Star/Galaxy Classification in a Galaxy Cluster Field Using Machine Learning Techniques

Zacharias Escalante
Brown University Department of Physics
GitHub

1 Introduction

The importance of distinguishing between light-emitting sources is vital in the field of high-precision observational cosmology, especially for ground-based telescopes, which must account for atmospheric distortions when taking long-exposure images. For the astrolensing group at Brown University, of which I am a member, improving the accuracy of star-galaxy classification leads to better estimates of the masses of galaxy clusters, which give us insight into the evolution of large-scale structure and the role of dark energy. I selected a catalog of sources found in field-of-view of the Abell 3266 galaxy cluster (see Figure 1), one of the clusters observed in our group's recent LoVoCCS survey [1, 2].

Figure 1: Combined coadded irg-band image of central region of Abell 3266 galaxy cluster.

The dataset consists of 1857685 rows (sources) and 47 columns (features). Four of these columns contained no information at all, while one column was simply an index counter, so these were dropped immediately. The 'extendedness' feature was chosen as the target variable. This feature takes a binary value of 0 if our prior analysis predicts the source as a star, and 1 for a galaxy. The 41 remaining original columns were chosen as the feature array. Twenty of these features contained magnitude information in 5 different photometric bands, 13 features described the shape of the source through different estimations, and the remaining 8 features contained position information and various other quantifications. For EDA, model training, and all other data analysis, the OSCAR supercomputer at Brown University's Center for Computation and Visualization was utilized.

2 EDA

I first searched the dataset for any non-physical infinite values (of which there were 1997) and replaced them with nulls. Of the 41 features used, the 'psf_used' feature was categorical, while the others contained float values. I then used .describe to examine some summary statistics of select features, especially the maximum and minimum values, to compare with their expected ranges. Out of the $\sim$1.8 millions points, around 72%

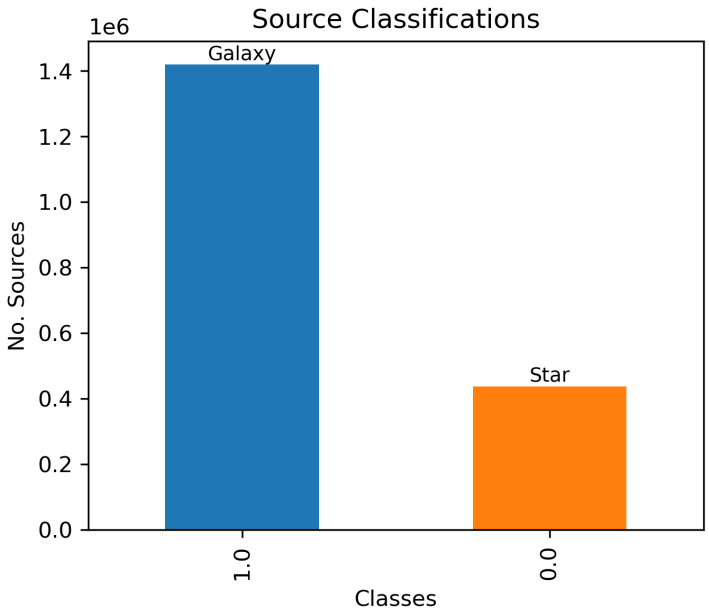

Figure 2: Source (target variable) distribution in the galaxy cluster field dataset. Though the classes are relatively balanced, a stratified split was used in the final analysis.

contained missing values. The target variable distribution is shown in Figure 2, with a ∼75%/25% split between class 1 and class 0 points, a fairly balanced class distribution.

Next, I made some visualizations for select features and compared them with the target variable. Shown in Figure 3 is one example of the quantity of outlying points found within some columns, with the feature 'e1' plotted with the target variable. We expect 'e1' – a measure of the source elongation along one axis – to lie in [-1,1]. One feature in particular – known as 'res' (resolution) – appeared to be most predictive of the target, with distinct distributions for each class as shown in Figure 4. We can note that a large proportion of class 0 points (stars) hold smaller values of 'res', with a trailing tail, while galaxies have a more uniform distribution of 'res' values.

