

I. Introduction To Your Project

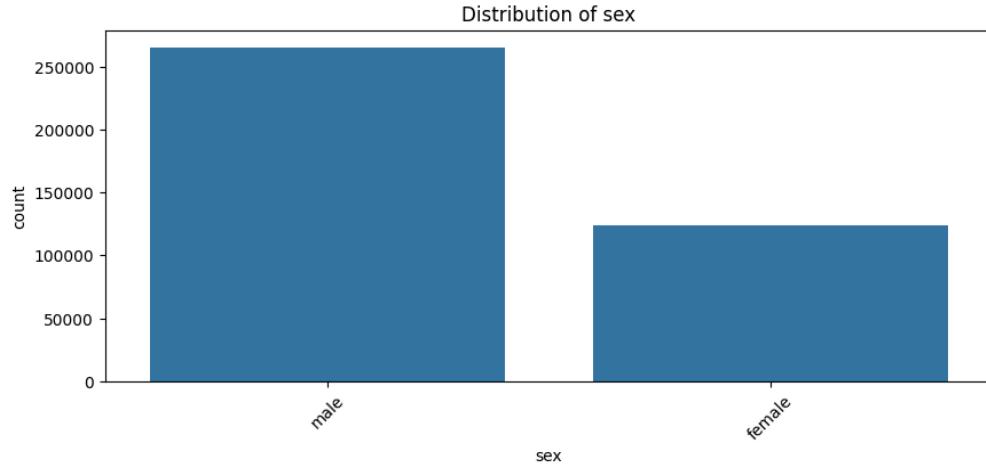
- A. This project is particularly important due to the fact that it addresses the early detection of skin cancer. This disease can have life-threatening consequences for many people, but, when detected early, has a much higher chance of treatment/recovery. A model that can utilize minimally-invasive, accessible data like a photograph to distinguish benign/malignant tumors would be an extremely useful tool for the diagnosis process. Quick and reliable results would become widely accessible to healthcare providers and greatly improve rates of early detection.

II. Methods Section

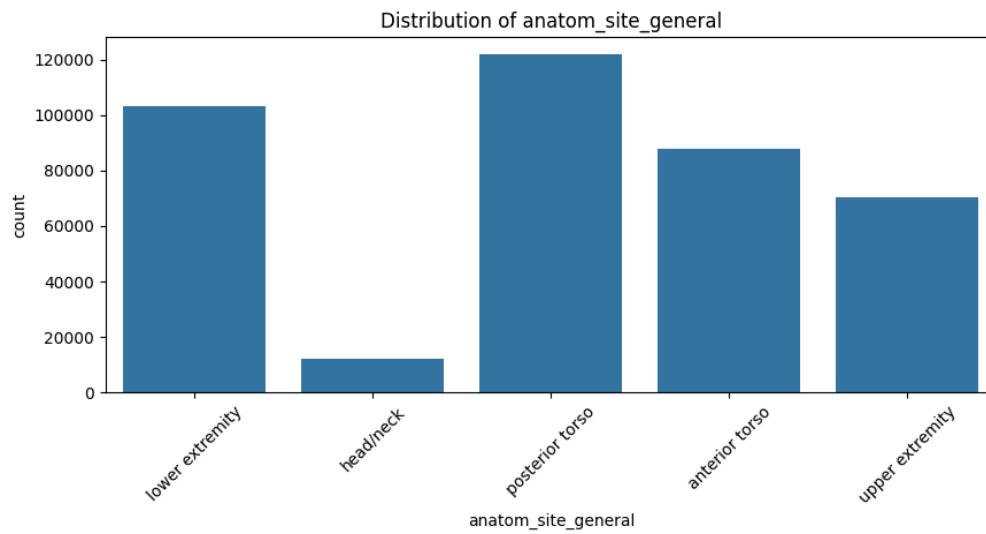
A. Data Exploration

Our initial data exploration focused primarily on the metadata associated with the images. This approach allows us to survey the full dataset in terms of feature analysis without the need to parse and analyze the features of all the images. We will be able to identify trends in features we expect to analyze as well as provide us with an understanding of biases that may exist in the dataset.

To begin with, we plotted the distributions of the most critical features to understand the existing trends or biases. An immediate observation was a significant gender imbalance, with notably more males than females in the dataset. This imbalance, however, is likely irrelevant towards the analysis of our dataset for the purpose of diagnosis.

