Developing Melanoma Identifier Model for Web and App Deployment

Brought to you by Steven Yan

Did you know?

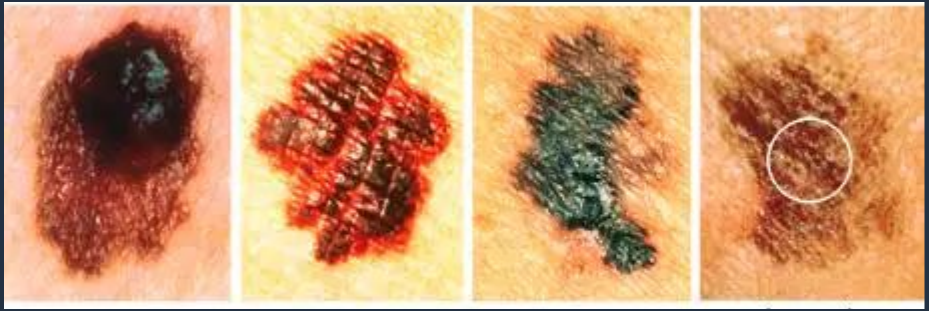
- Skin cancer is the most prevalent type of cancer
- Melanomas are responsible for 75% of skin cancer deaths despite being the least common
- When detected early, survival rate exceeds 95% and involves a minor surgical procedure
- In 2021, 106,000 new melanomas will be diagnosed, and just over 7000 people is expected to die of melanoma

Benefits:

- Recent incorporation of machine learning into medicine has shown to be more accurate than human expert observation
- Countless people need not die from melanoma through access to screening
- Equalize access to individuals without easy access to advice from expert clinician
    - Such population have a natural skepticism and reticence due to previous transgressions as well as cultural barriers

The ABCDE's of Skin Cancer:

- Asymmetry
- Border
- Color
- Diameter
- Evolution



Data Sources

- **2020 Training DICOMs and JPEGs**

  33126 dermoscopic training images split into training and validation

  - 467 melanoma vs. 26033 non-melanoma (training)
  - 117 melanoma vs. 6509 non-melanoma (validation)

- **2020 Metadata**

  Patient diagnostic information for 2000 patients

  - Patient ID, gender, approximate age, location of imaged site, diagnosis information, indicator of malignancy, binarized version of target variable (melanoma or not melanoma)

- **Minority Class Augmentation:**

  Acquired additional 4522 melanoma images from 2019 and 1114 melanoma images from 2018 datasets (5636 additional + 467 in training)

Data Preparation

EDA through metadata

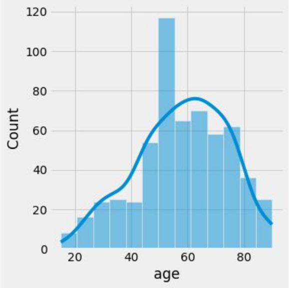
Age, Gender, Diagnosis, and Site (Melanoma vs. non-Melanoma, Testing vs. Training)

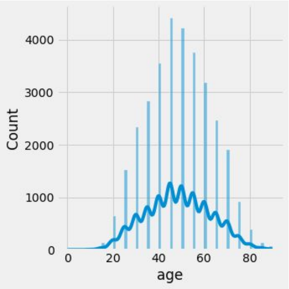
➔ **Distribution:** torso, lower extremity, upper extremity, head/neck, palm/soles, oral/genital in that order for both mel and non-mel

➔ **Gender:** 60/40 distribution, dataset matches industry established distribution

➔ **Diagnosis:** 27124 unknown, 5193 nevus, 584 melanoma, 133 seborrheic keratosis, and 5 subtypes (44, 37, 7, 1, and 1 respectively)

