Tabular Data

Upsample and Downsample

The training data is imbalanced, so I employed a common approach of downsampling the majority class and oversampling the minority class. The target distribution for the classes is set at 70% for the majority class and 30% for the minority class. Consequently, using the proportional sample size formula, the fraction for downsampling the majority class is approximately 0.01, while the factor for upsampling the minority class is 5.0.

```
Class Distribution Before Sampling (%):
target
0    99.901927
1     0.098073
Name: proportion, dtype: float64

Calss Distribution After Sampling (%):
target
0    67.075893
1    32.924107
Name: proportion, dtype: float64
```

Set Weights for Training Purpose

After resampling the training data, I set the class weights by computing the class weight using the `compute_class_weight` function from sklearn. This ensures that the model gives appropriate importance to the minority class during training.

Impute Missing Values for Both Categorical and Numerical Data

To handle missing values, I imputed categorical variables with the most frequent value and numerical variables with the median. Using the median as an imputation method helps avoid the influence of outliers in the data.

One-Hot Encode for Categorical Data

I used one-hot encoding for categorical data such as 'anatom_site_general' because this variable is nominal and does not have an inherent order. One-hot encoding allows each category to be represented as a binary vector, preventing the introduction of any unintended ordinal relationships that could arise with label encoding.

Scale Numerical Variables

I scaled numerical variables to ensure they are on a similar scale, which is crucial for neural networks. Scaling helps improve convergence during training by allowing the optimization algorithm to navigate the loss surface more efficiently. Neural networks are sensitive to the scale of input features; features with larger ranges can disproportionately influence the gradients during backpropagation. By standardizing or normalizing the numerical variables, I ensure that all features contribute equally to the training process, leading to more stable and faster convergence.

Convert Age from Float to Integer

The data type of the age variable is float; therefore, I converted it to integer.

Image Data

Resize Image and Normalized Image Pixels

I resized the image to 128 x 128 pixels and normalized the pixel values using a general ratio of 225. This preprocessing step ensures that the image dimensions are consistent for input into the neural network, while scaling the pixel values helps normalize the data, improving model performance and convergence during training. The entire process was not fully completed because I still need to integrate additional feature engineering into the data loader.