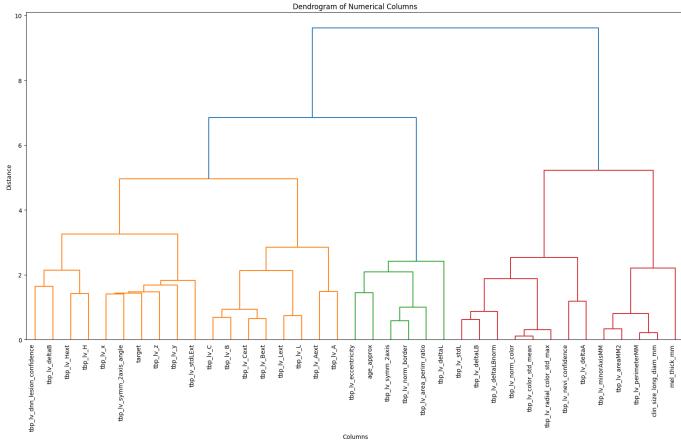
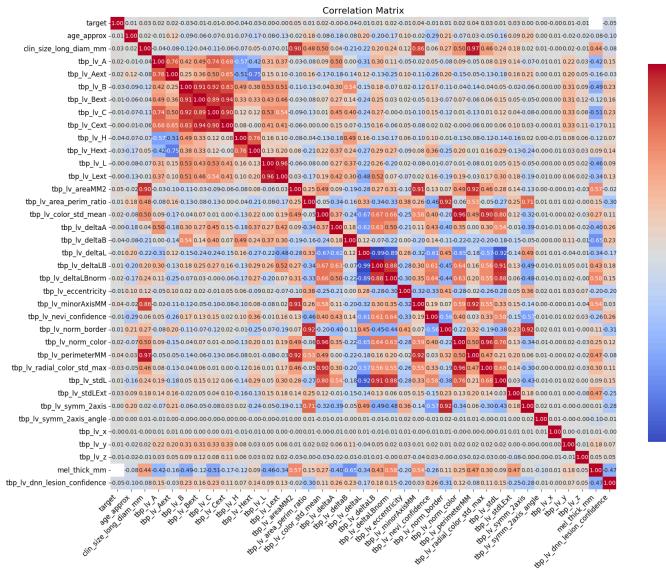


Furthermore, we observed that most samples in the data are from regions outside the head. This indicates that the head is underrepresentation might pose a challenge for our model in predicting lesions in the head area. We will have to experiment with hyperparameters and data stratification to see if we can address biases like this.



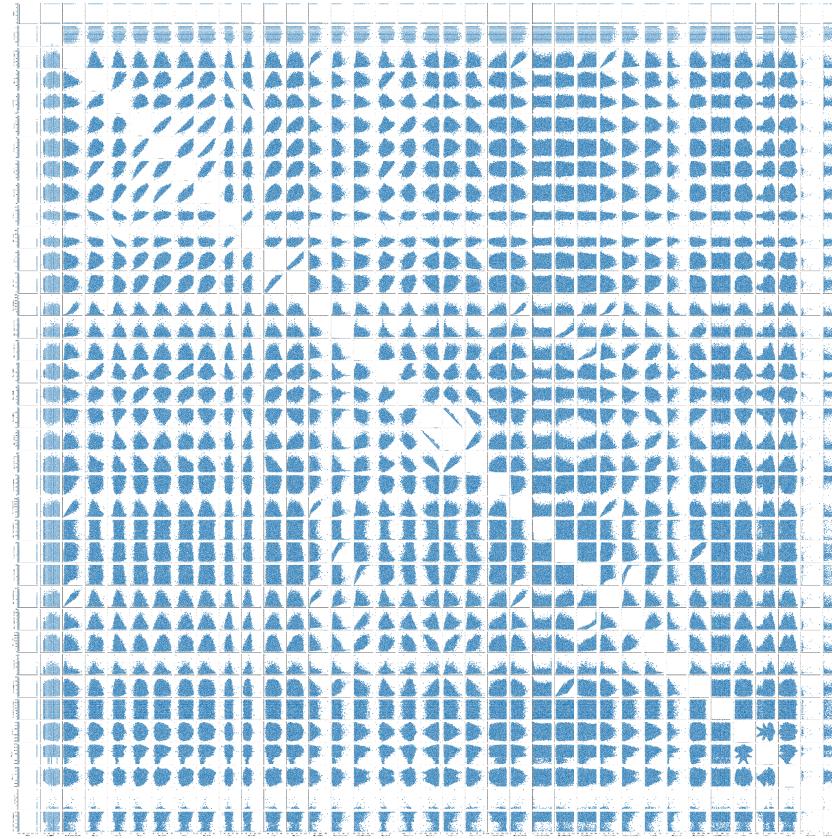
Additionally, the correlation matrix and corresponding dendrogram showed us closely related traits. While this information was not particularly useful in the overall design of our model as we would not be able to extract many of these metrics as independent features (like thickness, diameter, perimeter, etc.), it still showed us