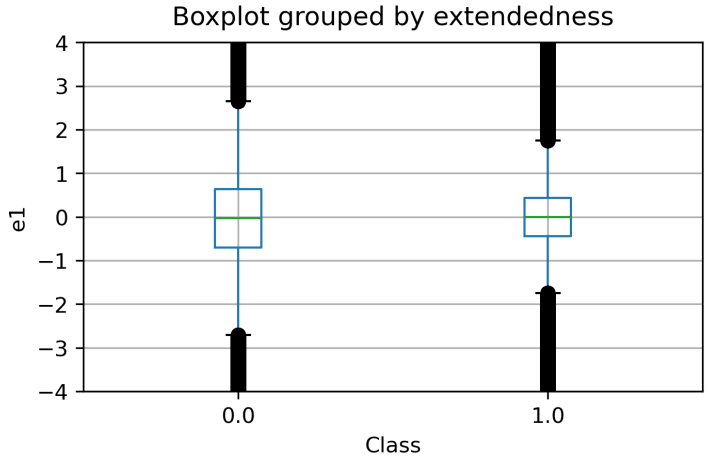

Figure 3: Boxplot of feature 'e1' plotted with target variable. The blue boxes denote the first and third quartile values, while the green horizontal line denotes their median values. Each extended blue "whisker" displays 1.5 times the inter-quartile range, and the thick, black lines denote outlier values.

Finally, I engineered 5 new features (one for each corresponding photometric band) of the magnitude differences 'psf_mag' and 'cmodel_mag' (e.g. 'u_psf_mag' - 'u_cmodel_mag'), based on prior knowledge that psf models are better suited for stars, while the cmodel method is suitable for both star and galaxy modeling. The result for one such band is shown in Figure 5. After feature engineering, the dataset contained 46 total features.

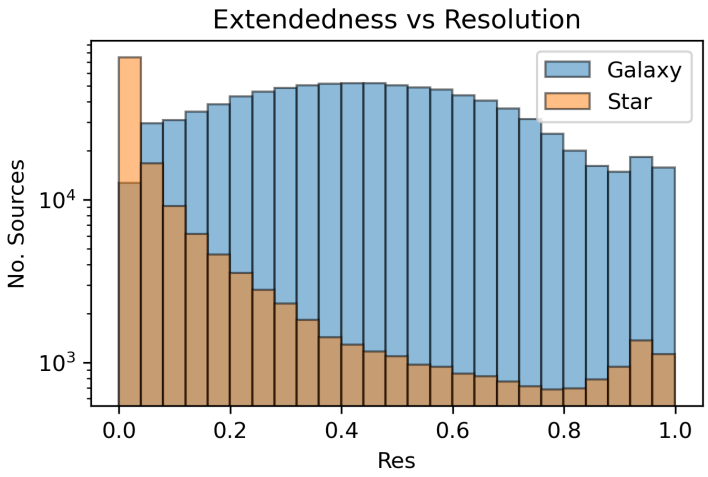

Figure 4: Distribution of sources over the range of 'res' values, for each class.

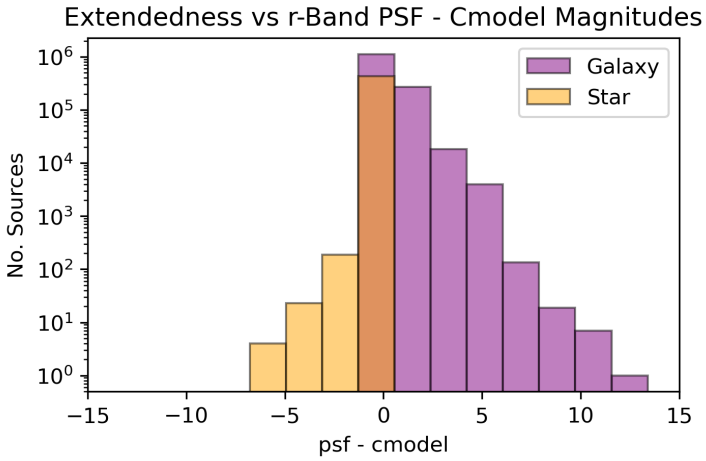

Figure 5: Category-specific histograms of magnitude differences for photometric r-band feature, for each 'extendededness' class. Class separation is more pronounced than distributions for each respective standalone magnitude.

