COSC 325 - Final Report

Daniel Blackston
The University of Tennessee, Knoxville
bwf589@vols.utk.edu

Grayson Gill
The University of Tennessee, Knoxville
ggill5@vols.utk.edu

Ethan Miller
The University of Tennessee, Knoxville
emill104@vols.utk.edu

***Abstract*— Galaxy classifications hold varied implications for astronomers and physicists but are prohibitively difficult to determine manually in large quantities. In this paper, we present a supervised learning approach to classify galaxies based on shape and structure. Our objective is to train, test, and implement a computationally simple and interpretable model for galaxy classification, in contrast to existing complex or computationally intensive deep learning models. This work contributes a lightweight solution to a longstanding problem, reducing the need for labor-intensive manual classification of galaxy data.**

Keywords—Machine Learning, Galaxy Morphology, Galaxy Classification, Supervised Learning

I. Introduction

A galaxy's morphology, or its shape and structure, offers valuable information regarding its age, formation, and environment. Accurate and efficient galactic classification is essential for astronomers and astrophysicists spanning countless applications, and the number of known galaxies far exceeds human capacity for manual classification. Our model, inspired by a previous Galaxy Zoo method [1], contributes a computationally viable solution to this problem. Traditionally, similar projects utilized mass human participation and/or complex neural network models to produce large-scale classifications. Though effective, these models require significant time and resources.

Our project explores a more lightweight solution. Using the Galaxy10 SDSS dataset, we have constructed a trio of test models that provide classifications well beyond random chance, all with minimal computational overhead. All 3 models (one-vs-rest stochastic gradient descent, k-nearest-neighbors, and random forest) intentionally avoid deep learning techniques.

Each model's performance is evaluated on raw accuracy using confusion matrices and F1 scores. Initial results show promise for certain galaxy classes, although challenges remain due to the ambiguity of certain classes and overall imbalanced class distribution within the dataset. To better interpret model behavior, our research group tracked a learning curve to evaluate accuracy as a function of training set size. This information provided clues to the scalability of our models as well as a quantitative measure of overfitting. Finally, our research group visualized a normalized confusion matrix for each model's predictions to better interpret misclassifications across classes.

II. Data Exploration

The Galaxy10 SDSS Dataset [2] is a collection of 21,785 galaxy images hand-labeled by expert human annotators into 10 classes based on form. Each image is a 69x69 matrix of pixels, and each pixel contains a green, red, and infrared scalar between 0 and 255. A threshold of 55% agreement on class between human classifiers was enforced for inclusion in the dataset to reduce the number of ambiguous images.

As visualized in Fig. 3, the red, green, and infrared components of each galaxy are highly correlated. Preliminary exploration of a stochastic gradient descent model trained on every nonempty subset of the green, red, and infrared channels showed no significant differences in model capacity as shown in Table 1. As such, the research team reduced the original dataset to only include red light data.

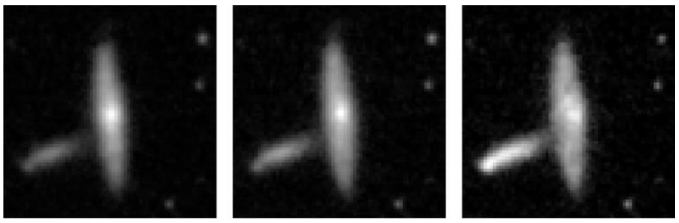

Fig. 1 – A class 4 galaxy filtered by Red, Green, and Infrared channels, as well as grayscaled for direct comparison.

Table 1	*Subset*	*Accuracy*
1	Green	34%
2	Red	35%
3	Infrared	35%
4	Green + Red	37%
5	Green + Infrared	35%
6	Red + Infrared	35%
7	Green + Red + Infrared	35%

Each image was flattened from a 2-dimensional array of dimensions (69, 69) to a 1-dimensional array of red values normalized to the range [0, 1]. The resulting X input data is a 21,785 x 4,761 matrix of values, and the y values are the corresponding classifications. The data was split into 90% training and 10% test data using y-stratification to ensure a uniform representation of classes in each set. Standard Scaler was used to center and scale the data. As a significant portion of each image is empty and contributes little to the model, Principal Component Analysis was implemented for n=1000 to reduce the number of features, both combating overfitting and lightening the computational load.

Even with stratified data splits, the class distribution is highly imbalanced. The number of samples in the original dataset corresponding to each class ranges from 17 to 6997. Additionally, certain classes (for example, Class 4 - Disk, Edge-On, Rounded Bulge and Class 5 - Disk, Edge-On, Boxy Bulge) are visually ambiguous as shown below, producing difficulties in machine classification that will be explored later.

