**COMPUTER VISION**

**PNEUMONIA DETECTION CHALLENGE**



**Capstone Project Group 1**

**Team**

|  |  |
| --- | --- |
| **Jyant Mahara** | **Capstone Project Mentor** |
| Sneh Sweta | Capstone Group 1 Team |
| Harsha Bhat | Capstone Group 1 Team |
| Renjini Jayakumar | Capstone Group 1 Team |
| Shivee | Capstone Group 1 Team |
| Revathy Jayakumar | Capstone Group 1 Team |

Table of Contents

[Executive Summary: 5](#_Toc183898224)

[PROBLEM STATEMENT 7](#_Toc183898225)

[EDA and Pre-processing 8](#_Toc183898226)

[Data Collection 8](#_Toc183898227)

[Load and read the dataset: 9](#_Toc183898228)

[Data Cleaning: 10](#_Toc183898229)

[Check for Missing Values 10](#_Toc183898230)

[Check for Duplicate Value 10](#_Toc183898231)

[Feature Analysis and Class Distribution 10](#_Toc183898232)

[a. Class Distribution 10](#_Toc183898233)

[Merge Data Sets 12](#_Toc183898234)

[Map Training and Testing Images to Their Classes: 12](#_Toc183898235)

[Dealing with DICOM Images: 13](#_Toc183898236)

[Visualize Data 14](#_Toc183898237)

[Distribution of Age vs Lung Opacity: 14](#_Toc183898238)

[Centers of Lung Opacity AP vs PA: 16](#_Toc183898239)

[Sample images: 16](#_Toc183898240)

[Model and Model Building: 18](#_Toc183898241)

[a. Model Architecture: 18](#_Toc183898242)

[b. Model Summary: 19](#_Toc183898243)

[c. Validation Loss and Validation Accuracy: 20](#_Toc183898244)

[d. Confusion Matrix: 21](#_Toc183898245)

[How to improve your model performance? 24](#_Toc183898246)

[Data Preparation 24](#_Toc183898247)

[Model Architecture 24](#_Toc183898248)

[Optimization Techniques 24](#_Toc183898249)

[Hyperparameter Tuning 25](#_Toc183898250)

[Training Techniques 25](#_Toc183898251)

[Use Pretrained Models 25](#_Toc183898252)

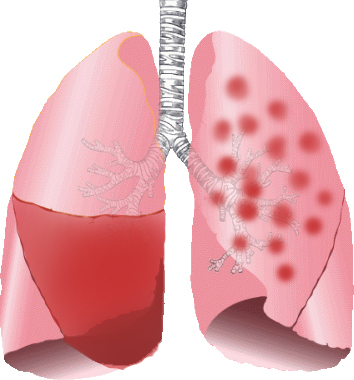
[Monitor Performance 25](#_Toc183898253)

[Debugging and Fine-Tuning 25](#_Toc183898254)

[Ensemble Methods 25](#_Toc183898255)

# Executive Summary:

Pneumonia is a life-threatening lung infection that affects millions of people worldwide. Traditional pneumonia diagnosis methods can be time-consuming and expensive. Artificial Intelligence (AI) has the potential to revolutionize pneumonia detection by enabling accurate and timely diagnoses.



AI algorithms can analyze chest X-rays and CT scans to identify patterns and anomalies associated with pneumonia. These algorithms can detect pneumonia with high accuracy, potentially reducing the need for invasive and expensive diagnostic procedures.

AI-powered pneumonia detection systems can also help healthcare professionals prioritize treatment for patients with severe cases of pneumonia, ensuring that they receive timely and appropriate care.

However, AI-powered pneumonia detection systems are not without their challenges. One major concern is the potential for bias in AI algorithms, which could lead to misdiagnosis or underdiagnosis of certain patient populations. Addressing these concerns will be crucial for ensuring the safe and effective use of AI in pneumonia diagnosis.