3 Methods

The class distributions were well-balanced for the purposes of this project, however I chose to use a stratified split for the sake of uniformity. Also, due to time constraints and size of the dataset, I did not use any k-fold methods, and instead chose a simple 60/20/20 data split for training, validation, and testing sets. Since the majority of points in my dataset contained missing values, I performed a multivariate imputation on my dataset instead of removing incomplete rows. This was carried out using scikit-learn's IterativeImputer method. For feature value scaling, I used MinMaxScaler on 18 features due to their clear theoretically-defined ranges, and StandardScaler on the remaining 27 continuous features. Since the single categorical feature was already one-hot encoded, these values passed through the pipeline column transformer unchanged. These preprocessing steps were then performed on the training data with .fit_transform() and on validation and testing with .transform().

Four different models were utilized for this project: Logistic Regression with Elastic Net, a Support Vector Classifier (SVC), a K-Neighbors Classifier (KNN), and a Random Forest Classifier (RF). Each model was trained on 5 random states to account for variance from data splitting, and evaluated using the accuracy metric, the number of correctly classified points out of the entire test predictions. This metric was chosen because of the equal importance given to the classification of each source type. A range of hyperparameters – shown in Table 1 – were fed to each model using ParameterGrid to search for the optimal combinations.

For each of the four models, a function was created which implemented random state iterations, data splitting, preprocessing, training over a parameter grid, and testing to find the optimal model for each state. The following were accepted as inputs: the feature and target arrays, parameter grid, MinMax and Standard

ML Model	Hyperparameter	Values
Log. Reg. (Elastic Net)	l1_ratio	np.linspace(0,1,3)
	C	1/np.logspace(-5,5,5)
SVC	gamma	[1e-1, 1e0, 1e1]
	C	[1e-2, 1e-1, 1e0, 1e1, 1e2]
	kernel	linear
K-Neighbors Class.	n_neighbors	[10, 100, 1000]
	weights	['uniform','distance']
Random Forest Class.	max_depth	[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
	n_estimators	[1, 3, 10, 30]

Table 1: Hyperparameters used in ParameterGrid for model training.

features to be scaled, and the number of random states desired. The best models were saved, and the best test scores and models were returned as outputs.

One important point to note is that while the logistic regression model was trained using 100% of the data, the other three models utilized only 2% of the entire dataset. This decision was driven by computational times for model training exceeding the project deadline times. For final model performance comparison, the models trained on 2% of the data were then evaluated on the same test set drawn from the full dataset.

4 Results

The optimal model parameters for each of the 5 random states are summarized in Table 2. With a uniform stratified split for each state, the baseline accuracy is 76%. Figure 6 displays the mean accuracy scores and standard deviations of each model, averaged over all random states. From the mean values alone, we find that the k-neighbors classifier performed best, with value of 0.86, while the logistic regression model performed worst at ∼80%. However, it is important to reiterate that the non-logisitic regression models were trained on 2% of the dataset, and in general performed consistently better on their respective subset of test data. While the KNN model score did not shift much when evaluated on the full test set – only a 1% drop in accuracy – the SVC and random forest classifier displayed mean accuracies of ∼99% on their subsets.

ML Model	Optimal Values
Log. Reg. (Elastic Net)	(l1_ratio = 0, C = 100000)
SVC	(gamma = 0.1, C = 100)
K-Neighbors Class.	(n_neighbors = 10, weights = 'distance')
	(n_neighbors = 10, weights = 'uniform')
Random Forest Class.	(max_depth = 4, n_estimators = 3)
	(max_depth = 7, n_estimators = 30)
	(max_depth = 9, n_estimators = 30)
	(max_depth = 7, n_estimators = 1)

Table 2: Optimal hyperparameters values found from 5 random states.

I then examined the results of one KNN model through a confusion matrix, shown in Figure 7, finding values of accuracy, precision, and recall of 0.85, 0.94, and 0.86, respectively. These relatively high values indicate low rates of false positives and false negatives, and a fairly balanced dataset.

Next, I inspected the feature importance of the same chosen KNN model. Starting with permutation global importance (Figure 8), we find that the 'res' feature does appear to have the greatest impact on model predictability, while 2 of the 5 engineered features (i and r band differences) appear in the top 10 as well, validating the importance of including these features. For the SHAP-generated global importances (Figure 9), we find that the r-band difference takes the highest importance, with i and z band differences also appearing. However, the resolution feature no longer appears in the top 10 for SHAP global importance, indicating the necessity of generating importance metrics using multiple methods to account for their variances.

