Machine Learning with Astronomical Data: Variable Object Classification

Sohum Berry

Menlo School, California

Abstract

In the current era of telescope development, new techniques must be used to make sense of the vast amounts of data that is being collected. Machine learning models present a neat solution for classifying variable objects by their measured features. The aim of this paper is to determine which machine learning model can more accurately determine the class of variable objects from the Gaia Data Release 3. The models tested are Logistic Regression Classification, Naïve Bayes Classification, Random Forest Classification, and finally Neural Networks. Random Forest Classification had the most accuracy and adjustability for this use case, however Neural Networks are also a viable option. When using a Random Forest Classification to reclassify Gaia low confidence predictions, the models largely agree on classifications of long period variable and solar like objects, however the results are not too consistent for other classifications.

Keywords: astronomy, variable objects, classification, machine learning, random forest

Introduction

With the introduction of modern day telescopes, the quantity of data that can now be collected has increased exponentially. New techniques are necessary to analyze such

quantities of data and new techniques were found in the form of machine learning. The introduction of machine learning algorithms such as a naive Bayes classifier, and more recently, neural networks, have allowed accurate and scaled classification with data collected by telescopes. Machine learning is a tool that can be used to effectively classify objects based on confirmed prior data. This type of machine learning is known as supervised learning. This paper focuses on the Gaia telescope from the European Space Agency. Located in Lagrange Point 2, Gaia's purpose is stated as such in (Gaia Collaboration et al., 2016):

> To measure the three-dimensional spatial and the three-dimensional velocity distribution of stars and to determine their astrophysical properties, such as surface gravity and effective temperature, to map and understand the formation, structure, and past and future evolution of our Galaxy.

Gaia records stellar movements and positions extremely accurately by repeatedly measuring their positions and emissions spectra time. Gaia's most recent data release was in 2022, known as the Gaia Data Release 3 (GDR3). In each of the data releases, the data is sorted into classified objects using machine learning models trained on other public data. This paper classifies the objects of GDR3 into classes that correspond to the predicted object and compares the results with those of (Eyer et al., 2023). Previous papers have explored the use of machine learning models to deal with the large quantities of data produced by modern telescopes, finding success (Sen, Agarwal, Chakraborty, & Singh, 2022). Additionally, within the larger scope of machine learning models, particular models have been identified to be more effective for astronomical use cases (Baron, 2019).

The models selected for use in this paper include the established techniques along with additional options. These models will then be tested and evaluated to determine the most

effective model for use with variable objects in particular in order to work with GDR3 data. The structure of this paper is as follows: Data will present the means of obtaining the GDR3 data, along with the various pre-processing steps to prepare it for machine model training. Method will highlight the various machine learning methods used to classify the data by using supervised learning. In Results and Discussion, the results of each model is summarized and the model best suited to such an application is determined. Additionally, potential steps that may lead to better results are provided. Finally, conclusions are drawn in Conclusions.

Data

Data Creation

Data from the Gaia Data Release 3 was obtained in two batches. The first was the variability data which includes data on minimum and maximum magnitudes of light variation through different filters along with their durations. The second is the classification summary created in (Eyer et al., 2023), which includes the confidence of the model's prediction along with the predicted class. Each object receives an objectID to ensure consistent labeling of objects across the two data groups. The data was congregated into a single file based on each object's objectID. There were occurrences of objects that were present in only one of the two datasets; these objects were removed from the dataset to ensure the model would have complete data to train on. See data quantities in Table 1.

Data Pre-processing

Class Filtering

There are 24 classes (Rimoldini et al., 2023) provided by GDR3 in which an object could be classified in. However, the models would have a significant drop in efficacy if there isn't enough

training data for a certain class. For that reason, the top 8 most common classes along with an OTHER class that contains all other objects in which the models may classify an object are selected. The selected objects are (Rimoldini et al., 2023):

1. ECL: Eclipsing binary. Two close stars moving in an orbit such that the light of one can at times be hidden behind the other in relation to Earth.

2. LPV: Long-period variable. A cool giant, or supergiant variable stars pulsating with periods from around a hundred days to more than a thousand days.

3. SOLAR-LIKE: Solar like star. A star whose brightness or other physical properties change over time as seen from Earth.

4. DSCT|GDOR|SXPHE: delta Scuti or gamma Doradus or SX Phoenicis star. Pulsating variable stars located in various constellations.

5. AGN: Active Galactic Nuclei or Quasar. A supermassive black hole emitting extreme amounts of electromagnetic radiation as gas falls towards the black hole.

6. RS: RS Canum Venaticorum variable. Close binary systems with two narrow components, both with active chromospheres and intense magnetism.

7. S: Short-timescale object. Objects with a variability that is significant enough at time lags shorter than 12 hr. (Roelens et al., 2018)

8. RR: RR Lyrae star. Pulsating stars that are old, low-mass, and have regular surface expansion and contraction.

9. OTHER: All other classifications of variable objects in order of popularity:

(a) YSO

(b) ELL

(c) BE|GCAS|SDOR|WR

(d) CEP

(e) ACV|CP|MCP|ROAM|ROAP|SXARI

(f) CV

(g) SN

(h) BCEP

(i) SPB

(j) SDB

(k) WD

(l) SYST

(m) ACYG

(n) EP

(o) MICROLENSING

(p) RCB

See Fig. 1 for the resulting spread of classes in the dataset.

Confidence Filtering

Each object in the created dataset has a confidence in its classification rated with it.