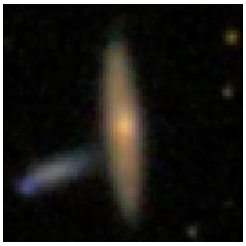

Fig. 2 – An example of a Disk, Edge-On, Rounded Bulge galaxy.

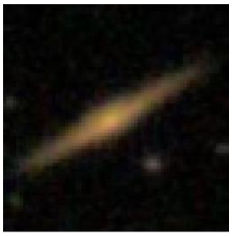

Fig. 3 – An example of a Disk, Edge-On, Boxy Bulge galaxy.

III. Technical Approach, Results, and Discussion

Between the midterm report and final report, the research team made a significant pivot away from our original proposed final model (Stochastic Gradient Descent), instead improving another model (Random Forest Classifier). Our decision to change models was based on a reanalysis of the learning curves, as well as the failure of all proposed model improvements to increase the accuracy of our original model.

Our original model, Stochastic Gradient Descent (SGD), was chosen for its efficiency and scalability. Instead of using the entirety of the data to calculate the gradient, SGD randomly samples a single image from the dataset for each iteration, making it more suitable for large datasets. Because this was a multi-class classification problem, we used the One Vs. Rest Classifier (OvR). OvR makes predictions based on the class with the highest confidence score. In our preliminary efforts, the model was trained with the following hyperparameters: a logistic loss function (log_loss), a maximum of 5000 iterations, and an "optimal" learning rate parameter. Overall, the resulting model's performance did not meet our expectations. The confusion matrix showed a significantly higher prediction rate for the classes with more data (Figure 4).

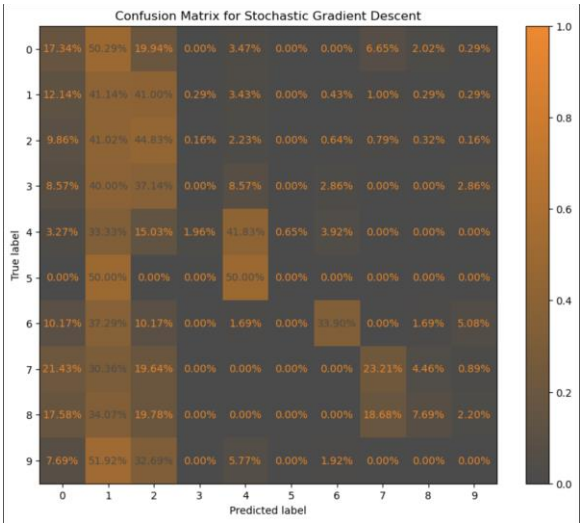

Fig 4. - A confusion matrix showing the predicted vs. true labels for the One Vs. Rest Classifier model with Stochastic Gradient Descent.

The classification report and learning curve displayed an average accuracy score of 35%, indicating significant overfitting and a failure to generalize to unseen data. To improve generalization and reduce overfitting, we introduced Ridge (L2) regularization. Contrary to our expectations, this adjustment did not yield any performance gains. As shown in the learning curve (Figure 5), the model's accuracy plateaued at around 35%, indicating no benefit from the regularization.

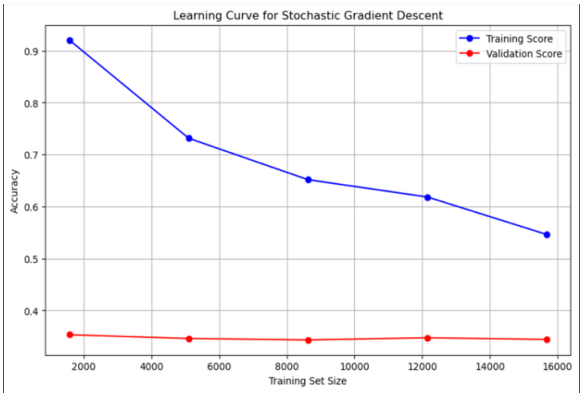

Fig 5. - A graph showing the learning curve of the One Vs. Rest Classifier with Stochastic Gradient Descent.

To address the issue of class imbalance, we set the "class_weight" hyperparameter to "balanced", allowing the model to automatically adjust class weights inverse to their frequencies. Additionally, we changed the "loss" hyperparameter from "log_loss" to "hinge". Despite these efforts, the model performed even worse, with accuracy dropping as low as 20%.