One example of local importance is displayed in Figure 10. For this specific point, we again find that the r-band difference has the greatest (positive) impact on the prediction, while the r-band psf magnitude also contributes (negatively). The fact that the r-band magnitude data appears among the most important features for all plots is no surprise: the coadded images taken in r-band have the highest resolution and depth, and are therefore the sole band used to obtain our cluster lensing maps and mass estimates.

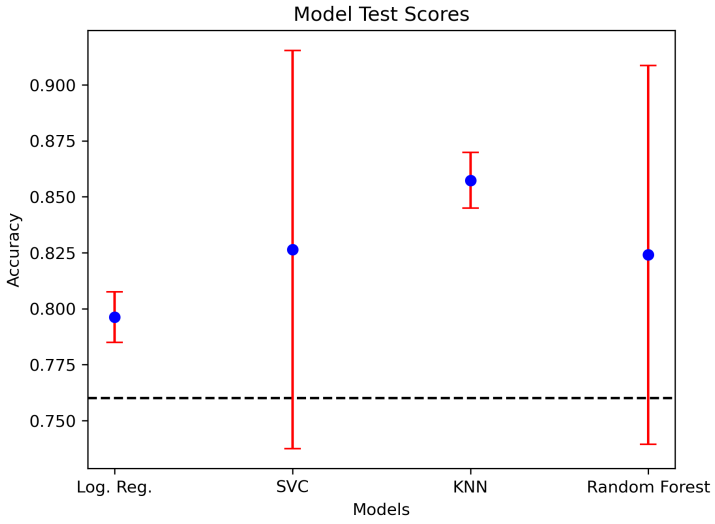

Figure 6: Model accuracy test scores averaged over 5 random states, with standard deviations plotted in red. The black, dashed horizontal line represents the baseline accuracy of 76%.

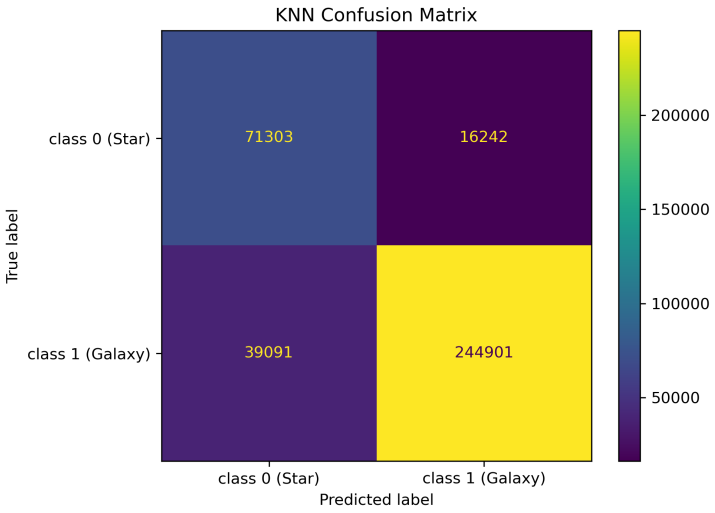

Figure 7: Confusion matrix results of KNN classifier for 1 of 5 random states.

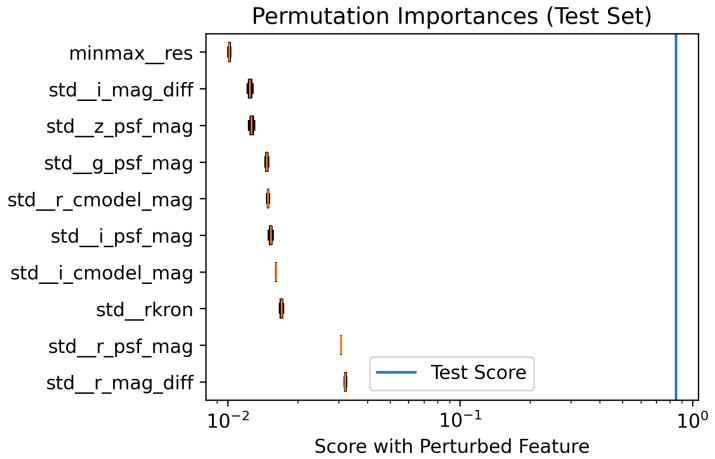

Figure 8: Permutation global feature importance visualization for the KNN model used in Figure 7.

