Analyzing Dermoscopic Images of Skin Lesions: A Logistic Regression Approach Using Spark

Dimas Molina
Halicioğlu Data Science Institute, UC San Diego

DSC232R: Big Data Analytics Using Spark
Dr. Edwin Solares
June 8, 2025

ABSTRACT

The American Cancer Society predicts that approximately 105,000 people in the United States will be diagnosed with skin cancer in 2025, making early detection of skin lesions paramount. This project explores the application of machine learning techniques for the classification of dermatoscopic images of skin lesions into three categories: seborrheic keratosis, nevus, and melanoma. The project will investigate the ML model of logistic regression. This project develops a classification framework that enhances diagnostic accuracy and contributes to early melanoma detection. This project aspires to advance the optimization of model architecture for medical image classification and increasing diagnostic reliability of automated tools, thereby supporting the effort to save lives through timely intervention.

INTRODUCTION

According to the Centers for Disease Control and Prevention (CDC), cancer was the second leading cause of death in the United States in 2022. The best way to combat this horrid disease is by catching it early, and part of that process includes getting screened. For melanoma – one of the deadliest forms of skin cancer – the screening process includes observing skin lesions and determining if they are of malignant origin. Dermoscopy analyzes such lesions under a microscope to obtain the determining origin and machine learning can be of great assistance in the process. I underwent this project because the topic of cancer hits close to home and the idea of contributing to timely intervention and giving people a second chance at life is incredible. By having a good predictive model, we contribute to catching skin cancer early and allowing patients to receive treatment as soon as possible.

METHODS

The dataset used in this report was obtained through Kaggle, a data science platform that hosts data science competitions and hundreds of thousands of public datasets.

Data Exploration

I obtained and utilized a 12.2 GB melanoma detection dataset of images that contain 3 different lesion types: nevus, seborrheic keratosis, and melanoma. The images, all of which are in ".jpg" format, were already split into a training set, a test set, and a validation set, which contained 2000, 600, and 150 images, respectively. The training set had 374 images of melanoma, 1372 images of nevus, and 254 images of seborrheic keratosis, totaling 2000 images in the training set. The test set had 117 images of melanoma, 393 images of nevus, and 90 images of seborrheic keratosis, totaling out to 600 images in the test set. The validation set had

30 images of melanoma, 78 images of nevus, and 42 images of seborrheic keratosis, totaling out to 150 images in the validation set. The sizes of these images were not uniform; over the entire dataset, the minimum height was 540 pixels and the minimum width was 576 pixels. The maximum height was 4499 pixels and the maximum width was 6748 pixels. Below in Figure 1 are dermatospopic images taken from the original training set, labeled by the nature of the displayed skin lesion:

