Big Data Analytics: Galaxy Classification

PSTAT 135 Final Project

Chuqiao Liu   Addison Luria-Roberson   Lauren Wong

University of California, Santa Barbara

Table of Contents

1. Abstract

Galaxies come in all shapes: from beautiful spirals to huge ellipticals. Understanding the types of galaxies as a function of shape is a critical piece for analyzing the evolution of the Milky Way. In order to classify galaxies into categories, we fit a Convolutional Neural Network (CNN) - Transfer Learning model to our data. After analyzing the training and validation accuracy and loss plots, as well as their precision and recall values, a two-class CNN model was chosen for our data.

1.1 Research Questions

1. Can regression models be used to classify images of spiral, edge, smooth, and otherly shaped galaxies?
2. Can the champion model be employed to determine the tightness of the arms of a spiral galaxy?

2. Data

The dataset is sourced from NASA's Sloan Digital Sky Survey and GalaxyZoo. The Sloan data consists of 61,578 images of galaxies and are sized 400 by 400 pixels. The GalaxyZoo data is a csv file of classifications of each Sloan image based on the responses of tens of thousands of GalaxyZoo users. Users answered questions on the shape of each galaxy and then answered questions depending on the responses to previous questions.

Image Data

The 61,578 images were all 400 by 400 pixels. Each image is centered around the center of the galaxy, star, or other object featured in the image. The object almost invariable occupied the middle 50% of the image. In other words, each object is roughly 200 by 200 pixels.

Classification Data

The classification dataset is 61,578 rows by 38 columns. The values of each variable represent the proportion users who answered yes to the corresponding question multiplied by the proportion of users who answered yes to the question which preceded. As such, for question 1, the highest possible value is 1. The highest value of each succeeding question is therefore limited by the highest proportion value of preceding questions.

2.1 Data Preparation

To prepare the dataset for use, the classification variables were reduced, the images resized, and the images were matched to a classification and separated into their own folder. These steps are outlined below:

Variable Transformations

The classification dataset originally featured 38 variables. Many of these variables had no relevance to our primary research questions which only concerns classification of galaxies into one of three shapes or other. To simplify the dataset, the file *Modify the CSV.ipynb* was created and uses 6 variables to create three categorical variables. These variables are named Smooth, Edge, and Spiral. A value of 1 was designated as a positive classification for that shape while all three variables equalling 0 indicating that the galaxy in question is of type “other”.

Image Transformations

In order to improve the efficiency of constructed models, each image was trimmed from 400 by 400 pixels to 300 by 300 pixels. This size was determined to be optimal due to the fact that all of the pictured objects were centered and did not take up the full size of the image. The cropped portions of the images contained the portion of space lying behind the object of the image. This manipulation was completed in the file *CropImages.ipynb.*

Image Separation

Due to the restrictions of the tools used to create the models, all images needed to be pre-sorted into folders based on classification. To accomplish this, the file *SplitImagesIntoFolders.ipynb* read in the modified CSV dataset, found the image corresponding to the ID of each row of the dataset, and moved the image into the appropriate folder.

2.2 Exploratory Data Analysis

With the data cleaned and sorted, exploratory data analysis was employed to determine the proportions of each galaxy shape in the dataset and find any other characteristics of note.

The dataset is 42% smooth galaxies, 25% spiral, 23% other objects, and 11% edge. This alerted us to the possibility that any models based on a random sampling of the data could be biased towards classifying images as smooth. Consequently, model 2 used an equal number of each image type when training the model to avoid bias.

The help visualize the images in aggregate, an average image was generated using five randomly selected images from each of the four classes. The resulting images were telling. The average smooth, spiral, and edge galaxies were nearly indistinguishable and appeared almost perfectly circular. The average edge image showed two cigar shapes at 90 degree angles to each other. This shows that of the four types, edge seems to be the most easily distinguishable.

3. Model Building

3.1. Model Selection

Although the research question was to see if we are able to utilize computer vision for image classification using regression models, our main goal became to find a model that was able to classify before proceeding to attempt regression methods. Convolutional Neural Network (CNN) Modeling proved to not only be the most popular modeling option for computer vision but also the most accurate and time efficient. We also preferred the CNN model due to its ability to reduce overfitting and provide high accuracy.

Following our modeling processes on CNN transfer learning, we proceeded to attempt regression methods, as outlined below, only to meet various obstacles on multiple ends, allowing us to conclude that CNN modeling is by far the best model for our data.

3.2. Model 1 - Transfer Learning

Transfer Learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. Utilization with CNN enables us to freeze the early convolutional layers of the network and only train the last few layers that make the prediction. Consequently, we believe this method to be the most efficient and optimal model for our initial testing and predictions.

Initially, due to the size and time factor of modeling our data, we decided to split our four classifications into two before opening up our model to the full dataset. For our initial 2-Class CNN model, we chose the galaxy classifications of edge and spiral due to their unique and relatively easily identifiable feature characteristics.

CNN's feature extraction method upon where the model reuses an existing neural network and only reclassifies and retrains on the new classification - in our case, spiral, edge, smooth, and other - was highly preferred and eliminated hours of time when training and testing our models. Furthermore, as another time saving method, we decided to partition 2,000 images from the set of 60,000, trained on 3,200, and tested on 800. By utilizing CNN, which reduces the images into a form that is easier to process, without losing features that are crucial to obtaining an accurate prediction, we were able to determine a better fit from the reduced number of parameters and reusability of weight.