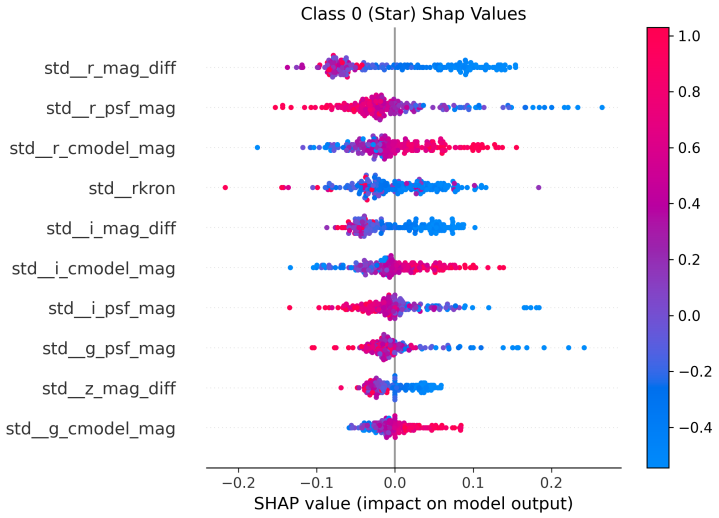

Figure 9: SHAP global feature importance for the KNN model used in Figure 7. This plot is generated from class 0 points.

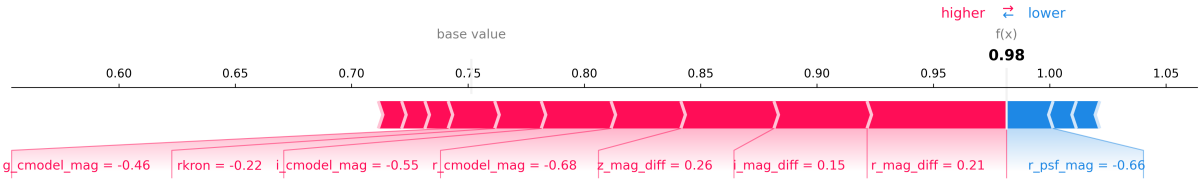

Figure 10: SHAP force plot for local feature importance of a randomly-selected point.

5 Outlook

The primary means of improving this project lies in including the full dataset to train all models, rather than using only 2% for 3 of the 4 classifiers. This would allow for a more consistent model comparison. Next, I would have liked to search a wider parameter range in each of the models to more carefully find the optimal validation scores. Finally, it is possible that the catalog may not be entirely i.i.d., due to source blending in our field-of-view leading to duplicate rows, but this would require a more careful source selection prior to aggregation into a catalog, and is beyond the scope of this project.

References

[1] S. Fu, I. Dell'Antonio, Z. Escalante, J. Nelson, A. Englert, S. Helhoski, R. Shinde, J. Brockland, P. La-Duca, C. Larkin, L. Paris, S. Weiner, W. K. Black, R.-R. Chary, D. Clowe, M. C. Cooper, M. Donahue, A. Evrard, M. Lacy, T. Lauer, B. Liu, J. McCleary, M. Meneghetti, H. Miyatake, M. Montes, P. Natarajan, M. Ntampaka, E. Pierpaoli, M. Postman, J. Sohn, D. Turner, K. Umetsu, Y. Utsumi, and G. Wilson, "LoVoCCS. II. Weak Lensing Mass Distributions, Red-Sequence Galaxy Distributions, and Their Alignment with the Brightest Cluster Galaxy in 58 Nearby X-ray-Luminous Galaxy Clusters," (2024), arXiv:2402.10337 [astro-ph].

[2] S. Fu, I. Dell'Antonio, R.-R. Chary, D. Clowe, M. C. Cooper, M. Donahue, A. Evrard, M. Lacy, T. Lauer, B. Liu, J. McCleary, M. Meneghetti, H. Miyatake, M. Montes, P. Natarajan, M. Ntampaka, E. Pierpaoli, M. Postman, J. Sohn, K. Umetsu, Y. Utsumi, and G. Wilson, The Astrophysical Journal **933**, 84 (2022), publisher: IOP ADS Bibcode: 2022ApJ...933...84F.