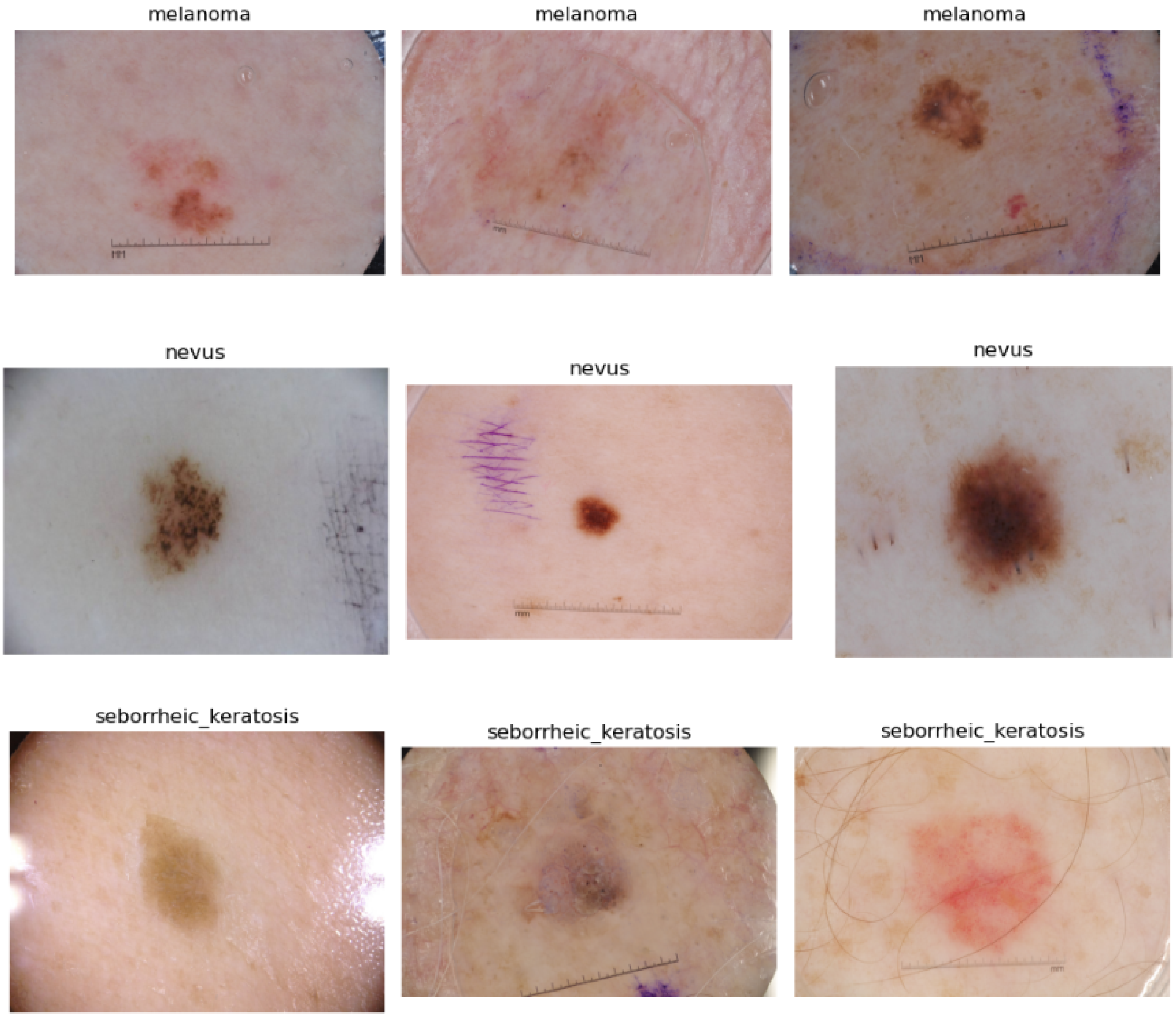

Figure 1: Three images each of melanoma, nevus, and seborrheic keratosis from the dataset's training data

Pre-processing

The images were of various different sizes, as aforementioned. Therefore, the images were shrunk to a target size of 64 by 64 pixels, in order to reduce computational load. The images were processed and converted into RGB form and into arrays. The original data arrived in 9 different folders named by lesion type, which served as implicit class labels. Once the images were transformed into arrays, the label for each image was obtained by extracting the folder name from the path of the image and then using those categories to create a new variable called "labels" into our data frames. Following that, the labels were converted into numerical representations and that variable was placed into a new indexed dataframe, which would be used for model training and fitting. Nevus was assigned a label index of 0, melanoma was assigned a label index of 1, and seborrheic keratosis was assigned a label index of 2. I changed the split of the training, test, and validation sets and evened out the proportion of the testing and validation sets in the process. Approximately 70% of the data was placed into a new training set, and the remaining validation and test sets each contained approximately 15% of the data. Each of these sets were then constructed to be their own dataframe.

Model 1

I trained a multinomial logistic regression model. The aforementioned training, test, and validation data frames were converted to vectors, using only raw pixel values as features. This is what would be fed into the model. With the "features" and "label index" variables of our indexed data frame, I defined a logistic regression model that ran on a maximum of 20 iterations and an

L1 regularization parameter of 0.1. Then the training set was fitted into the logistic regression model and used a multiclass evaluator from the PySpark ML classification library to analyze the accuracy of the training and test sets.

RESULTS

When the model was trained and the accuracies were analyzed, the training accuracy was approximately 72.6% and the test accuracy was approximately 66.7%. Based on various numbers of iterations, the model fitted our data in either an overfitting or underfitting manner.