what biological traits were closely related. It also gave us insight into what features might be worth focusing on or prioritizing our model towards.



The pairplot further elaborates on this information by showing us which biological features are most correlated with one another. If a lot of strongly weighted features are strongly correlated or if a lot of weakly weighted features are strongly correlated, it could hint at some relevant features we can artificially weight with stratification or pre-processing (i.e. if the color data is found to be relatively irrelevant,

pre-processing the data to use grayscale would help prevent fitting the model to data that could throw it off).



B. Preprocessing

1. **Hair Removal** – For image preprocessing, we tried to implement a hair removal algorithm consisting of Grayscale Morphological Closure, identifying thin and long structures which are the individual hair strands, and bilinear interpolation. Although we did not get a working version of this preprocessing technique, we wanted to try and implement it because it could have increased the accuracy for our model by using cleaner images. For Grayscale Morphological Closure, our goal was to convert the image to grayscale and apply a morphological closure operation in order for us to enhance the darker regions of the image which in this case would correspond to hairs. Then, to identify thin and long structures corresponding to the hair follicles, we created a binary image and identified selected regions corresponding and

filtered these according to aspect ratio in order to get the hair strands. Lastly, the bilinear interpolation involves creating a mask for the hair pixels that we have found, and then using the areas surrounding this to populate and replace the areas with hair. The ideal end result would be an image with all the hair removed so that our model is able to run on a completely clean image in order to see whether the lesion is malignant or benign.

Unfortunately, we did not get a fully functioning version of this due to time constraints.

C. Resnet

1. For our initial model, we utilized a pretty simple preprocessing step (conversion to a PIL, resizing to ResNet's preferred resolution, transform to tensor + normalization) and the Resnet18 pre-trained model. We wanted to use this as a baseline for future iterations and Resnet18 was a good starting point as it was a well-regarded model with a wide pre-training dataset, strong benchmarks, generalized nature, and efficient runtime. More specific/niche models could be implemented based on the observed performance here.

D. Logistic with Polynomial Features

1. We did a logistic regression using polynomial features, one with a single feature 'tbp_lv_color_std_mean' and another with two features 'tbp_lv_color_std_mean', 'age_approx'. We used these features and we speculated that these features could be the most influential in determining whether a lesion was benign or cancerous.

E. Logistic

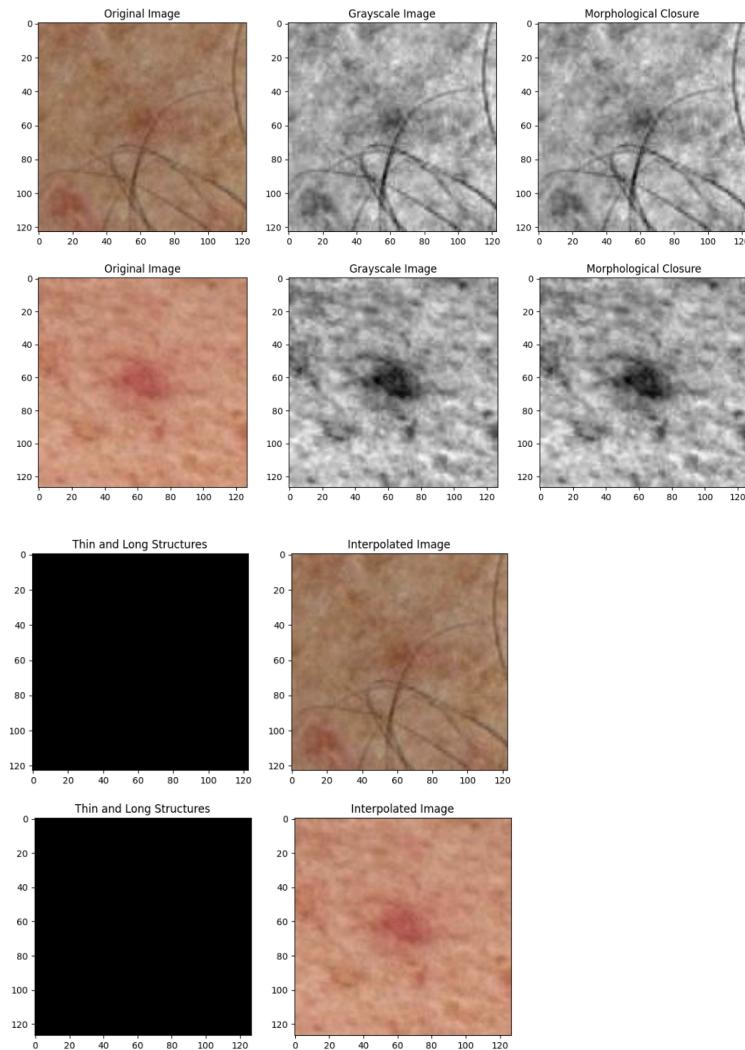
1. We did a single logistic regression with the two features 'tbp_lv_area_perim_ratio' and 'clin_size_long_diam_mm'. We chose these features because we speculated that there could be a relationship between these two features, which could then be used to determine if the lesion was benign or cancerous. Out of the models we covered in class, we believed that a logistic regression was a straightforward method of binary classification (as opposed to an ordinary polynomial or linear regression).