So, it stands to reason that the most effective objects to train models on would be those with the highest confidence.Data filtering thresholds are set at 20% accuracy, 30% accuracy, 50% accuracy, and 75% accuracy. The 75% and 50% datasets were used for training the models in Method.

Data Preparation

To prepare the data for model training, the input features and classified class must be extracted to continue with the supervised learning. This dataset contains 54 input features of the objects that Gaia had recorded, and the model is instructed to classify each set of features into one of 9 classes in sec. Class Filtering. Additionally, the objects must be split into a training set and a test set. $\frac{1}{10}$ of the total data was used for the test set.

Method

Four general machine learning techniques were used to classify the GDR3 variability data in order to determine what type of objects was the cause of the detected changes of brightness of a particular object. The scikit-learn library in Python was used to create these models. The following subsections will explain the models used in this paper.

Logistic Regression Classification

A multinomial logistic regression classifier (Cramer, 2002) was used to classify the light variability into the 9 classes explained in Class Filtering as opposed to a binary classification problem. Multiclass logistic regression is a machine learning technique that utilizes the softmax function to ensure that the predicted probabilities for each possible class add up to one. This allows the model to select the class with the highest probability to

return. This is preferred over multilabel logistic regression because the features correspond to mutually exclusive classes. Logistic regression classification is most frequently trained using Cross-Entropy Loss, calculating and minimizing the difference between the true class label and the predicted probability distribution.

Naïve Bayes Classification

Gaussian Naïve Bayes is the second classification technique used to classify the variable light curves. This technique uses a probabilistic approach that assumes each class follows a normal distribution. However, a Naïve Bayes classifier operates under the assumption that the features would follow a normal distribution, however feature distributions often follow skewed distributions or power laws in astronomy, which wouldn't work well with this model.

Random Forest Classifier

Random Forest is an ensemble method for classifying that creates several decision trees at training time. (Breiman, 2001) A decision tree is a hierarchical classification of data, resulting in a tree like structure with the resulting classification at the leaves after following the conditions for the data. Decision trees used individually tend to have very poor results with an inclination to over-fitting, low bias, and high variance. However, when using a large amount of such trees such as in a random forest, the errors are diminished, although at the cost of a small increase in the bias and loss of ease in interpreting the model. This is because of the use of bagging and many uncorrelated trees which lowers variance. Bagging is an abbreviation of Bootstrap Aggregating, where individual data point may be used more than once throughout the entire forest that are trained independently and in parallel.

Decision Trees

A decision tree creates splits in the given training data that demarcates the classes as best

as possible by placing thresholds on various features. (Quinlan, 1986) In higher dimensions (e.g. the 54 unique features used to train this random forest model), it becomes necessary to have a quantitative way to determine the efficacy of a split. This is where a metric known as Gini Impurity comes in. Gini Impurity is calculated by measuring how likely it is to make a wrong guess if class labels are assigned according to the distribution of classes in that node. It is calculated with the following equation:

$$G = \sum_{i=1}^{c} p(i) * (1 - p(i)) \quad (1)$$

where G is the Gini Impurity, c is the total number of classes that the object can be assigned to, and $p(i)$ is the probability of picking an object of class i, which can also be said as the percent of the current dataset that is of class i. A Gini Impurity of 0 means the model has found a perfect split. Gini Impurity is not the final step to determine the quality of a selected threshold for a decision tree though, the model must first calculate the Gini Gain. This is done by calculating the Gini Impurity of the current dataset, then the left and right branches of the proposed split. Then subtract the Gini Impurity of both branches, weighted by the number of objects in each one, from the current dataset Gini Impurity to find the Gini Gain. A higher Gini Gain corresponds to a better split, and the model calculates the Gini Gain for a certain number of potential splits before selecting the highest one, and moving onto the next branches. The process is repeated until the depth exceeds a given value or the results have narrowed down into a perfect Gini Impurity. The constants of the given maximum depth and split attempts are known as hyper-parameters, and

they have to be assigned before training. To find the best combination for a particular dataset, one must tune the hyper-parameters of their Decision Tree or Random Forest Classification Model.

Hyper-parameter tuning

The meaningful hyper-parameters that are able to be tuned in scikit-learn are as follows:

1. n_estimators: The number of decision trees to build for the ensemble classifier. This can impact the variance of the model.

2. min_samples_leaf: The minimum number of samples that must be remaining in a node for it to be a leaf node. This helps determine the depth at which each tree decides to lock in the end results.

3. max_features: The number of features to be considered by a tree in the bagging process. Using a max_features value that is less than the number of features can introduce randomness and independence among each tree in the random forest model.

4. max_depth: The maximum depth that the tree is allowed to go to. An unlimited max_depth may induce over-fitting, but a low value can increase the bias while lowering the training time.

5. min_impurity_decrease: The minimum Gini Gain that must be achieved to make a split within each tree. A low value will allow the model to capture finer details, but a high min_impurity_decrease will increase bias with limited splits.

In the Random Forest model, two hyper-parameter sets were selected after testing with scikit-learn's RandomizedSearchCV method. The first is as follows:

n_estimators = 300.

min_samples_leaf = 7.

max_features = 30.

The second has a similar accuracy, but requires less time to train:

n_estimators = 400.

min_samples_leaf = 1.

max_features = 10.