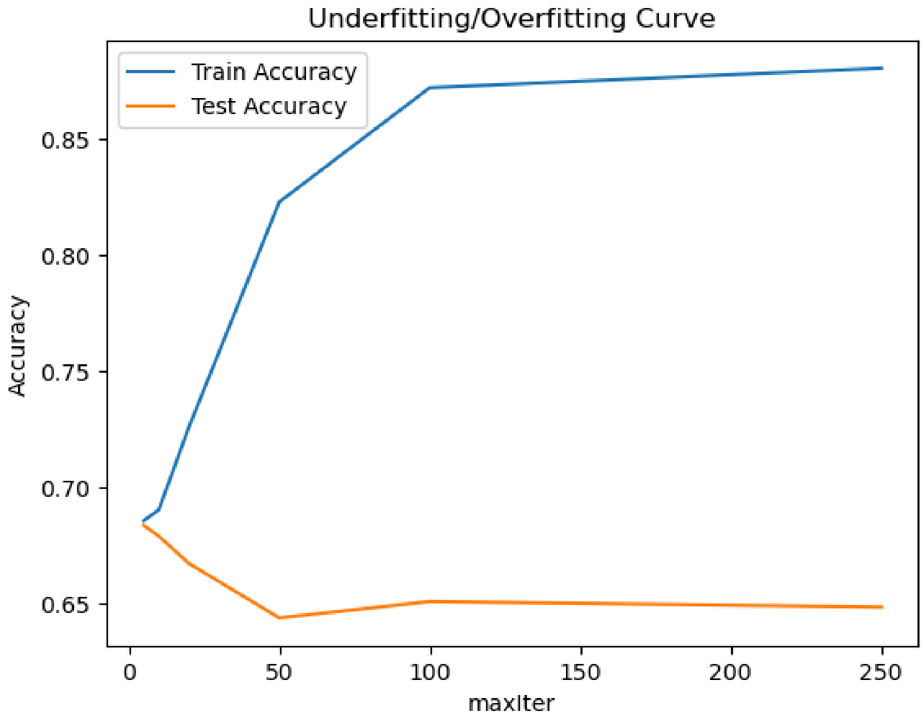

Figure 2: A graph representing the number of max iterations versus training and test accuracy

Figure 2 demonstrates how the accuracy on the training and test sets fluctuated over 5, 10, 20, 50, 100, and 250 iterations. As the training accuracy continuously increased, the test accuracy decreased and plateaued when going past 50 iterations. The model returned a numeric label for the vector of an image. In Figure 3 below, we analyze the efficiency of the model, from the viewpoint of the images:

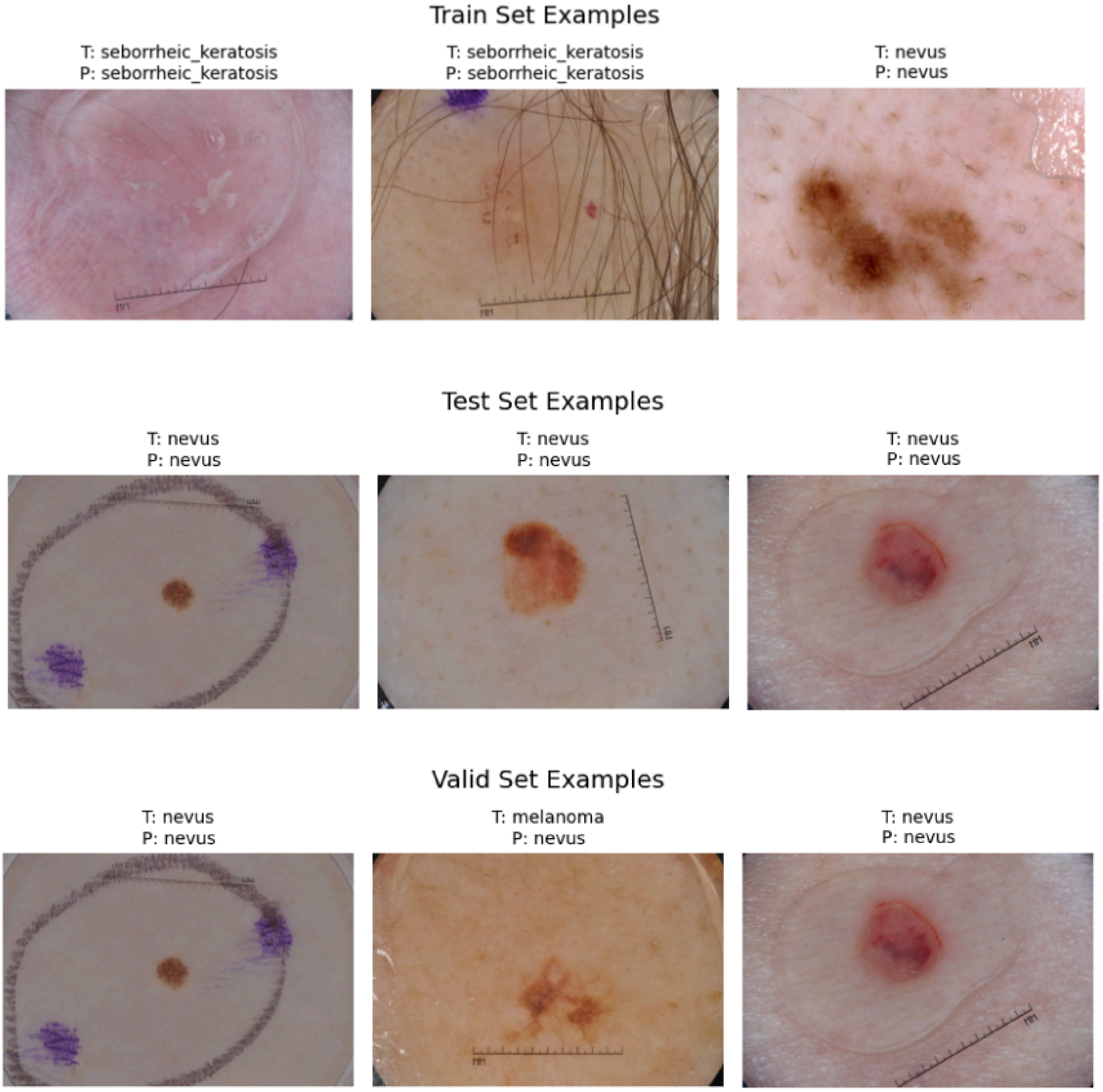

Figure 3: Example model outputs (labeled P for "predicted") and their true labels (labeled T for "true")

Due to timing circumstances and issues with the San Diego SuperComputer (SDSC), which was utilized for completing this project, I was unable to complete any more training of different models. My final model is the logistic regression model on raw pixels and at the optimal number of iterations of 20, which was suggested by the graph in Figure 2, the model yielded a 72.6% training accuracy and a 66.7% test accuracy.

DISCUSSION

Data Exploration

When going over the images, two major aspects of the entire dataset stood out. The first aspect I noticed was how the images were already split into training, validation, and test sets. The splits among the three sets were not as conventional as I thought they would be, namely with the validation set only accounting for 3.6% of the dataset. This guided my motivation for creating a new split of the three sets, which will be discussed later. The second aspect I noticed was how the images were not uniform in size. As aforementioned in the “Data Exploration” subsection of the “Methods” section, the smallest images had dimensions around 540 to 576 pixels, whereas the larger images were nearly 12 times larger. This would complicate the task of fitting image data into any model.

Pre-processing

The target size for the images to be uniform in size became 64 by 64 pixels. Reducing the target size to 64 by 64 pixels led to loss of color and texture cues, which are critical for classification. However, the reason for not increasing the target size was because whenever I did raise the target size (i.e. 128 by 128 pixels), the computer crashed whenever the images were being trained into the logistic regression model. Keeping the images to such a small size not only saved time on training the model, but also computer drive and Spark executor memory. Memory constraints limited higher-resolution training; future work should optimize resource allocation to enable larger input sizes.

When it came to splitting the dataset into the training, test, and validation sets, I knew I wanted more representation among the validation set, because this could present an opportunity for my model to perform better when it was already at a disadvantage with the minute uniform size. One aspect I fell short on addressing was how much representation there was of the skin lesion of nevus in the dataset. Although the representation was incredibly high, I should have taken into consideration if the reason for such high representation was because that is how common it is among people in the real world. The imaging model would be a useful real-world tool that can be used by imaging practitioners, and I believe the model should have reflected the proportions of how common nevus, melanoma, and seborrheic keratosis are among people, depending on the context in which the model was being applied.

Model 1

While logistic regression provided a reasonable baseline of approximately 67% test accuracy, more sophisticated models (i.e. CNNs) are expected to improve performance. Such models would be more appropriate for imaging tasks, as they would not compromise as much of the crucial aspects of the images compared to the methods used in this project. The logistic regression model analyzed the small images to the best of its ability, but it ultimately serves best as a simple debugging baseline.

CONCLUSION

This project underscores the need for robust image-based classification in dermatoscopy. Such a model would be heavily relied upon by not just imaging practitioners, but also doctors and their patients. It is of utmost importance to utilize the image analysis libraries made available in a responsible manner. Doing so could not just save many lives, it can also promote timely intervention by everyone involved. Given the logistic regression model is my final model, I do wish I could have found a way to resize the images in a way that did not compromise their integrity, texture, and structure. I would have ventured into the use of principal component analysis (PCA) and its advantages. This could have allotted space for resizing the images to a larger size or perhaps to run more iterations on my model without concern for the executor memory crashing.

STATEMENT OF COLLABORATION

Dimas Molina: I was the sole contributor: I wrote all the code and maintained the GitHub repository, which you can follow below:

https://github.com/d25molina/DSC232R_DFMProject/tree/main