III. Results Section

A. Data Exploration

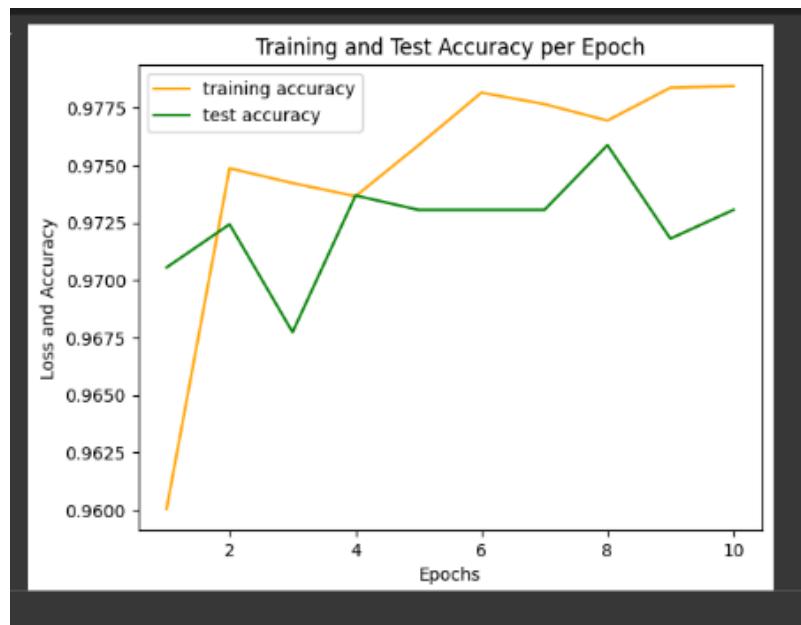
B. Preprocessing

1. Hair Removal



As you can see above, we were not successfully able to complete the hair removal preprocessing as we are able to get a grayscale and morphological closure, but ran into issues with identifying the thin and long structures of the image and returning the proper interpolated image without any hair follicles.

C. Resnet18



```

Σ Epoch [1/10], Loss: 0.1314, Accuracy: 0.9624
    Validation Loss: 0.1063, Accuracy: 0.9734
    Epoch [2/10], Loss: 0.0936, Accuracy: 0.9738
    Validation Loss: 0.1292, Accuracy: 0.9696
    Epoch [3/10], Loss: 0.0815, Accuracy: 0.9762
    Validation Loss: 0.0865, Accuracy: 0.9727
    Epoch [4/10], Loss: 0.0783, Accuracy: 0.9764
    Validation Loss: 0.0876, Accuracy: 0.9712
    Epoch [5/10], Loss: 0.0863, Accuracy: 0.9736
    Validation Loss: 0.0850, Accuracy: 0.9727
    Epoch [6/10], Loss: 0.0711, Accuracy: 0.9786
    Validation Loss: 0.0930, Accuracy: 0.9718
    Epoch [7/10], Loss: 0.0768, Accuracy: 0.9753
    Validation Loss: 0.0891, Accuracy: 0.9731
    Epoch [8/10], Loss: 0.0711, Accuracy: 0.9775
    Validation Loss: 0.0828, Accuracy: 0.9693
    Epoch [9/10], Loss: 0.0748, Accuracy: 0.9761
    Validation Loss: 0.0798, Accuracy: 0.9724
    Epoch [10/10], Loss: 0.0626, Accuracy: 0.9791
    Validation Loss: 0.0818, Accuracy: 0.9756