As a final attempt to improve the SGD model, the research team implemented an ensemble method of models trained not only on the red channel, but on a mixture of channels and differences between channels. Though we hypothesized that by-pixel differences in light values across bands would reveal elements of galaxy form that are useful to the classifier, the information gain from these additional ensemble models contributed little to nothing to the overall classifier. As such, we chose to entirely abandon the SGD model.

Next, we attempted to use a K-Nearest Neighbors Classifier (KNN) model inspired by one of the papers we researched [4]. KNN makes predictions based on the labels of the K closest training data points in the feature space. After tuning the "n_neighbors" hyperparameter, we found "5" to be the optimal K-value. The confusion matrix showed a high prediction accuracy with Class 1 due to its high frequency in the dataset. However, for example, the classifier predicted Class 5 for every Class 4 galaxy, an issue likely stemming directly from the low representation of Class 5 in the original dataset (Figure 6).

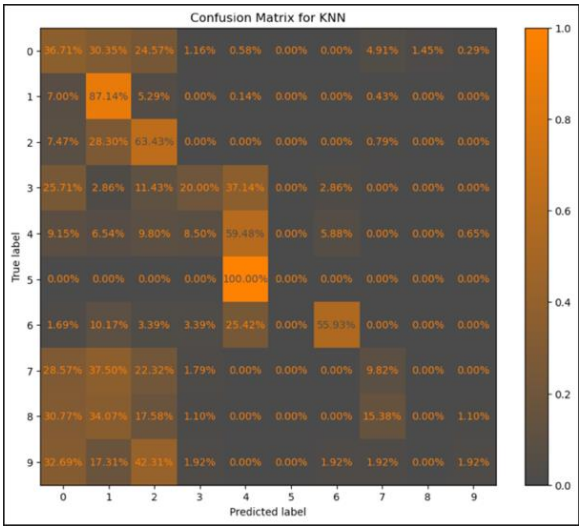

Fig. 6 – A confusion matrix showing the predicted vs. true labels for the KNN Classifier model.

The classification report and learning curve displayed an average accuracy score of 58% and evidence of moderate overfitting. The gap between the training and validation scores was consistently between 10 and 15%, indicating moderate improvement and generalization (Figure 7).

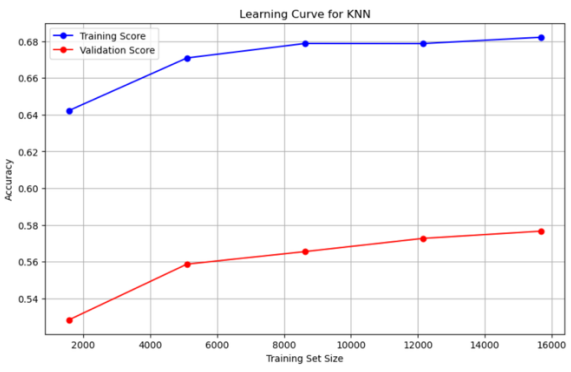

Fig. 7 – A graph showing the learning curve of the KNN Classifier model.

Overall, the KNN model achieved a substantially higher accuracy compared to SGD. It also exhibited rapid execution with minimal computational overhead. However, the model's complete misclassification of Class 4 as Class 5 was a significant problem, highlighting KNN's sensitivity to class inbalance, and it revealed that underrepresented classes are often situated near more dominant classes in the dataset. Despite KNN's improved overall performance, the incorrect classification of minority classes suggested a need for additional strategies.

Following the limitations observed from the KNN model, we lastly chose to refine our previous Random Forest Classifier (RFC) model. Random Forest is an ensemble method that compiles the predictions of multiple decision trees. Each tree gets a random subset of the data to ensure the model is less

prone to individual tree errors. Initially, the model was configured with default hyperparameters, except for "n_jobs" set to "-1" to ensure all available cores of the CPU could be used.

The confusion matrix of the original RFC model showed a better balance of predictions than both the SGD and KNN classifiers, but there was still a poor performance on classes with less data, as shown in the confusion matrix (Figure 8).

