the parameters by minimizing the loss function associated with the function that is the algorithm.

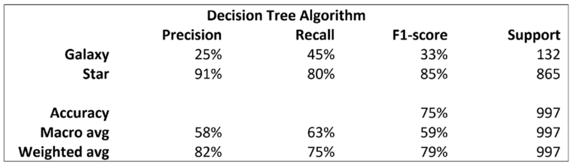
Experimental Results

I. Decision Tree Results

The accuracy of the Decision Tree testing data was around 75% on average, with the Stars averaging 85% accuracy and Galaxies 33%. This was achieved with a maximum tree depth of 6 and a minimum number of leaf nodes of 2. Ideally this accuracy will be improved with further data training and research.

While there was substantial accuracy with the Star images, it can be noted that there was a bit of a discrepancy between how many Star images there were compared to Galaxy images, which could contribute to the Star accuracy being higher.

Further finetuning of hyperparameters such as maximum tree depth and minimum leaf nodes, as well as using Principal Component Analysis or PCA.



Decision Tree Algorithm				
	Precision	Recall	F1-score	Support
Galaxy	25%	45%	33%	132
Star	91%	80%	85%	865
Accuracy			75%	997
Macro avg	58%	63%	59%	997
Weighted avg	82%	75%	79%	997

Figure 1: Classification report for Stars and Galaxy accuracy using the Decision Tree Algorithm

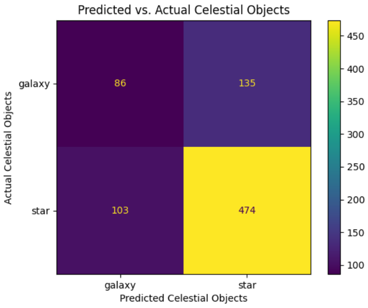
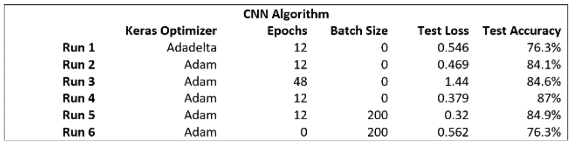


Figure 2: Confusion Matrix for Predicted vs. Actual Celestial Objects

II. CNN Results

For the initial experiment, I used the Sloan Digital Sky Survey data preprocessed by Agrawal. However, I eventually added The Galaxy Zoo dataset from Kaggle [8] which had over 60,000 images. Due to processing speeds and computational bandwidth, only a fraction of that was used (around 10,000 images). There are a total of 3,986 Star and Galaxy images used, 997 for testing and 2,989 for training. Since there wasn't enough variety in the dataset, a maximum of 87% accuracy was achieved.

The best run took 15 minutes and utilized the Adam optimizer, had 12 epochs, and a batch size of 0. Interestingly, increasing the number of epochs did not necessarily improve accuracy, and in fact led to overfitting of the data. A comparable run was that with 8 epochs, and actually got an accuracy of 86.1% but only took 10 minutes to run.



CNN Algorithm					
	Keras Optimizer	Epochs	Batch Size	Test Loss	Test Accuracy
Run 1	Adadelta	12	0	0.546	76.3%
Run 2	Adam	12	0	0.469	84.1%
Run 3	Adam	48	0	1.44	84.6%
Run 4	Adam	12	0	0.379	87%
Run 5	Adam	12	200	0.32	84.9%
Run 6	Adam	0	200	0.562	76.3%

Figure 3: Comparison of Different Runs for the CNN Algorithm

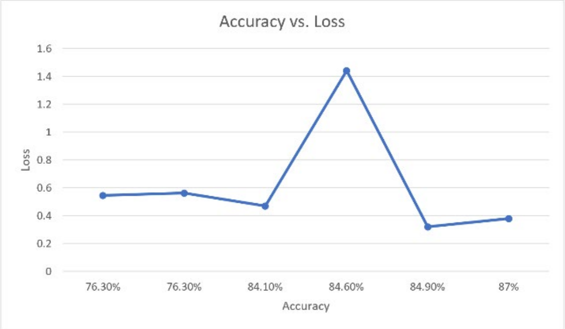


Figure 4: Accuracy vs. Loss for the CNN Algorithm

Future Work

With the rudimentary nature of this dataset, my intention is to eventually work my way into specific types of stars and galaxies, in addition to quasars and planets in my dataset. For example, I would be able to detect neutron and binary stars, and spiral and elliptical galaxies. I initially started with a simple classification of stars and galaxies to lay a strong foundation. As such, I will eventually utilize at least one additional dataset, which I have referenced at the end of this report.