Neural Network

Neural Networks have been gaining popularity over the last decade, gaining new advancements and adoption by many people, and the rise of Large Language Models accelerating their growth. At a high level, it consists of layers of nodes that are a inspired by a human brain. The connections between various nodes hold a weight that is adjusted in a neural network's training periods. In all neural networks, there is an input layer, which receives the raw data (variability features in this case); hidden layer(s), which extract patterns and features in the data and adjust the weight accordingly; and an output layer, which provides the final classification. Since there are several classes that an object may be classified as, the neural network must use a softmax function, similar to that of logistic regression (Logistic Regression Classification). The model has an input layer of 54 nodes, the number of input features; two hidden layers of 128 nodes each and a dropout of 0.3 between; and finally an output layer of 9 nodes, the number of possible classes.

Results and Discussion

Confusion Matrices

The most effective model for classification of such objects is certainly a Random Forest Classifier, as seen in the strongest diagonal classification indicating that the predicted classifications are correct. See 4 compared to 3 and 2. The Random Forest model found the most success in classifying Active Galactic Nuclei, but struggled classifying RS Canum Venaticorum variables with a correct classification percentage of 91.36%.

ROC Curves

The Random Forest Classifier presents the best receiver operating characteristic curve for all classes. See 7 compared to 6 and 5. The Random Forest model achieved a micro-average area under the curve of 0.9997 and a macro-average area under the curve of 0.9991, performing remarkably well. The Logistic Regression model is the runner up, achieving a micro-average area under the curve of 0.9930 and a macro-average area under the curve of 0.9765. However, the Logistic Regression model struggled in classifying Active Galactic Nuclei.

Next Steps

An area for improvement is the handling of the 'OTHER' category (Class Filtering), which comprises 2.0% of the dataset (Fig. 1). Previous papers have used a combination of multiple classifiers to more accurately determine the class of an object even with lacking representation in the dataset. Additionally, testing the models on a similar but different dataset could help confirm the findings of this paper.

Comparing With Gaia

In an attempt to see how this model would compare with that used by the Gaia team, the model was tested on Gaia classifications with a confidence rating below 45%, 30%, and 20% and compared the results to that of the given classification by Gaia in confusion

matrices. The model, of course, could not be trained on the same data it is being tested on, so the model only used classifications with a 75% confidence rating or higher to train, before testing on data it had never seen before. Confusion Matrices were created to compare the classification of this paper's Random Forest model to classifications of low confidence in the GDR3 dataset. Results can be found for a confidence threshold of 45% 8, 30% 9, and 20% 10.

Conclusions

Four different models were created to classify variable objects from the Gaia Data Release 3 trained on the high confidence classifications from the release. Comparing the results, the paper finds that a Random Forest Classification model was most effective. The success of the Random Forest Classifier may be attributed to it's effective handling of a large number of input features—54 in total—through splitting the input data based on the most relevant features. An ensemble model is also ideal for a dataset with a large number of objects such as GDR3's. Another benefit to the usage of Random Forest Classifiers are the transparency that comes with their usage. The model provides feature importance levels which identify which input features have the most impact. Although the training may be more computationally intensive when compared to other models, Random Forest Classifiers result in one of the best choices for astronomical variable object classification.

The next steps using the model would be to compare Gaia's low confidence classifications to that of the Random Forest model, and determine the abilities of each one.

Acknowledgments

Many thanks to Dr. Daniel Muthukrishna, Manan Agarwal, and the Cambridge Centre for International Research for their assistance in creating this paper.

References

Baron, D. (2019). *Machine learning in astronomy: a practical overview.* Retrieved from https://arxiv.org/abs/1904.07248

Breiman, L. (2001). Random forests. *Mach. Learn.*, *45*(1), 5–32.

Cramer, J. (2002, 01). The origins of logistic regression. *Tinbergen Institute, Tinbergen Institute Discussion Papers*. doi: 10.2139/ssrn.360300

Eyer, L., Audard, M., Holl, B., Rimoldini, L., Carnerero, M. I., Clementini, G., . . . Süveges, M. (2023, June). Gaiadata release 3: Summary of the variability processing and analysis. *Astronomy & Astrophysics*, *674*, A13. Retrieved from http://dx.doi.org/10.1051/0004-6361/202244242 doi: 10.1051/0004-6361/202244242

Gaia Collaboration, Prusti, T., de Bruijne, J. H. J., Brown, A. G. A., Vallenari, A., Babusiaux, C., . . . Zschocke, S. (2016, November). The Gaia mission. *aap*, *595*, A1. doi: 10.1051/0004-6361/201629272

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81-106. Retrieved from https://api.semanticscholar.org/CorpusID:189902138

Rimoldini, L., Holl, B., Gavras, P., Audard, M., De Ridder, J., Mowlavi, N., . . . Eyer, L. (2023, June). Gaiadata release 3: All-sky classification of 12.4 million variable sources into 25 classes. *Astronomy & Astrophysics*, *674*, A14. Retrieved from http://dx.doi.org/10.1051/0004-6361/202245591 doi: 10.1051/0004-6361/202245591

Roelens, M., Eyer, L., Mowlavi, N., Rimoldini, L., Lecoeur-Taïbi, I., Nienartowicz, K., . . .