```

```

summary(model,input_size=(3,224,224))

Layer (type)          Output Shape       Param #
=====
Conv2d-1              [-1, 64, 112, 112]      9,468
BatchNorm2d-2          [-1, 64, 112, 112]      128
ReLU-3                [-1, 64, 112, 112]      0
MaxPool2d-4            [-1, 64, 56, 56]      0
Conv2d-5              [-1, 64, 56, 56]      36,864
BatchNorm2d-6          [-1, 64, 56, 56]      128
ReLU-7                [-1, 64, 56, 56]      0
Conv2d-8              [-1, 64, 56, 56]      36,864
BatchNorm2d-9          [-1, 64, 56, 56]      128
ReLU-10               [-1, 64, 56, 56]      0
BasicBlock-11          [-1, 64, 56, 56]      0
Conv2d-12              [-1, 64, 56, 56]      36,864
BatchNorm2d-13          [-1, 64, 56, 56]      128
ReLU-14               [-1, 64, 56, 56]      0
Conv2d-15              [-1, 64, 56, 56]      36,864
BatchNorm2d-16          [-1, 64, 56, 56]      128
ReLU-17               [-1, 64, 56, 56]      0
BasicBlock-18          [-1, 64, 56, 56]      0
Conv2d-19              [-1, 128, 28, 28]     73,728
BatchNorm2d-20          [-1, 128, 28, 28]     256
ReLU-21               [-1, 128, 28, 28]     0
Conv2d-22              [-1, 128, 28, 28]     147,456
BatchNorm2d-23          [-1, 128, 28, 28]     256
Conv2d-24              [-1, 128, 28, 28]     8,192
BatchNorm2d-25          [-1, 128, 28, 28]     256
ReLU-26               [-1, 128, 28, 28]     0
BasicBlock-27          [-1, 128, 28, 28]     0
Conv2d-28              [-1, 128, 28, 28]     147,456
BatchNorm2d-29          [-1, 128, 28, 28]     256
ReLU-21               [-1, 128, 28, 28]     0
Conv2d-30              [-1, 128, 28, 28]     147,456
BatchNorm2d-31          [-1, 128, 28, 28]     256
ReLU-32               [-1, 128, 28, 28]     0
BasicBlock-33          [-1, 128, 28, 28]     0
Conv2d-34              [-1, 256, 14, 14]     294,912
BatchNorm2d-35          [-1, 256, 14, 14]     512
ReLU-37               [-1, 256, 14, 14]     0
Conv2d-38              [-1, 256, 14, 14]     589,824
BatchNorm2d-39          [-1, 256, 14, 14]     512
Conv2d-40              [-1, 256, 14, 14]     32,768
BatchNorm2d-41          [-1, 256, 14, 14]     512
ReLU-42               [-1, 256, 14, 14]     0
BasicBlock-43          [-1, 256, 14, 14]     0
Conv2d-44              [-1, 256, 14, 14]     589,824
BatchNorm2d-45          [-1, 256, 14, 14]     512
ReLU-46               [-1, 256, 14, 14]     0
Conv2d-47              [-1, 256, 14, 14]     589,824
BatchNorm2d-48          [-1, 256, 14, 14]     512
ReLU-49               [-1, 256, 14, 14]     0
BasicBlock-50          [-1, 256, 14, 14]     0
Conv2d-51              [-1, 512, 7, 7]      1,179,648
BatchNorm2d-52          [-1, 512, 7, 7]      1,024
ReLU-53               [-1, 512, 7, 7]      0
Conv2d-54              [-1, 512, 7, 7]      2,359,296
BatchNorm2d-55          [-1, 512, 7, 7]      1,024

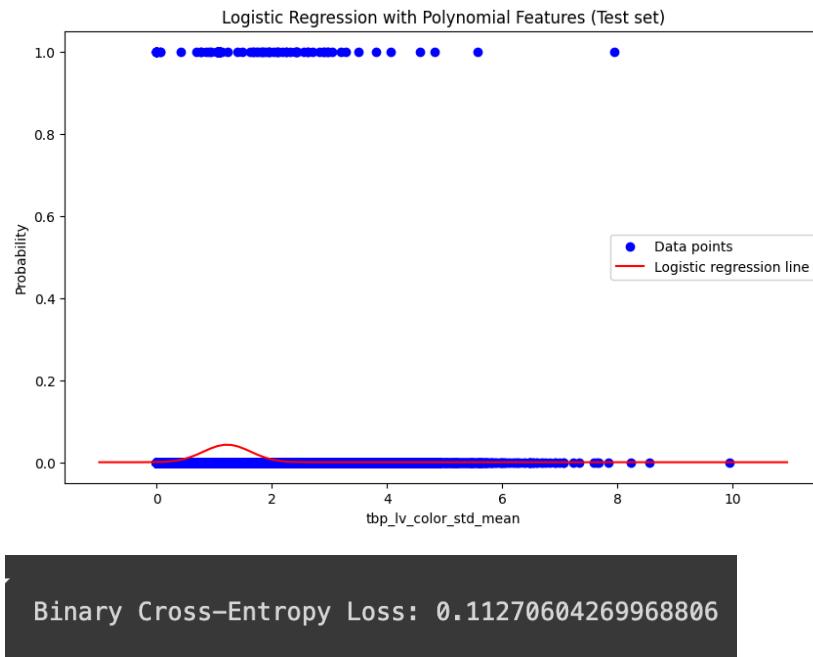
```

