Draft version December 1, 2020
Typeset using LaTeX **modern** style in AASTeX63

ASTP 720 Final Project: Logistical Regression

Isabella Cox – igc5972@rit.edu

1. INTRODUCTION

With the vast amounts of data available to use (e.g., from the Sloan Digital Sky Survey [SDSS]), we need to develop robust ways to mass-categorize and analyze all the data. One technique that can be used to separate different populations when considering two (or more) variables, is logistic regression.

Logistic regression is a special case of regression, where there is a binary label (0 or 1) variable determined based on the values of predictor variables. This is done by casting each data point to an outcome label by using the logistic function, discussed in the next section.

The applications of logistical regression are broad, and are well-suited to many astrophysical problems. One such application is to classifying galaxies "types" (spiral or elliptical) on the basis of their location on a color-mass diagram. The SDSS has morphology classifications from Galaxy Zoo for thousands of their observed galaxies. Figure 1 shows a color-mass diagram for SDSS galaxies. It is clear that there is a bimodality - two clear populations, one that more massive and redder, and the other that is less massive and blue-er. In the same paper this figure comes from, the authors correlated the points with Galaxy Zoo classifications, and found that most of the more massive, redder galaxies are elliptical galaxies and the blue-er galaxies are more often spiral galaxies (which makes sense in the scheme of galaxy evolution). In subsequent sections, I will describe how I predicted a classification label for each galaxy (elliptical or spiral) using logistical regression and then compared those classifications to the original ones each galaxy was given. First however, in the immediate following section, I will describe the data set I will work with and how I derived it.

2. DATA

SDSS has acquired spectra of millions of galaxies. And for a great subset of these galaxies, ancillary data exists, such as stellar mass and morphological classification (from Galaxy Zoo). These data sets can be very large and unwieldy to handle, so for a simplified setup, I simulated two populations of galaxies, a "blue cloud" and a diagonal red strip, which were assumed to be spiral and elliptical galaxies respectively. A color-mass plot for this sample is shown in Figure 2. The mechanics of the rest of the project is the same for this sample as it would be for a true observational sample

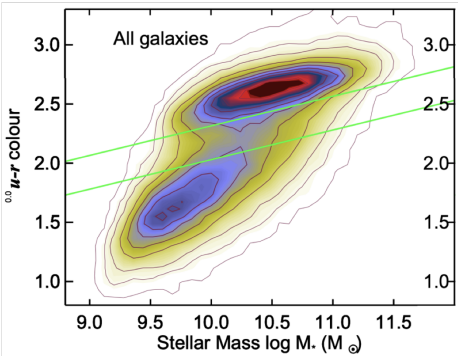

Figure 1: Color-mass diagram for SDSS galaxies, showing a clear bimodality in population. Image credit: Schawinski et. al, MNRAS 400.1, 2014.

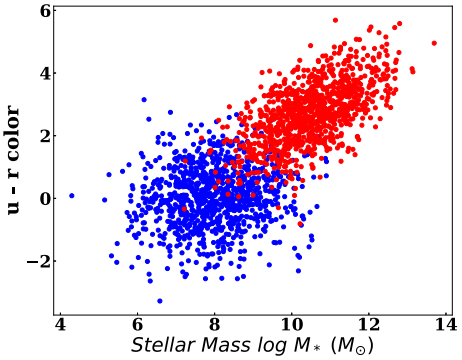

Figure 2: Color-mass diagram for sample of the pretend galaxies used for this project.

from SDSS, with the benefit of being lighter and easier to work with. (And yes, it is true that my simulated sample goes to much lower stellar masses then probable, but the basic principle of this work still holds).

3. IMPLEMENTATION OF CODE

3.1. *Hypothesis Function*

The hypothesis function for logistic regression is a sigmoid (logistic) function, as shown in Figure 3. The sigmoid function is an S-shaped curve that at extreme values of x, tends towards 0 (for smaller values of x) and 1 (for larger values of x). The functional form of the sigmoid is given below in Equation 1.

$$h(x) = \frac{1}{1 + e^{-x}} \tag{1}$$

We can exploit this form by using the sigmoid function to cast each galaxy to a binary extreme, either an elliptical or spiral galaxy on the basis of if the functional sigmoid value is greater than or equal to the midway point (0.5). So we use the sigmoid function to *tip a probability into one of the two discrete classes*. This will be further discussed in section 3.4.