Wevers, T. (2018, December). Gaia data release 2: Short-timescale variability processing and analysis. *Astronomy & Astrophysics*, *620*, A197. Retrieved from http://dx.doi.org/10.1051/0004-6361/201833357 doi: 10.1051/0004-6361/201833357

Sen, S., Agarwal, S., Chakraborty, P., & Singh, K. P. (2022, February). Astronomical big data processing using machine learning: A comprehensive review. *Experimental Astronomy*, *53* (1), 1-43. doi: 10.1007/s10686-021-09827-4

Table 1.

Quantity of Objects From GDR3 Files.

Dataset	Items in Dataset
Variable Summary	15,939,968
Variable Classification	9,976,881
Total Combined Dataset	15,939,968
Filtered Combined Dataset	514,315

Figure 1

Number of objects classified as each class from the combined dataset along with their percentage of total objects in the legend. OTHER contains classes that were not common enough to train a model on.

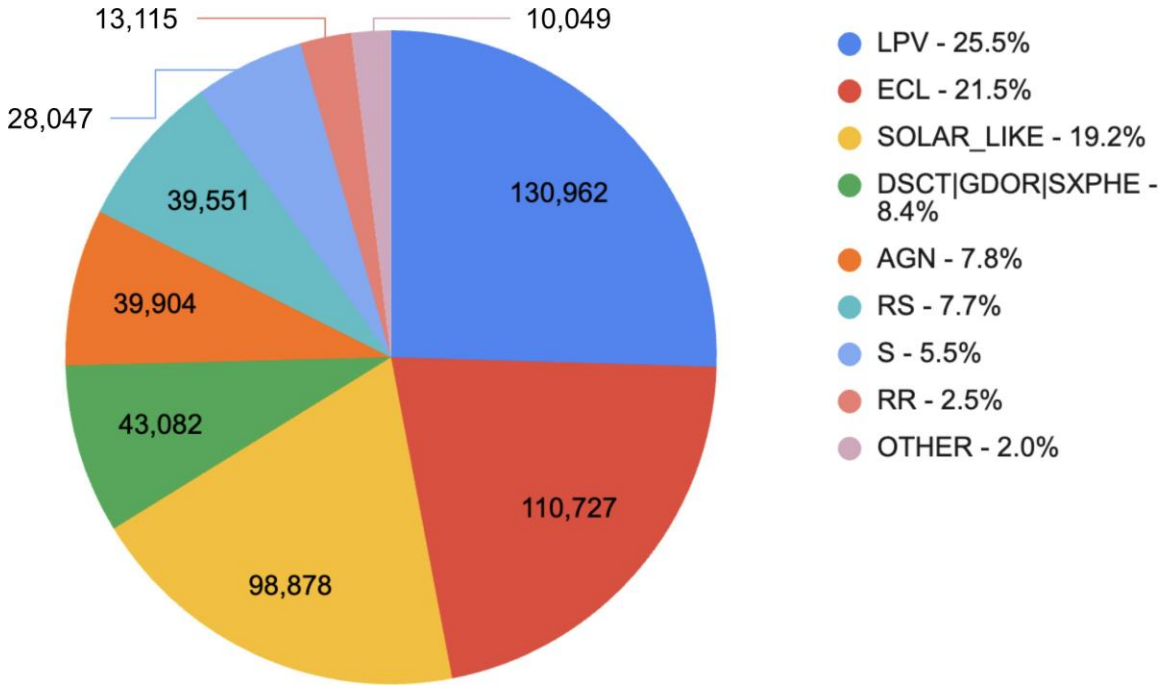

Figure 2.

Confusion Matrix from Logistic Regression Classification normalized over the predictions.

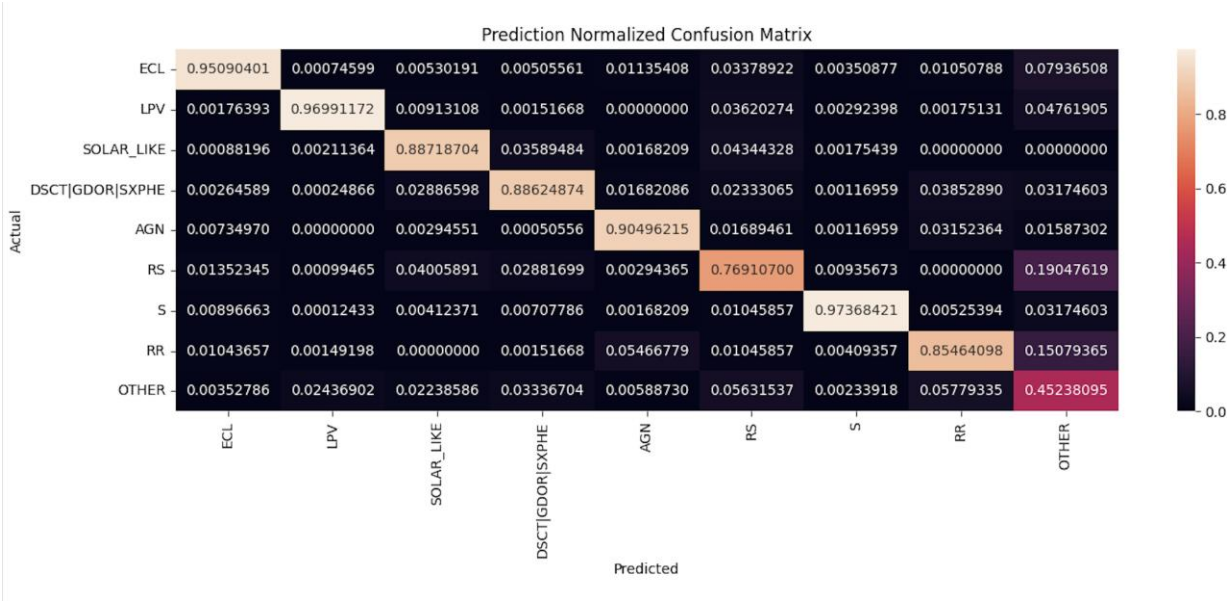

Figure 3.