[] # Training loop

	Shape	Count
ReLU-21	[-, 128, 28, 28]	0
Conv2d-22	[-, 128, 28, 28]	147,456
BatchNorm2d-23	[-, 128, 28, 28]	256
Conv2d-24	[-, 128, 28, 28]	8,192
BatchNorm2d-25	[-, 128, 28, 28]	256
ReLU-26	[-, 128, 28, 28]	0
BasicBlock-27	[-, 128, 28, 28]	0
Conv2d-28	[-, 128, 28, 28]	147,456
BatchNorm2d-29	[-, 128, 28, 28]	256
ReLU-29	[-, 128, 28, 28]	0
Conv2d-31	[-, 128, 28, 28]	147,456
BatchNorm2d-32	[-, 128, 28, 28]	256
ReLU-33	[-, 128, 28, 28]	0
BasicBlock-34	[-, 128, 28, 28]	0
Conv2d-35	[-, 256, 14, 14]	294,912
BatchNorm2d-36	[-, 256, 14, 14]	512
ReLU-37	[-, 256, 14, 14]	0
Conv2d-38	[-, 256, 14, 14]	589,824
BatchNorm2d-39	[-, 256, 14, 14]	512
Conv2d-40	[-, 256, 14, 14]	32,768
BatchNorm2d-41	[-, 256, 14, 14]	512
ReLU-42	[-, 256, 14, 14]	0
BasicBlock-43	[-, 256, 14, 14]	0
Conv2d-44	[-, 256, 14, 14]	589,824
BatchNorm2d-45	[-, 256, 14, 14]	512
ReLU-46	[-, 256, 14, 14]	0
Conv2d-47	[-, 256, 14, 14]	589,824
BatchNorm2d-48	[-, 256, 14, 14]	512
ReLU-49	[-, 256, 14, 14]	0
BasicBlock-50	[-, 256, 14, 14]	0
Conv2d-51	[-, 512, 7, 7]	1,179,648
BatchNorm2d-52	[-, 512, 7, 7]	1,024
ReLU-53	[-, 512, 7, 7]	0
Conv2d-54	[-, 512, 7, 7]	2,359,296
BatchNorm2d-55	[-, 512, 7, 7]	1,024
Conv2d-56	[-, 512, 7, 7]	131,872
BatchNorm2d-57	[-, 512, 7, 7]	1,024
ReLU-58	[-, 512, 7, 7]	0
BasicBlock-59	[-, 512, 7, 7]	0
Conv2d-60	[-, 512, 7, 7]	2,359,296
BatchNorm2d-61	[-, 512, 7, 7]	1,024
ReLU-62	[-, 512, 7, 7]	0
Conv2d-63	[-, 512, 7, 7]	2,359,296
BatchNorm2d-64	[-, 512, 7, 7]	1,024
ReLU-65	[-, 512, 7, 7]	0
BasicBlock-66	[-, 512, 7, 7]	0
AdaptiveAvgPool2d-67	[-, 512, 1, 1]	0
Linear-68	[-, 1]	513
<hr/>		
Total params:	11,177,025	
Trainable params:	11,177,025	
Non-trainable params:	0	
<hr/>		
Input size (MB):	0.57	
Forward/backward pass size (MB):	62.79	
Params size (MB):	42.64	
Estimated Total Size (MB):	106.00	