➔ **Age:**





Data Preparation

➔ File and folder management

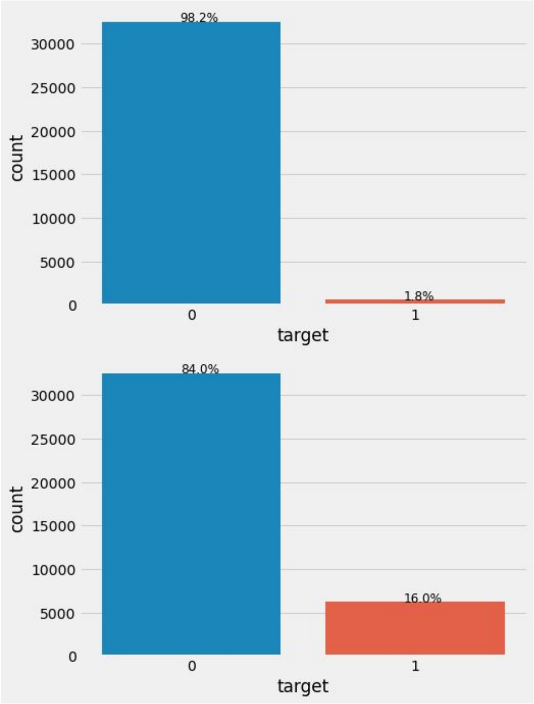
Challenges with unstructured data

◆ Keras requires the data to be organized into training, validation, and testing folders with the classes organized as subfolders to create the testing sets
◆ Time consuming process of moving folders
◆ Challenge of incorporating folder

➔ Class Imbalance

Employ a variety of methods to address severe class imbalance

◆ ImageDataGenerator
◆ Albumentations



Modeling

Baseline Model:

- Sequential()
- 2 convolutional layers with input shape (224, 244, 3) with filters applied to extract different features:
  - Filters: number of filters that convolutional layer will learn
  - Kernel_size: specifies width and height of 2D convolutional window
  - Padding: same ensure that spatial dimensions are the same after convolution
  - Activation: activation function that will be applied for convolutional layers

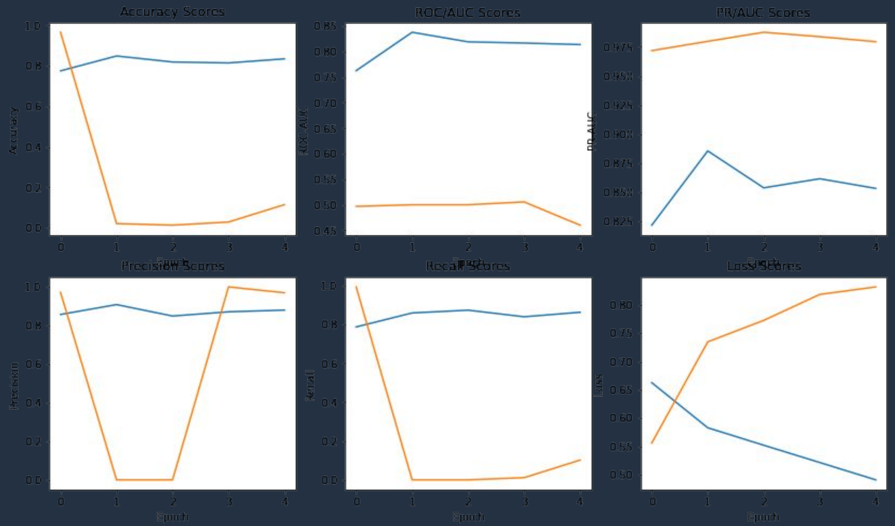
  -
    ```
    layers.Conv2D(input_shape=(224,224,3), filters=64, kernel_size=(3,3),
    padding="same", activation="relu"))
    ```

Modeling

- BatchNormalization()
  - acts like standardization or normalization for regression models
- MaxPool2D()
  - To reduce dimensionality of images by reducing number of pixels in output
  - `layers.MaxPool2D(pool_size=(2,2),strides=(2,2))`
- Flatten()
  - To be able to generate a prediction, flatten output of convolutional base
  - `layers.Flatten()`
- Dense layers feeds output of convolutional base to neurons
  - `layers.Dense(units=4096, activation="relu"))`
- Loss function: `loss= ‘binary_crossentropy’`
- Optimizer: Adam(learning_rate=0.01)

Metrics

- Accuracy
- Precision (Positive Predictive Value)
- Recall (True Positive Rate)
- ROC-AUC Score
- PR-AUC Score
- Training Model Scores:
  - Acc: 0.8435, Prec: 0.8906, Rec: 0.8645, AUC: 0.8283, PR-AUC: 0.8624
- Validation Model Scores:
  - Acc: 0.1141, Prec: 0.9697, Rec: 0.1017, AUC: 0.4600, PR-AUC: 0.9794



Next Steps:

Moving to Amazon AWS for greater computing power

Exploring different techniques to address class imbalance

Building a rudimentary iOS and Android app for deployment

Contact Information:

Steven Yan

Email: stevenyan@uchicago.edu

LinkedIn: https://www.linkedin.com/in/examsherpa

Github: https://www.github.com/examsherpa

Feel free to reach out with any questions or to collaborate on any projects!