3.2. *Cost Function*

Just like with linear regression, the goal of the cost function is to minimize the error. Since our hypothesis function is non-linear (which is different than the case of linear regression), we need to choose a special cost function. The most common choice is a cross-entropy cost function, also known as "log-loss." The equation for the cross-entropy cost function is given in Equation 2,

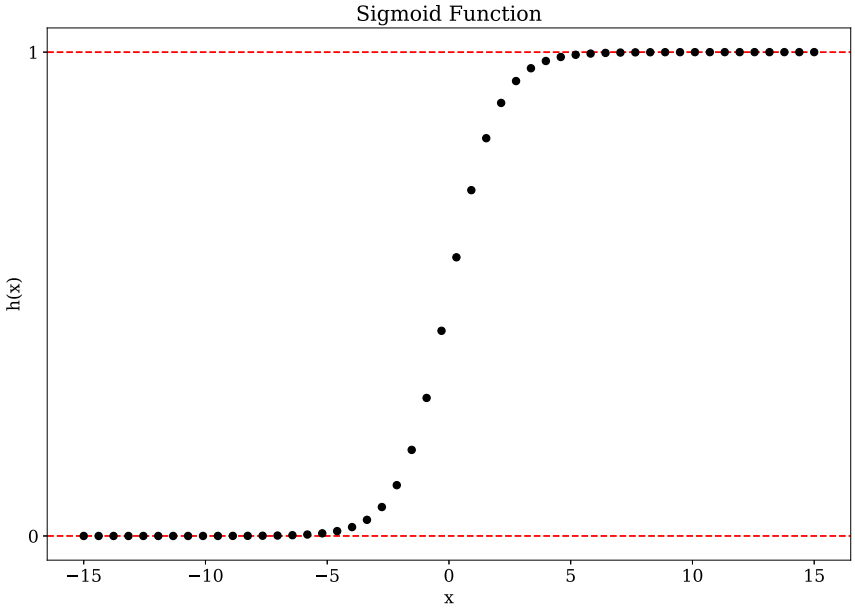

Figure 3: The sigmoid (logistical) function for values resulting in the functional values tending towards 0 and 1.

$$J(\theta) = \frac{-ylog[h(z)] - (1-y)log[1-h(z)]}{m} \tag{2}$$

where $h(z)$ is the hypothesis (sigmoid) function, m is the number of data points, y is the predicted binary label, and θ represents each of the parameters. This cost function ends up making sense because you can see it as the summation of two different parts, one for y = 1, and one for y = 0. Equation 2 can be split up for these two cases as,

$$J(\theta) = \begin{cases} -log[h(x)], & \text{if y} = 1. \\ -log(1-[h(x)]), & \text{if y} = 0. \end{cases} \tag{3}$$

where you can visualize that the cost would be reduced if h(x) tended towards 1 for the y = 1 case and reduced if h(x) tended towards 0 for the y = 0 case.

3.3. *Gradient Function*

The question becomes how we reduce the cost function to find the best values for our parameters. We will minimize our cost function by running gradient descent on each parameter. For a large number of iterations, we want to update theta by decreasing it by a learning rate, α, times the re-calculated gradient.

3.4. *Main Frame of Code*

The main body of the code (once the above functions were written) is to go through a large number of iterations, and for each step, calculate the current sigmoid functional value based on the current θ value, and then calculating the corresponding gradient rate. θ is then decreased by α times the gradient function.

Then, we can make predictions for each galaxy. We will calculate the sigmoid functional value (for the final θ value) and then will evaluate if that value is greater than 0.5 (cast to 1 [elliptical] classification) or less than 0.5 (cast to 0 [spiral] classification).

The θ values we are searching for are also known as the "weights" of the fit. There are three θ values, two for each feature (color and mass) and the third for the y-intercept.