In relation to our processes, we were able to utilize these layers to capture all necessary features such as edges, color, and gradient orientation before sending the data into the pooling process. After flattening our final outputs, we then were able to feed it into the regular neural network, testing on 30 epochs with 100 gradient updates for final classification.

Testing on 30 epochs took roughly 40 minutes and our accuracy rates exhibited a good model fit as the rates ranged from the mid 80th percentiles to the low 90th percentiles. By viewing our training and validation accuracy plot, we confirmed that we are not overfitting as both the train and validation accuracies trended closely to each other. In addition, because the accuracy continuously increased as the epoch increased, we were able to intuitively conclude that increasing the epoch size will likely give us an even higher accuracy. Similarly, analyzing the training and validation loss plot, further confirms against overfitting due to the training and validation losses trending down together. These results enabled us to intuitively conclude that the loss will likely decrease when increasing epoch size as well.

Therefore, based on our high accuracy rates from the 2-Class CNN model, we hope to achieve the same results when opening up our modeling to our full data.

3.3. Model 2 - Transfer Learning Expanded

Model 2 is based on Model 1 but is re-trained with all four classes. With twice the amount of data, extra image preprocessing was applied in an attempt to optimize the model building process. The main difference is that all images were converted to greyscale so as to cut the size of the data in half by eliminating two color channels. Like Model 1, Model 2 is a CNN Transfer Learning model. It was trained on 320 images and tested on 80 images. The model employed 20 epochs and was built slowly taking an average of 1.5 hours on each attempt.

Model 2 is significantly less accurate than Model 1. Despite its high training accuracy (91%), the model correctly predicted classification 22% of the time on average. Interestingly, the class with the highest accuracy is “other”, at an average of 33%.

There are several potential reasons for the severe underperformance of the model which are listed below:

Image Preprocessing

As part of the preparation for the model, all images were converted to greyscale. While this undoubtedly sped up the creation of the model, it’s possible that the filter deleted important information. Model 1 use color gradients and edge detection as variables and both methods are impossible or less accurate with grayscale images. The loss of information is likely compounded by the similarity of some classes. However, this same logic can explain the relatively high accuracy for the “other” class. While other tends to be stars or other circular object, some “other”s are image artifacts and lens flares meaning that their unique shape may have led to higher accuracy in the absence of color.

Similarity of Classes

As revealed in the exploratory data analysis, the smooth, spiral, and other galaxy types appear very similar. It is possible that Model 1 performed well because it tested edge vs one of the other three. Edge is distinct in shape due to its non-circularity.

3.4. Model 3 - Other Models

In order to optimize the classification method, we explored other methods including Logit Regression model, Decision Tree model and K-nearest Neighbors (KNN) models. After processing and gray-scaling large volumes of images from our datasets , we failed to flatten pixel matrixes and fit available algorithms.

4. Conclusion

Based on previous results, the 2-Class CNN model is the champion model of our data, which has approximately 90 percent high accuracy. The 4-Class CNN model performs high accuracy on training dataset and low accuracy on testing dataset. Neither could we decide if regression models can be used to classify images of spiral, edge, smooth, and otherly shaped galaxies nor could we employ the champion model to determine the tightness of the arms of a spiral galaxy. Those questions require further research.

4.1 Future Work

More work must be done to construct an accurate, four-class model. This can be done by not converting the images to grayscale and instead using Principal Component Analysis as an alternative method of reducing file size. Should this not improve the situation, a Logit or Naive-Bayes model can be generated and compared for relative accuracy. If any of these models prove viable, the final step is to re-train the model on the full set of images.

References

https://www.kaggle.com/c/galaxy-zoo-the-galaxy-challenge

https://towardsdatascience.com/image-detection-from-scratch-in-keras-f314872006c9?fbclid=IwAR30g14XE6YwOSbhx2zUl0P79GV5FO6pD51EbjTdB1OQ4O_JCh7Wu2scB7w

https://spark.apache.org/docs/2.3.0/api/python/_modules/pyspark/ml/image.html?fbclid=IwAR1GcsEN4fyB5vfbegwP4miXxIdef8f3IXXNzEiqC2BIfjwxUpEfzSEdOuA

https://databricks.com/blog/2018/12/10/introducing-built-in-image-data-source-in-apache-spark-2-4.html

https://github.com/FavioVazquez/deep-learning-pyspark/blob/master/SparkDL.ipynb

https://machinelearningmastery.com/display-deep-learning-model-training-history-in-keras/

https://learnandshare645.blogspot.com/2016/06/feeding-your-own-data-set-into-cnn.html?fbclid=IwAR2t1KDvhsI9vu-_qMwrrAGmuteeHrcFlxB9CwsEWPMtmmDiK8M5b8pOuM4

https://learnandshare645.blogspot.com/2016/06/feeding-your-own-data-set-into-cnn.html?fbclid=IwAR2t1KDvhsI9vu-_qMwrrAGmuteeHrcFlxB9CwsEWPMtmmDiK8M5b8pOuM4