Overall, AI-powered pneumonia detection systems have the potential to improve the accuracy and speed of diagnosis. AI systems can analyze medical images and patient data to identify patterns and make predictions, allowing for more accurate diagnosis and treatment. This technology can also help healthcare professionals to make more informed decisions about patient care, reducing the risk of misdiagnosis and improving patient outcomes. However, there are still challenges to be addressed, such as the need for high-quality data and the need for AI algorithms to be validated in clinical settings. With continued development and refinement, AI- based pneumonia detection has the potential to revolutionize the field of medical imaging and improve the care of patients with respiratory infections.

# PROBLEM STATEMENT

**DOMAIN**: Health Care

**CONTEXT**: Computer vision can be used in health care for identifying diseases. In Pneumonia detection we need to detect Inflammation of the lungs. In this challenge, you’re required to build an algorithm to detect a visual signal for pneumonia in medical images. Specifically, your algorithm needs to automatically locate lung opacities on chest radiographs.

**DATA DESCRIPTION**: The goal is to build a computer vision algorithm to detect pneumonia by identifying lung opacities in chest radiograph images. The dataset includes medical images in DICOM format, containing both pixel data and metadata. The classification task involves three classes:

* Pneumonia - Lung opacities indicating pneumonia.
* Not Normal No Lung Opacity - Abnormalities without pneumonia that might resemble pneumonia in appearance. This extra third class indicates that while pneumonia was determined not to be present, there was nonetheless some type of abnormality on the image and oftentimes this finding may mimic the appearance of true pneumonia.
* Normal - Healthy cases with no visible abnormalities.

The challenge is to automatically locate and identify lung opacities accurately, distinguishing true pneumonia cases from other abnormalities to assist in diagnosis.

Dicom original images: - Medical images are stored in a special format called DICOM files (\*.dcm).

You can refer to the details of the dataset in the above link –

Acknowledgements: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/overview/acknowledgements.

# EDA and Pre-processing

Our approach is to have the following steps:

## Data Collection

The dataset contains the following files and folders:

**1. stage\_2\_train\_labels.csv**

Attributes:

* patientId: A unique identifier for each patient. This is used to group related data points and link them to the same individual.
* x, y, height, width: These coordinates represent the bounding box information for the lung opacity regions. The bounding box outlines the suspected areas of pneumonia in the medical images.
  + x: The x-coordinate of the upper-left corner of the bounding box.
  + y: The y-coordinate of the upper-left corner.
  + height: The vertical size of the bounding box.
  + width: The horizontal size of the bounding box.
* target: The dependent variable, indicating the presence or absence of lung opacity.
  + - 0: No lung opacity (normal or other abnormalities not mimicking pneumonia).
    - 1: Lung opacity present, which may indicate pneumonia.

**2. stage\_2\_detailed\_class\_info.csv**

Attributes:

* patientId: A unique identifier for each patient. This is used to group related data points and link them to the same individual.
* class: Class label which mentions about the patient's Lung condition - No Lung Opacity/Not Normal , Lung Opacity, Normal (3 classes are there)

Apart from the above-mentioned data files (in csv format), the dataset also contains the images folders