Confusion Matrix from Naïve Bayes Classification normalized over the predictions.

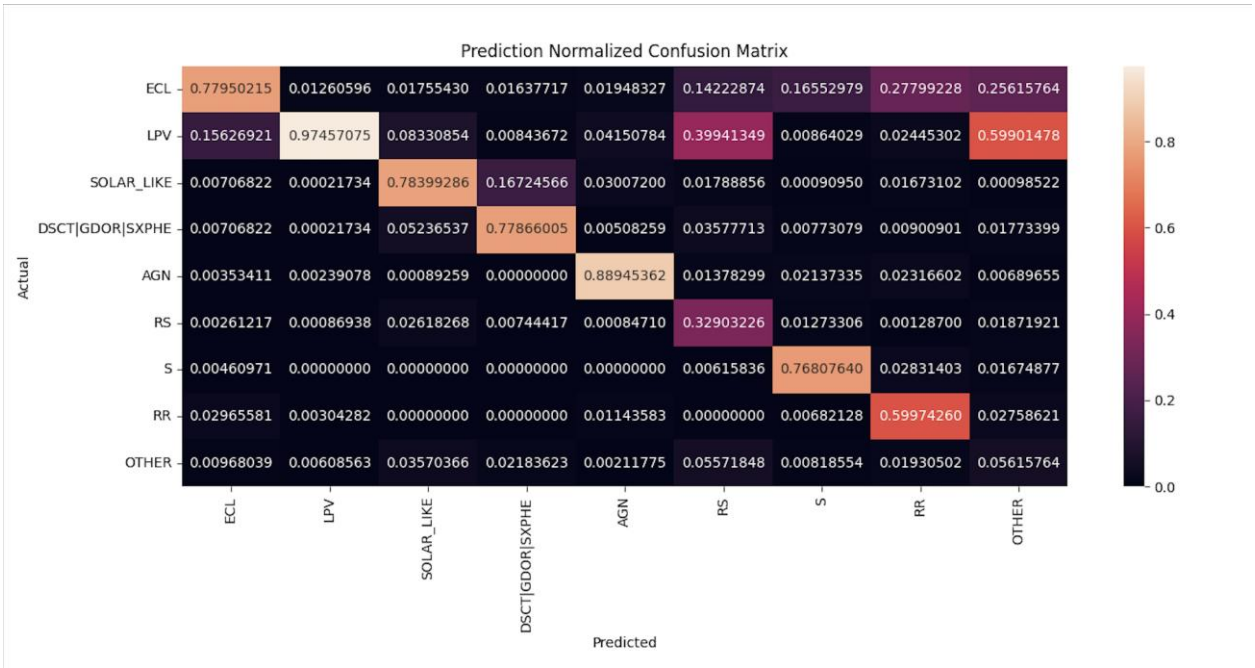

Figure 4.

Confusion Matrix from Random Forest Classification normalized over the predictions.

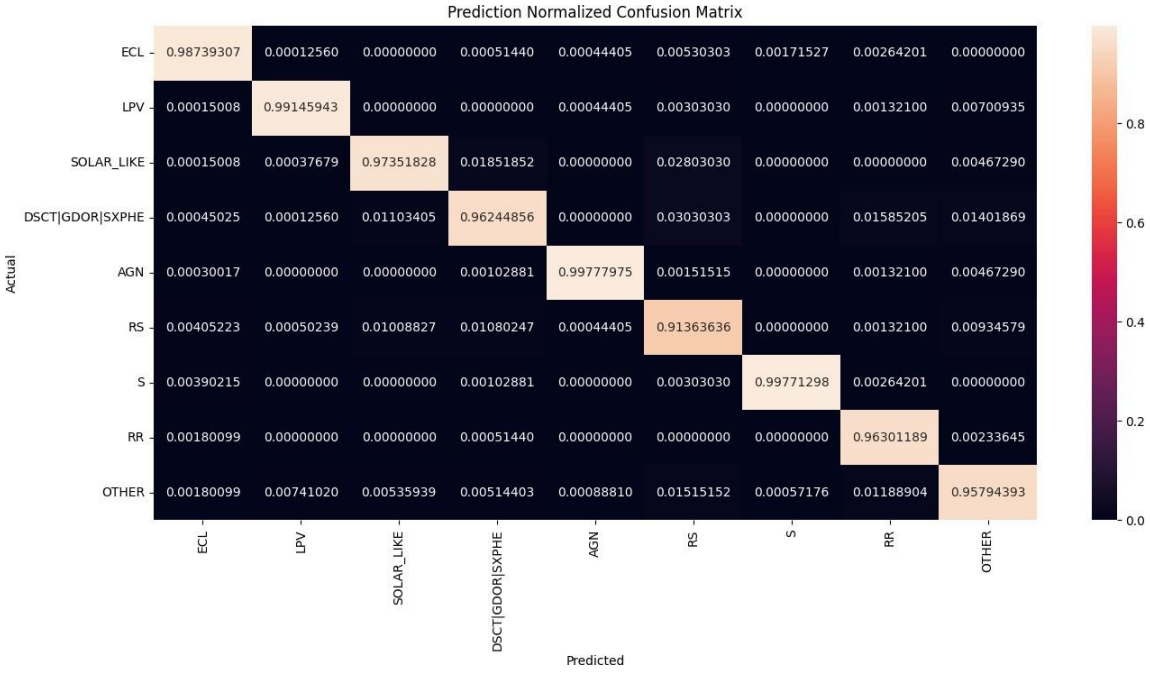

Figure 5.