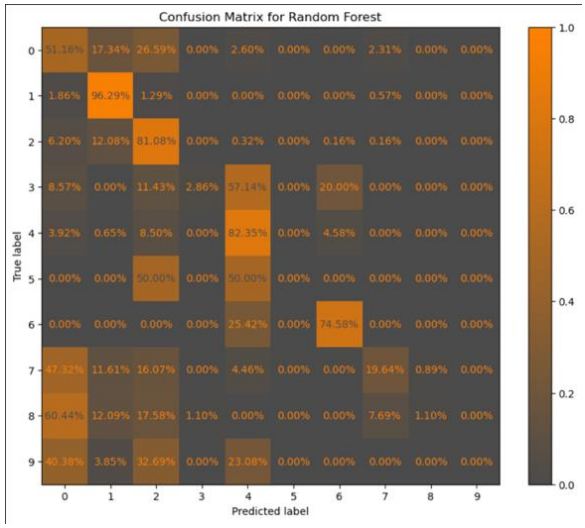

Fig. 8 – A confusion matrix showing the predicted vs. true labels for the original Random Forest Classifier model.

The classification report and learning curve displayed an average accuracy score of 68%, evidence of more robust generalization to unseen data than the previous models. To further improve performance, we applied hyperparameter tuning using Grid Search Cross Validation. The resulting hyperparameters were as follows: "bootstrap=False", max_depth=none", "min_samples_leaf=1", "min_samples_split=5", and "n_estimators=200". This tuning process led to an increase in accuracy to 71% over the baseline model's 68%. However, the resulting learning curve showed significant overfitting, with the training accuracy never falling below 100% (Figure 9). This indicated that the model was memorizing the training data.

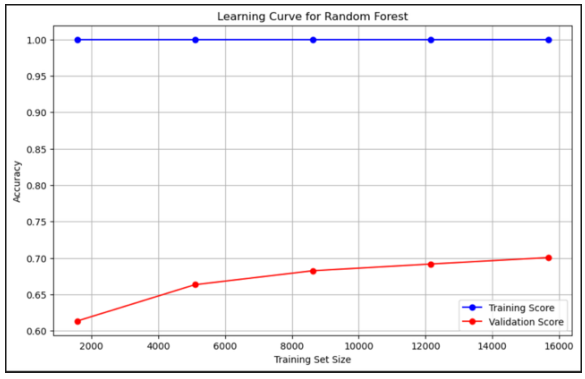

Fig. 9 – A graph showing the learning curve of the Random Forest Classifier model post hyperparameter tuning.

Given the overfitting introduced by the Grid Search hyperparameter tuning, we opted for a more controlled approach by manually adjusting the RFC model's hyperparameters. We set "n_estimators" to "100" and "max_depth" to "10". Additionally, we applied Stratified K-Fold Cross Validation with 5 folds to ensure a more balanced evaluation across class distributions and to further prevent overfitting. The resulting confusion matrix with this configuration showed a slight decrease in overall classification accuracy (Figure 10).

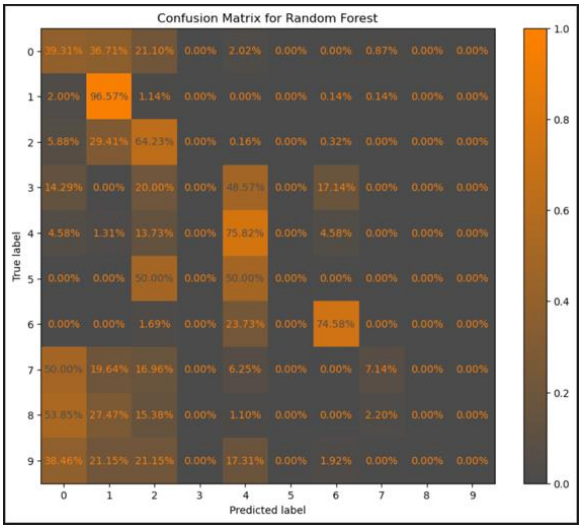

Fig. 10 – A confusion matrix showing the predicted vs. true labels for the improved Random Forest Classifier model.

The classification report and learning curve displayed an average accuracy score of 63%, slightly worse than Grid Search RFC model (Figure 9). While there was a decrease in classification accuracy, the learning curve significantly improved. The training accuracy no longer remained fixed at 100%, and the validation-training score gap narrowed to approximately 12%. These results indicated that the manually adjusted RFC model significantly reduced overfitting while maintaining relatively strong performance.

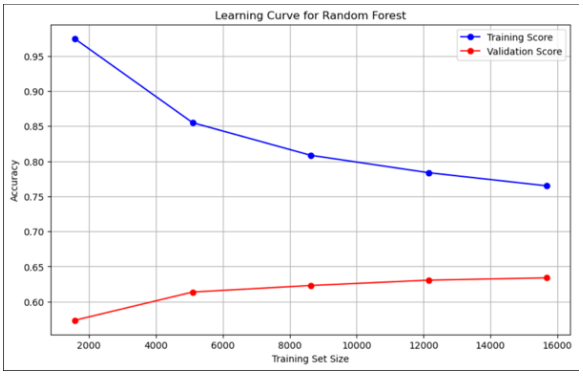

Fig. 11 – A graph showing the learning curve of the improved Random Forest Classifier model.