D. Logistic with Polynomial Features

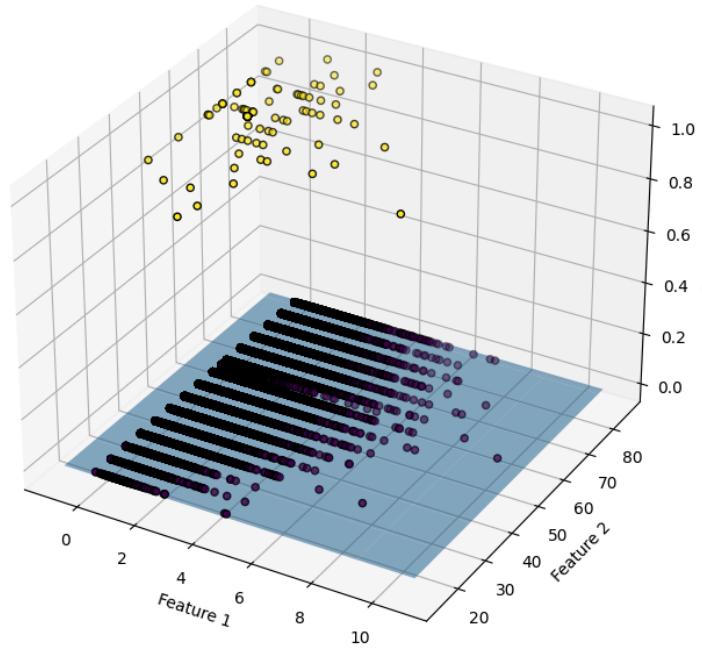
1. Single feature binary cross-entropy loss



2. Double feature Binary Cross-Entropy Loss

```
Binary Cross-Entropy Loss: 0.10194112247226822
```

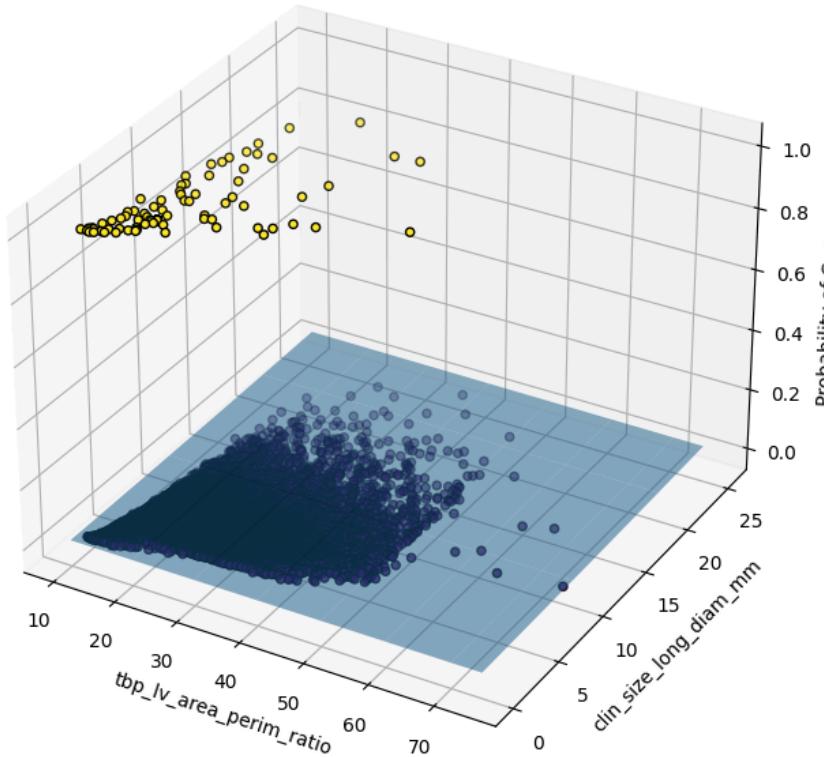
Logistic Regression with Polynomial Features (Test set)



3.