4. RESULTS

The results from Figure 2 are replotted in Figure 4, which shows the decision boundary. The equation for the decision boundary line is

$$y = -(\theta_0 + x\theta_1)/\theta_2 \tag{4}$$

where the θ values are the final weights of the fit. The derivation of where this comes from is not very clear, but the basic idea is that at the decision boundary, both labels must be equally probable. And we can calculate the probability of either label using the logistic function, and the probability of the other label is 1 minus the aforementioned value. So by equating them, you find the condition that $\theta^T x = 0$, and from that, with our three values of θ and two values of x, you can solve for Equation 4.

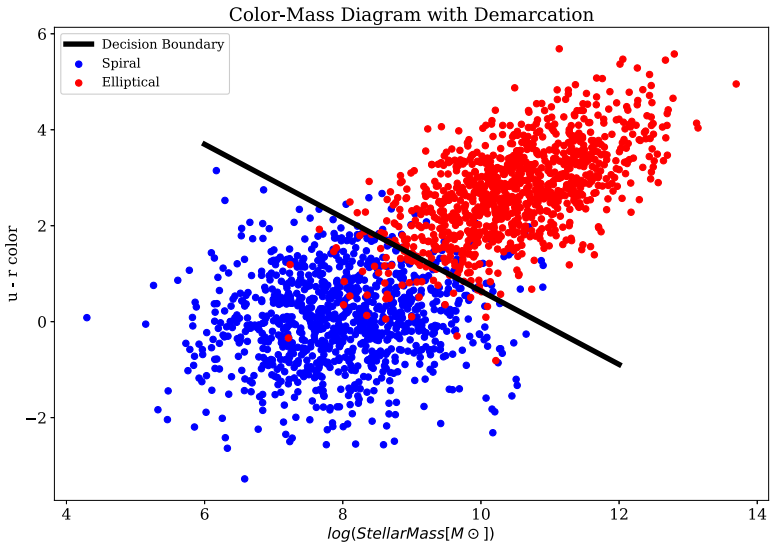

Figure 4: Figure 2 re-plotted with the decision boundary over-plot.

The outcome prediction labels can be compared to the prior known labels and visualized in a confusion matrix, shown in Figure 5. On the y-axis are the predicted labels and on the x-axis is the priorly known label. We can see that there was an over 95% success rate.

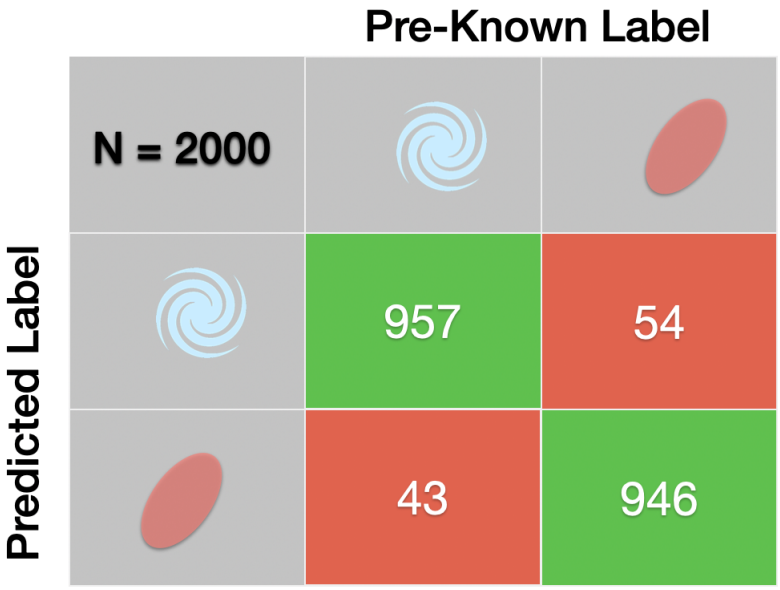

Figure 5: Confusion matrix.

REFERENCES

Chandrasekaran, Dinesh. "Logistic Regression from Scratch Using Python." (2019).

Jurafsky, Daniel. "Logistic Regression." (2019).

Schawinski, Kevin et al. "The green valley is a red herring: Galaxy Zoo reveals two evolutionary pathways towards quenching of star formation in early- and late-type galaxies". Monthly Notices of the Royal Astronomical Society 440. 1(2014): 889–907.

Ungar, Lyle. "Logistic Regression." (2020).