Considering the results above, the research team has concluded that the final RFC model best meets our goals. It generalizes to unseen data most accurately and cleanly, while maintaining solid computational efficiency. Our original Midterm Report decision to use the SGD model was based on both a misinterpretation of the learning curves, and unfounded optimism regarding how our proposed improvements would affect the SGD model's performance. The convergence of training and test scores after readjusting the hyperparameters for the RFC was an unexpected benefit that we did not consider before the midterm report. Overall, we are satisfied with our final model's performance.

IV. Contributions

Ethan Miller was responsible for the processing of the data in its entirety. He wrote the process for flattening, normalizing, splitting, and scaling the original data, as well as all three baseline baseline models. Additionally, he designed all of the graphics, including confusion matrices and learning curves.

Daniel Blackston was responsible for multiple key decisions in the data handling process, including the decisions to isolate the red channel and apply Principal Component Analysis for feature reduction, as well as researching possible future directions for the project. He produced the entire final draft of the midterm report document, revising and rewriting his groupmates' first draft for a consistent style and improved clarity. He designed the slides for the entire final presentation, as well as chose to switch our model from SGD to RFC.

Grayson Gill was responsible for the layout and first draft of the entire midterm report (excluding the data exploration and baseline model sections). He produced all of the references and formatting for the report, as well as the formatting and comments for the Jupyter notebook submission. The specific baseline model selections came from his literature review, and he was instrumental in keeping our group on track with deadlines. After the final models were created, Grayson was responsible for a significant portion of the Technical Approach, Results, Discussion, and Conclusion sections for the final report.

All three members feel that the distribution of work was even and fair. It is also important to note that our group worked alongside another group (the group analyzing the same dataset) throughout the entire process, and we would like to attribute much of our progress to that mutual collaboration.

V. Conclusion

Our goal was to achieve a lightweight model for galaxy classification using the Galaxy10 dataset, a collection of approximately 22,000 images across 10 distinct galaxy classes. During our exploration, we aimed for an efficient and minimally hardware-intensive solution so that the implementation could eventually be included in something like a telescope or a smartphone. Our final model used a Random Forest approach, which ultimately offered the best accuracy and scalability. It achieved a maximum accuracy of 64 percent, with solid but inconsistent precision across classes and some signs of overfitting. We also evaluated two other models, Stochastic Gradient Descent (SGD) and K-Nearest Neighbors (KNN), but KNN was very computationally intensive, and SGD did not generalize well. The dataset showed an uneven distribution in the galaxy classes, with some classes having as few as 900 images while others had thousands. Despite applying stratification and hyperparameter tuning to address the class imbalances, we observed that certain classes still performed poorly due to the skewed data. Achieving higher accuracy is still a challenge we face, and we believe that using a more advanced technique such as convolutional neural networks (CNNs) may be necessary to develop a high-performing and accurate model with an image set this large. Additionally, we were inspired by the work of other groups. We recognize that shrinking input image sizes from 69 x 69 to a 23 x 23 pixel format could lead to further improvements in not only performance but accuracy as well and is a promising direction for our next steps.

VI. References

[1] A. Guruprasad, "Galaxy classification: A machine learning approach for classifying shapes using numerical data," arXiv preprint, arXiv:2312.00184, Dec. 2023.

[2] "Galaxy10 SDSS Dataset — astroNN 1.1.0 documentation," Readthedocs.io, 2017. https://astronn.readthedocs.io/en/stable/galaxy10sdss.html (accessed Apr. 30, 2025).

[3] Ibrahim, Mohamed R., and Terry Lyons. "ImageSig: A signature transform for ultra-lightweight image recognition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

[4] K. Mukundan, P. Nair, J. Bailin, and W. Li, "Automating galaxy morphology classification using k-nearest neighbours and non-parametric statistics," *Monthly Notices of the Royal Astronomical Society*, vol. 533, no. 1, pp. 292–312, Sep. 2024

[5] Ma, Xiaohua, et al. “Galaxy image classification using hierarchical data learning with weighted sampling and label smoothing.” *Monthly Notices of the Royal Astronomical Society* 519.3 (2023): 4765-4