ROC Curve from Logistic Regression classification.

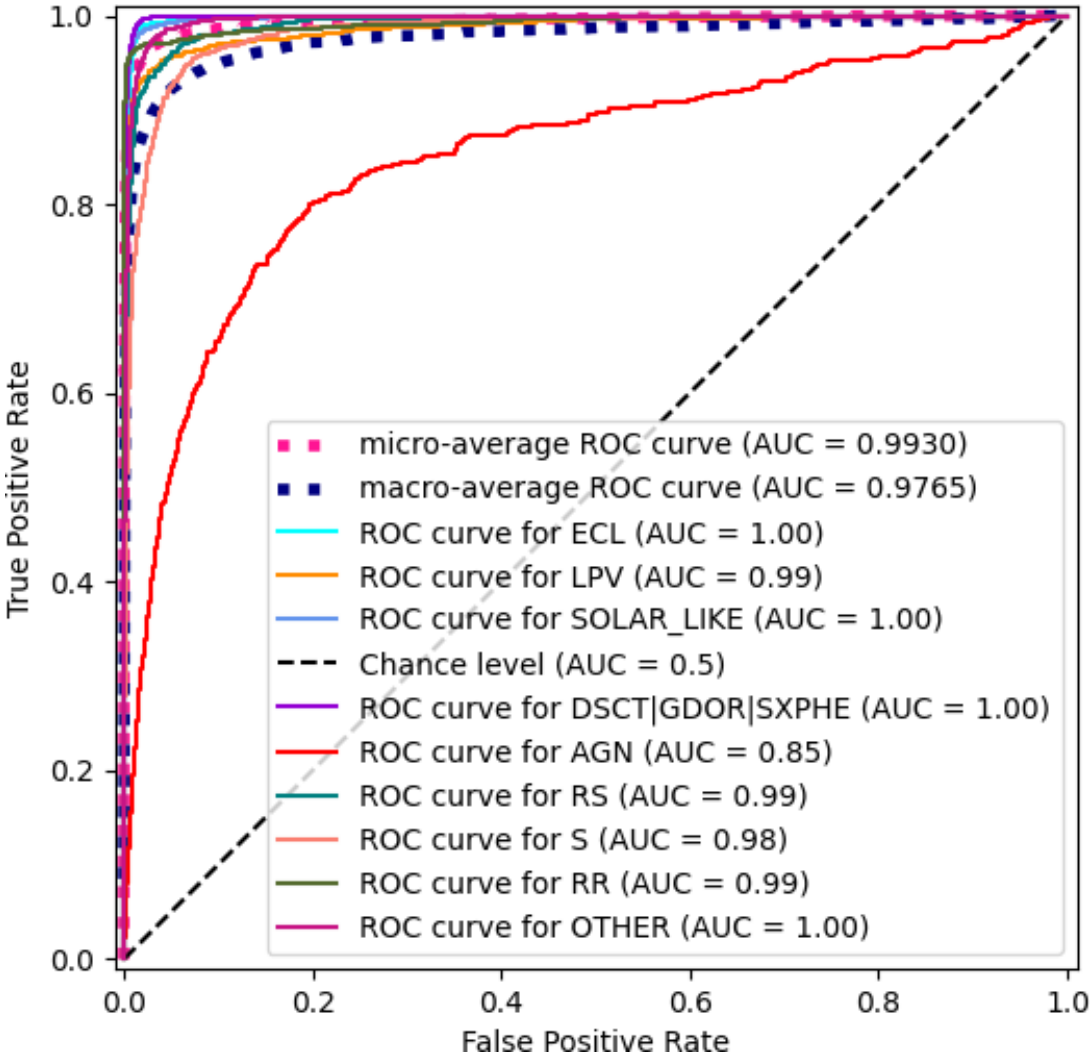

Figure 6.

ROC curve from Naïve Bayes classification.