Conclusion

As part of this project work, various methods for Star-Galaxy classification are compared. Over the years various scientists have proposed new methods for the classification of celestial objects which have aided in our search for extraterrestrial life.

In this project, as the number of layers in the systems used increased, the accuracy as well as the computational time increased. Comparing the two models, Decision Tree and initial CNN model, show that the CNN model works with better accuracy—the Decision Tree with an average of 75% and the CNN model with a 87% accuracy.

While the two models have differing accuracy rates, the introduction of more images will likely increase the accuracy for both. However, since this is an image classification problem, the CNN model will likely still be the better of the two since it is built to handle problems like this. This highlights the need for the CNN architecture for more complex problems.

Acknowledgement

I would like to express my gratitude to our professor, Dr. Femi William, who is guiding the process of this research project with respect to CS 7267 Machine Learning.

I would also like to recognize Agrawal and Clark et. al for their original work that helped guide my research and further thanks to Kaggle and Papers with Code for providing access to quality research.

References

[1] Ball, N. M., Brunner, R. J., Myers, A. D., & Tcheng, D. (2006). Robust machine learning applied to astronomical data sets. I. Star-Galaxy Classification of the sloan digital sky survey DR3 using decision trees. The Astrophysical Journal, 650(1), 497–509. https://doi.org/10.1086/507440

[2] Bai, Y., Liu, J. F., Wang, S., & Yang, F. (2018). Machine learning applied to star–galaxy–QSO classification and stellar effective temperature regression. The Astronomical Journal, 157(1), 9. https://doi.org/10.3847/1538-3881/aaf009

[3] Viquar, M., Basak, S., Dasgupta, A., Agrawal, S., & Saha, S. (2018). Machine learning in astronomy: A case study in Quasar-Star Classification. Advances in Intelligent Systems and Computing, 827–836. https://doi.org/10.1007/978-981-13-1501-5_72

[4] Baqui, P. O., Marra, V., Casarini, L., Angulo, R., Díaz-García, L. A., Hernández-Monteagudo, C., Lopes, P. A., López-Sanjuan, C., Muniesa, D., Placco, V. M., Quartin, M., Queiroz, C., Sobral, D., Solano, E., Tempel, E., Varela, J., Vílchez, J. M., Abramo, R., Alcaniz, J., … Taylor, K. (2021). The miniJPAS survey: Star-Galaxy Classification Using Machine Learning. Astronomy & Astrophysics, 645. https://doi.org/10.1051/0004-6361/202038986

[5] Agrawal, D. (2021, June 12). Star-Galaxy Classification Data. Kaggle. Retrieved November 26, 2022, from https://www.kaggle.com/datasets/divyansh22/dummy-astronomy-data

[6] Clarke, A. O., Scaife, A. M. M., Greenhalgh, R., &amp; Griguta, V. (2020, May 21). Identifying galaxies, quasars, and stars with Machine Learning: A new catalogue of classifications for 111 million SDSS sources without spectra. arXiv.org. Retrieved November 27, 2022, from https://doi.org/10.48550/arXiv.1909.10963

[7] Star-galaxy classification using deep convolutional Neural Networks. Papers With Code. (n.d.). Retrieved November 27, 2022, from https://paperswithcode.com/paper/star-galaxy-classification-using-deep

[8] *Galaxy Zoo - the galaxy challenge*. Kaggle. (n.d.). Retrieved December 12, 2022, from https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge