Star Galaxy Separation

Darshan Makwana
Dept. of Mechanical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
21d100003@iitb.ac.in

Tanmay Ganguli
Dept. of Mechanical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210100156@iitb.ac.in

Atharv Hardikar
Dept. of Mechanical Engineering
Indian Institute of Technology, Bombay
Mumbai, Maharashtra
210100034@iitb.ac.in

***Abstract*—Separating Stars from galaxies has been an important question in the field of cosmology since decades. However classifying them is quite challenging due to the similarity between stars and distant galaxies. In this project we provide a novel approach for the classification of stars from galaxies that may outrun traditional neural networks in asymptotic limits of image sizes. Numerical results shows that our model is efficient for some image projections. We also provide an explanation for the observations and also lay a building block for future research work**

I. Introduction

To distinguish between stars and Galaxy might seem easy to the naked eye. But, cosmological surveys like Dark Energy Survey (DES) are primarily interested in observing as many *distant* galaxies as possible, as these provide the most useful data for constraining cosmology. However,at such vast distances, both stars and galaxies begin to look like low-resolution point sources, making it difficult to isolate a sample of galaxies.

Thus correctly identifying low resolution point sources of light as stars or galaxies is crucial for any cosmological survey. In this project we attempt to use standard ML techniques to build a model for the required classification and compare the accuracy of the algorithms used. This challenge, known as ”star-galaxy separation” is a crucial step in any cosmological survey.

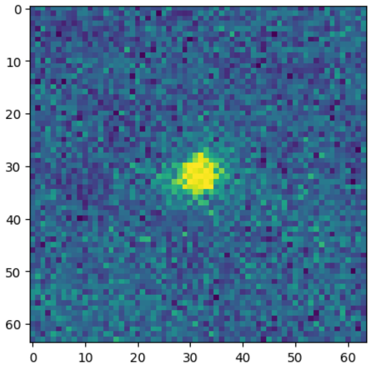
II. Data

The dataset used is taken from this https://www.kaggle.com/datasets/divyansh22/dummy-astronomy-data

The images were captured by the in-house 1.3m telescope of the observatory situated in Devasthal, Nainital, India. The original images captured were 2000x2000 in size which was reduced to 64x64 cutouts from the images to isolate the sources in a single image.

For labelling the images, image segmentation was used to identify the sources in the image, and finally the center coordinates of the found sources were queried with the SDSS database to give a label corresponding to each 64x64 cutout.

This dataset is generated from scratch using the real-world data. We are using this dataset to train and compare various machine learning models to classify stellar sources like stars and galaxies in the telescope images. The Processed data where each image is 64x64:



III. Methods

Data Preprocessing

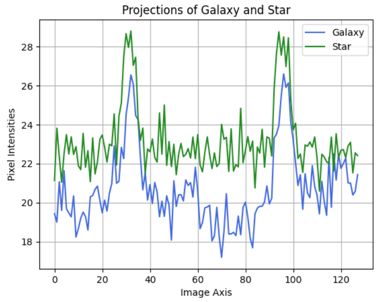
The two directories containing images of stars and galaxies were first imported.

With each image a list [x,y] was associated, x and y being 1x64 arrays, representing the projections of pixel intensities along x and y axes, i.e. sum of pixel intensities along each row and along each column respectively.

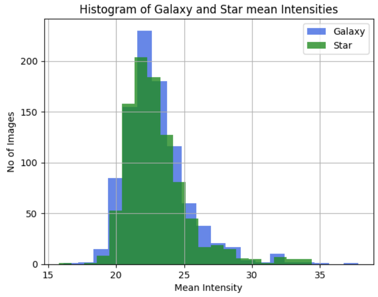
Rather than running the ML algorithms on the entire 64x64 array representing each image, they were run on the smaller two dimensional lists representing projections along x and y axes, as discussed above. This was done to reduce overall time and computational complexity while maintaining a decent accuracy.

Exploratory Data Analysis

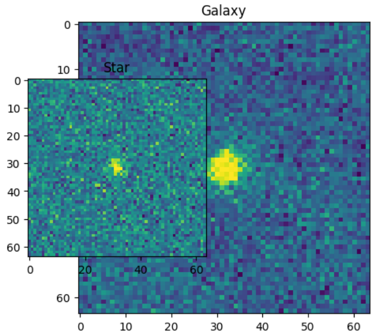
The sum of pixel intensities along each row and each column were plotted for stars and galaxies and their mean values were compared in a single plot.



The central parts of the images were brighter as compared to the fringes, indicating that the light source (galaxy/star) was more or less located near the center of the 64x64 image. Overall, galaxies seemed to be brighter than stars.



The star and galaxy images are compared below:

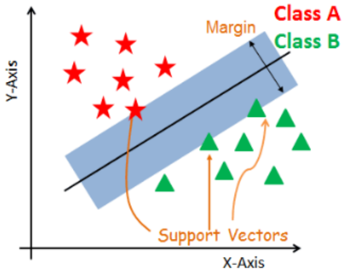


We have used ML as a blackbox, i.e. using predefined functions in sklearn module.

We have explored the following ML algorithms and their accuracies have been compared:

A. *SVM:*

Support Vector Machine or 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.



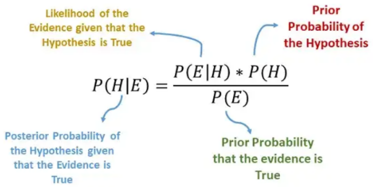
The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

The goal is to maximize the minimum distance

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{||w||_2}$$

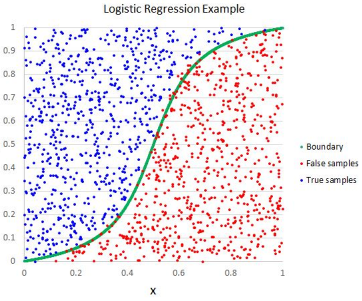
B. *Naive Bayes:*

Naive Bayes is a classification technique that is based on Bayes' Theorem with an assumption that all the features that predicts the target value are independent of each other. It calculates the probability of each class and then pick the one with the highest probability. It has been successfully used for many purposes, but it works particularly well with natural language processing (NLP) problems.


Likelihood of the Evidence given that the Hypothesis is True
Prior Probability of the Hypothesis
$P(H|E) = \frac{P(E|H) * P(H)}{P(E)}$
Posterior Probability of the Hypothesis given that the Evidence is True
Prior Probability that the evidence is True

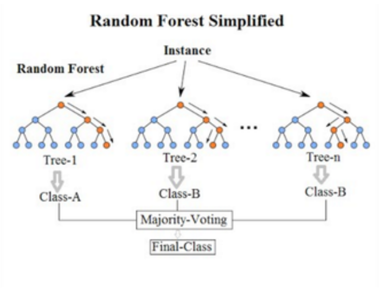
C. *Logistic Regression:*

This type of statistical model (also known as logit model) is often used for classification and predictive analytics. Logistic regression estimates the probability of an event occurring, such as voted or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1. In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure. This is also commonly known as the log odds, or the natural logarithm of odds, and this logistic function is represented by the following formulas:



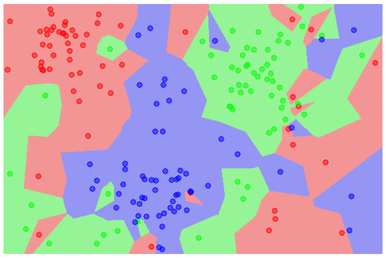
D. *Random Forest:*

Random forests are a supervised Machine learning algorithm that is widely used in regression and classification problems and produces, even without hyperparameter tuning a great result most of the time. It is perhaps the most used algorithm because of its simplicity. It builds a number of decision trees on different samples and then takes the majority vote if it's a classification problem. One of the biggest problems in machine learning is Overfitting. We need to make a generalized model which can get good results on the test data too. Random forest helps to overcome this situation by combining many Decision Trees which will eventually give us low bias and low variance.



*E. **KNN:***

KNN stands for K-nearest neighbour, it's one of the Supervised learning algorithm mostly used for classification of data on the basis how it's neighbour are classified. KNN stores all available cases and classifies new cases based on a similarity measure. K in KNN is a parameter that refers to the number of the nearest neighbours to include in the majority voting process. Where $K = \sqrt{n}$, where n is the total number of data points(if in case n is even we have to make the value odd by adding 1 or subtracting 1 that helps to select better.) We used KNN to distinguish between the classes of stars and galaxies by setting the parameter K as 2.



IV. Results

Algorithm Name	Accuracy in (%)
SVM	78.69
Regression	76.94
Gaussian Naive Bayes	68.29
Multinomial Naive Bayes	72.93
Random Forest	79.19
KNN	70.17

V. Discussions

As we had expected that the accuracy obtained by random forest is the greatest among all the ML models tested. We consider this to be true because of the it's ensemble learning method for classification in which the output of the random forest is the class selected by most trees.

VI. Conclusions

We found that we are able to achieve accuracy in classification of the images into stars and galaxies in the range of about 70-80 percent using several ML algorithms. Thus our approach of associating each image with the projections of pixel intensities along x and y axes is justified. It significantly reduces time and computation complexity, but not at the cost of accuracy.

Support Vector Machines and Random Forest Classifiers are the most accurate whereas Gaussian Naive Bayes and KNN are the least accurate in classification of the given data.

VII. Future Works

Hough Transform can also be explored as a classification technique. Although computationally heavy as compared to the ML techniques used, it can yield higher accuracy. Traditional Neural Networks can be used to increase the accuracy of the classification but the asymptotic limits of their time complexity would be much greater than the method implemented above.

VIII. References

[1] High-Level feature extraction: fixed shape matching, Mark S. Nixon, Alberto S. Aguado, in Feature Extraction and Image Processing for Computer Vision (Fourth Edition), 2020

[2] Pedregosa et al (2011). Scikit-learn: Machine Learning in Python. JMLR: 12, pp. 2825-2830.

[3] The 13th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine At: Newport, RI Volume: pp.448-452

[4] Soumagnac et al (2013). Star/galaxy separation at faint magnitudes: Application to a simulated Dark Energy Survey. arXiv:1306.5236