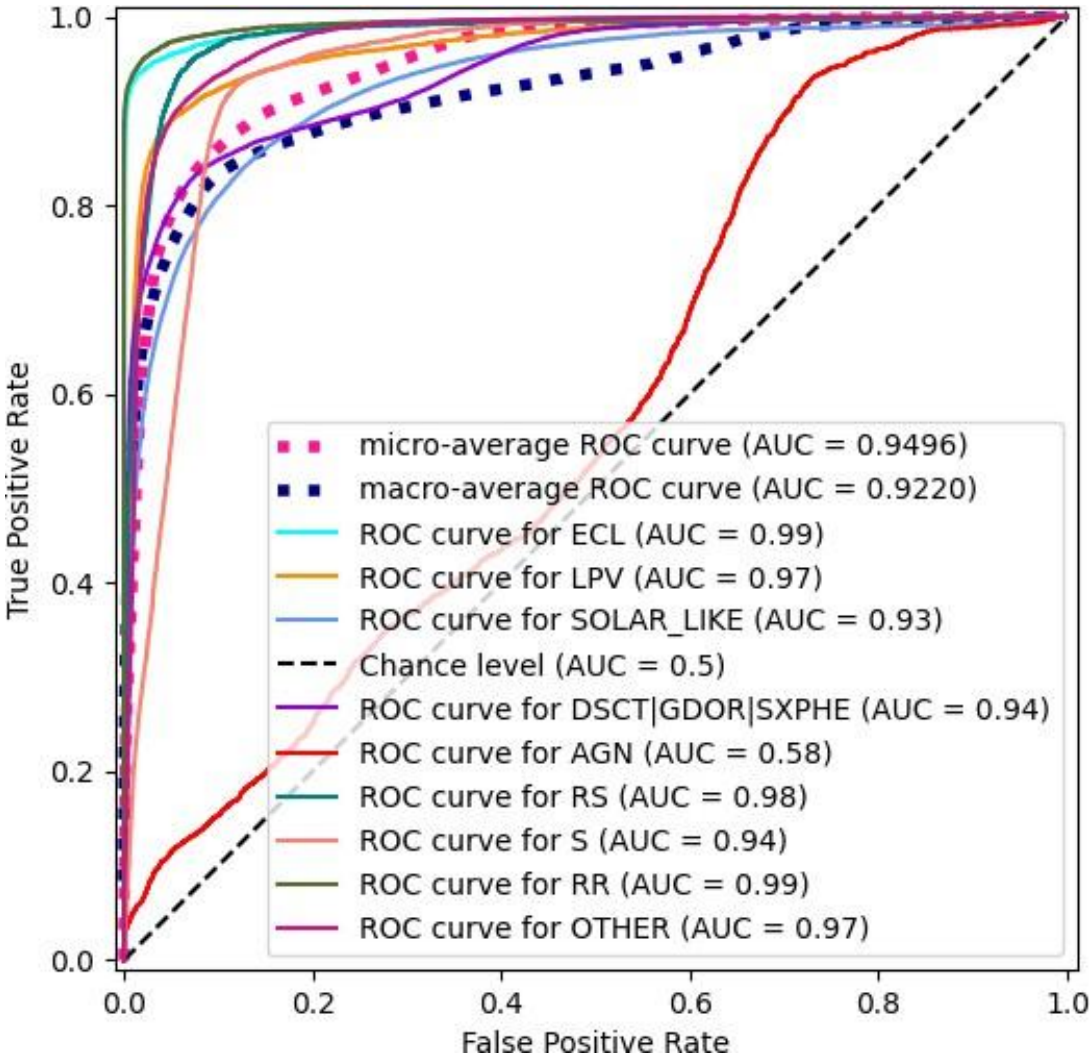

Figure 7.

ROC curve from Random Forest classification.