E. Logistic

Logistic Regression (Test set)



```
Cross-validation accuracy scores: [0.98481261 0.98481261 0.98481261 0.98481261 0.9848124 ]  
Average cross-validation accuracy: 0.9848125648239933
```

```
Binary Cross-Entropy Loss: 0.07791468318890882
```

IV. Discussion Section

- A. Preprocessing – Although we attempted to do a hair removal image preprocessing step, we were not able to get it fully working. I think one of the issues we ran into when trying to do this process was that for identifying things and long structures and bilinear interpolation we needed a mask and mask data. This was confusing because we did not have access to this data and were a little lost in how to proceed. If done correctly, the preprocessing file would be able to remove any hairs that exist on the image, especially the lesion portion, which could hinder our model efficacy to some extent.
- B. Logistic regression; double feature – Since our ultimate goal was to detect cancer based on images of skin lesions, it made sense to use a

simple logistic regression for classification. More specifically, we wanted to use features that might have the most influence on whether the lesion was benign or cancerous, so we chose the total lesion area to perimeter ratio and longest diameter (mm) as they also give strong visual clues for the model to learn from. Establishing a baseline using a straightforward model like a logistic regression also allows us to compare the results of more complex models. We used two measurements for accuracy: cross-validation and binary cross-entropy loss. We wanted to reduce the variance associated with a single train-test split, so using cross-validation provided us with a more robust estimate of the model's performance. Furthermore, we wanted to ensure that the model doesn't confidently give a wrong prediction, so by using binary cross-entropy loss we were able to penalize the model when it made an incorrect prediction. In regards to the cross-validation specifically, the consistent results indicate that the model did not overfit to a particular subset of the data, which is ideal for unseen data. However, given that logistic regression assumes a linear relationship between the features, it is possible that the two features chosen interact in nonlinear ways (besides just numerically) that the model cannot capture. In terms of how the model could be improved, methods such as regularization could help generalize across the multiple datasets instead of overfitting to any single dataset.

- C. Second Logistic regression function; while the previous logistic regression approach used specific features (area to perimeter ratio and longest diameter), the second logistic regression function used the raw pixel values from images
- D. Logistic regression with polynomial features; single and double feature
 - We chose to do another logistic regression, but with polynomial features as the polynomial features would allow the model to capture non-linear relationships between the features and the target variable. We chose the feature 'tbp_lv_color_std_mean' for the single feature and 'age_approx' in addition to 'tbp_lv_color_std_mean' for the double feature as we thought there could be a strong correlation for those features. For the model's performance, the low binary cross-entropy loss (0.1127 for single feature and 0.1019 for double

feature) indicates that the model's predicted probabilities are close to the actual class labels.

- E. Resnet18: The first figure shows the training loss vs testing loss of the resnet18 model. We chose 10 epochs as this is the "golden number" in computer vision models. We can see from the blue line that our training loss steadily decreases over time. This is to be expected, as our model is simply predicting a binary 0 or 1 based on if the image is malignant or not. The size of the training set was ~20k images, with the majority of them being benign. The test loss is much different, as we first decrease, then increase, then find our lowest loss around epoch 7 before the test loss starts to increase again.

V. Conclusion

- A. This is where you do a mind dump on your opinions and possible future directions. Basically what you wish you could have done differently. Here you close with final thoughts.
- B. Our goal from the beginning was to develop a model capable of early skin cancer detection, with image-derived features. Although we did not have a fully accurate model by the end, this experience has demonstrated to us the importance and capabilities of machine learning. While the models we implemented showed some promise, in retrospect we could have implemented more specialized models that were not covered in class, in addition to methods like combining more features and better balanced data. Based on the results we got, it seems like Resnet18 was the most successful model, although it was also the most sophisticated model. If we had more time, model ensembling would have been worth exploring, as we already have multiple different models that could have been combined. Better preprocessed data with more malignant samples would have also likely made this process smoother. While we did attempt to implement a hair removal algorithm, in the future we would have liked to get this algorithm fully working, which would likely contribute to higher quality data altogether. Overall, this project opened our eyes up to the power and versatility of machine learning and we hope to continue to try and harness this to help solve real world problems.

VI. Statement of Collaboration

- A. Srivatsa Gangavarapu, Group Member: Contributed to code and report
- B. Michael Hu, Group Member: Contributed to code and report