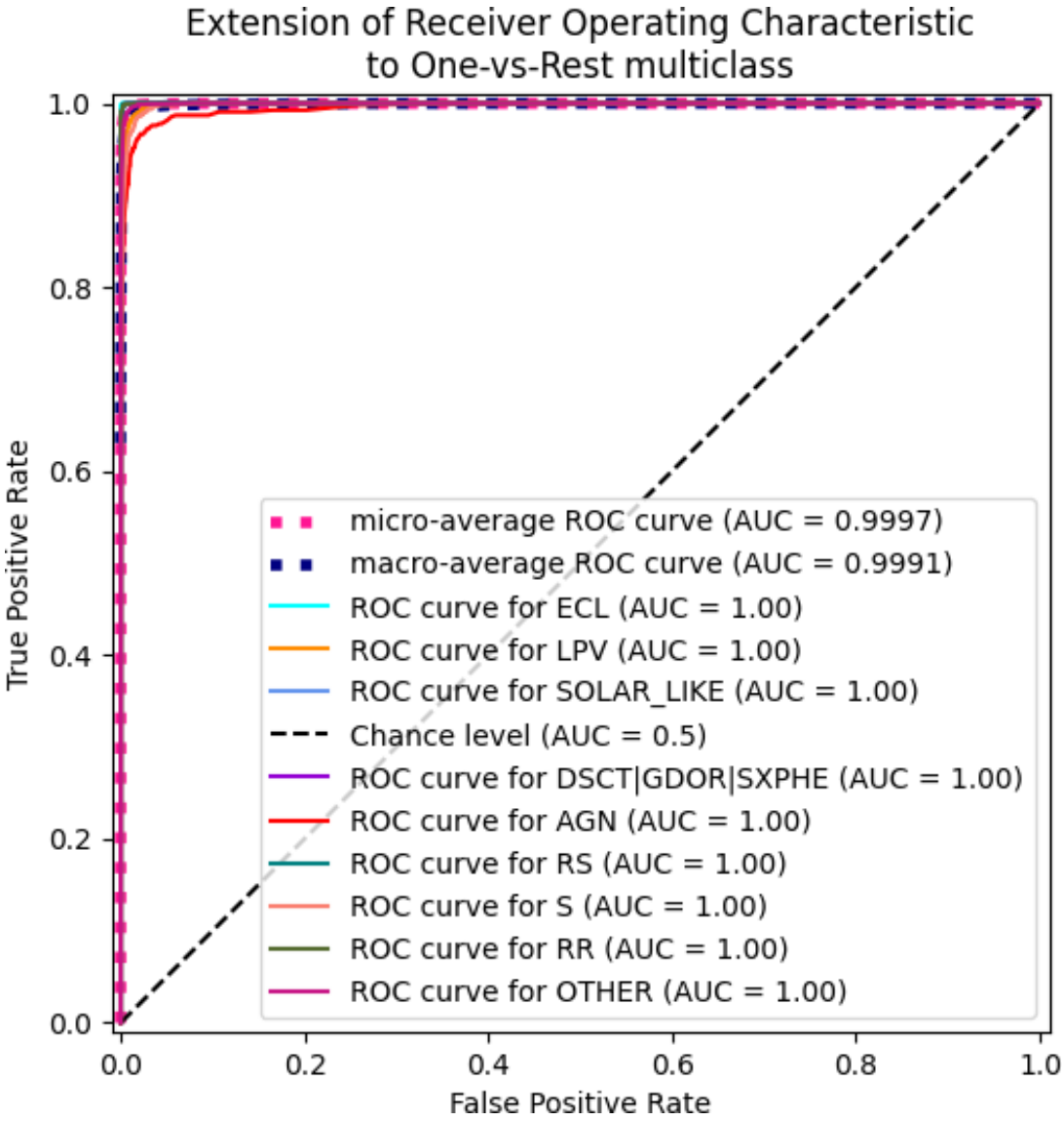

Figure 8.

Confusion Matrix from Random Forest Classification Compared to Gaia Confidence Predictions Below 45%.

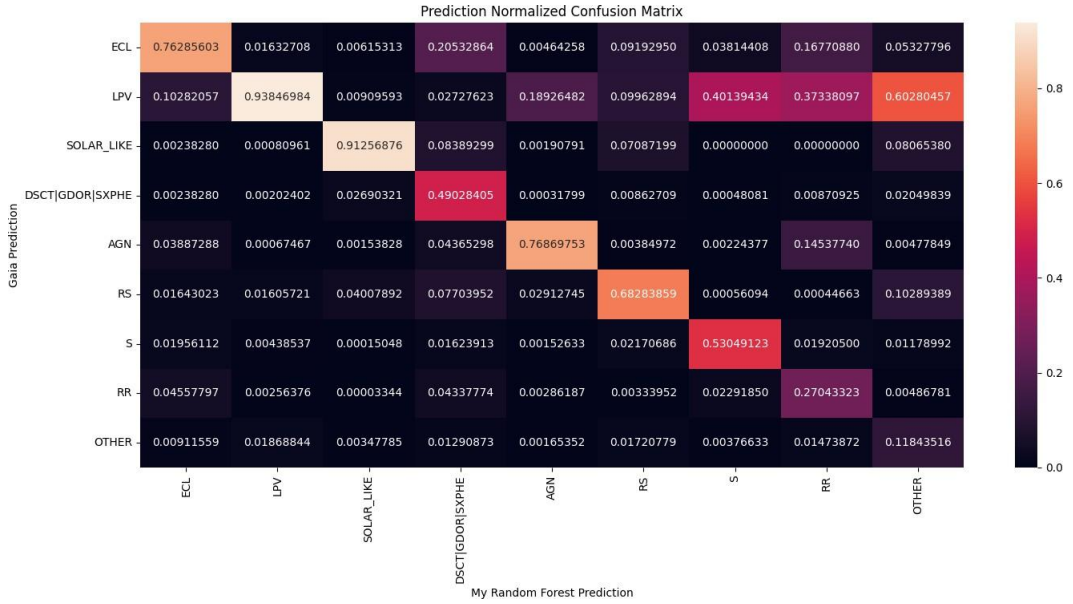

Figure 9

Confusion Matrix from Random Forest Classification Compared to Gaia Confidence Predictions Below 30%.

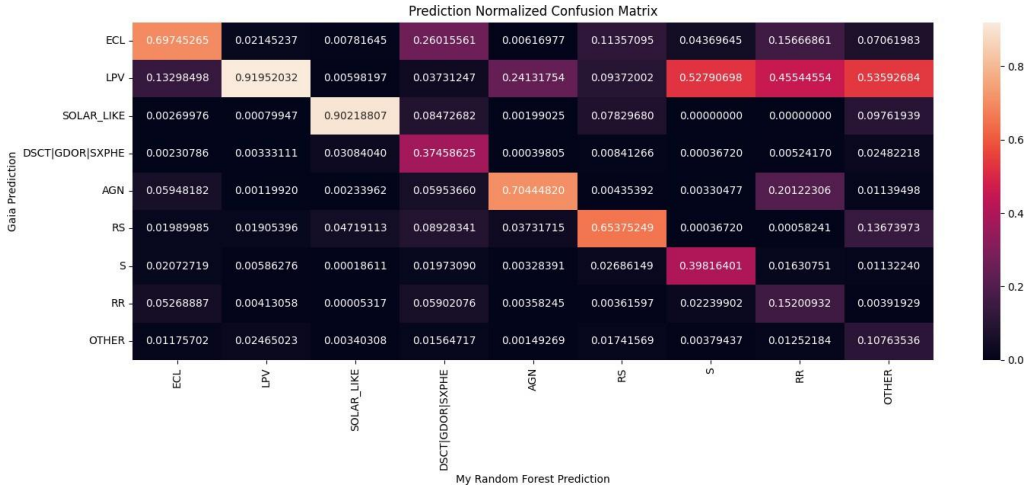

Figure 10

Confusion Matrix from Random Forest Classification Compared to Gaia Confidence Predictions Below 20%.