**stage\_2\_train\_images**

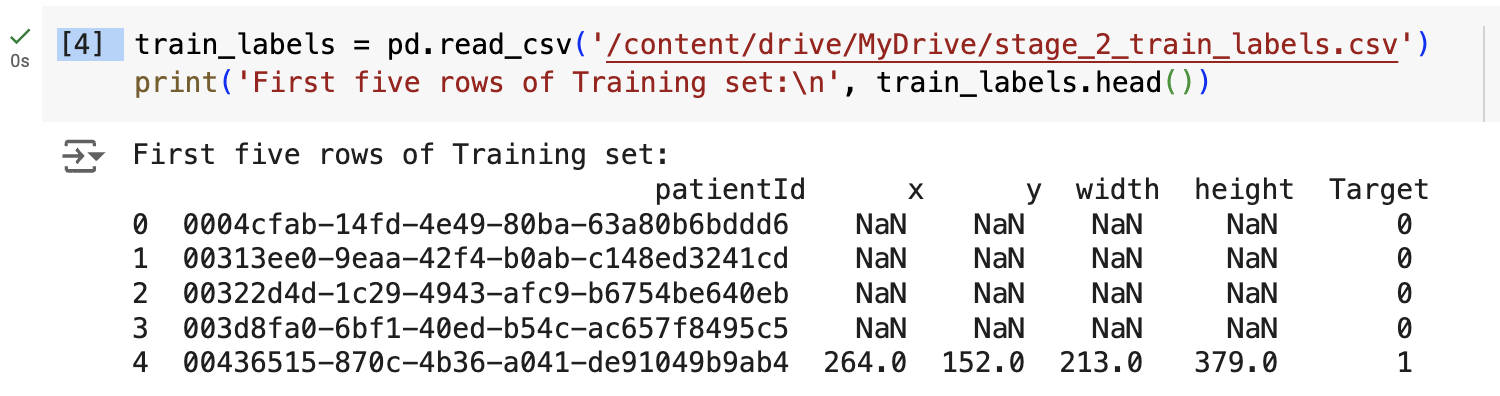
**stage\_2\_test\_images**

The images in the above-mentioned folders are stored in a special format called DICOM files (\*.dcm). They contain a combination of header metadata as well as underlying raw image arrays for pixel data.

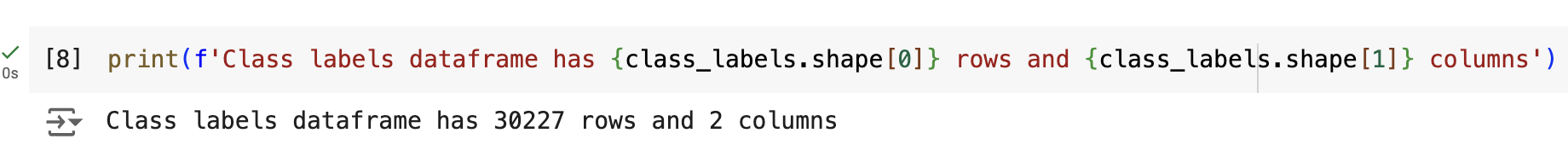
## 

## Load and read the dataset:

Check Shape of the dataset



Shape of dataset:



## Data Cleaning:

### 

### Check for Missing Values



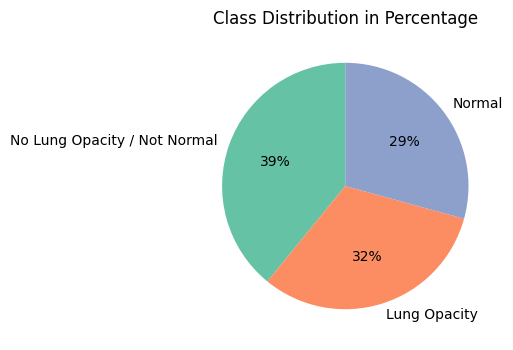
### Check for Duplicate Value



## Feature Analysis and Class Distribution

### 

### a. Class Distribution





To handle and analyze this data, we can break it down as follows:

1. **Data Overview:**

We have a dataset with:

Patient IDs (some of which are duplicated due to multiple bounding boxes per patient).

Bounding box coordinates (defined by x, y, width, height).

A binary Target column, where:

Target = 1 indicates a finding of pneumonia.

Target = 0 indicates no definitive evidence of pneumonia.

1. **Class Labels:**

The dataset also has a class label with three possible values:

Lung Opacity: Indicates evidence of pneumonia (corresponds to Target = 1).

Normal or No Lung Opacity / Not Normal: Corresponds to Target = 0.

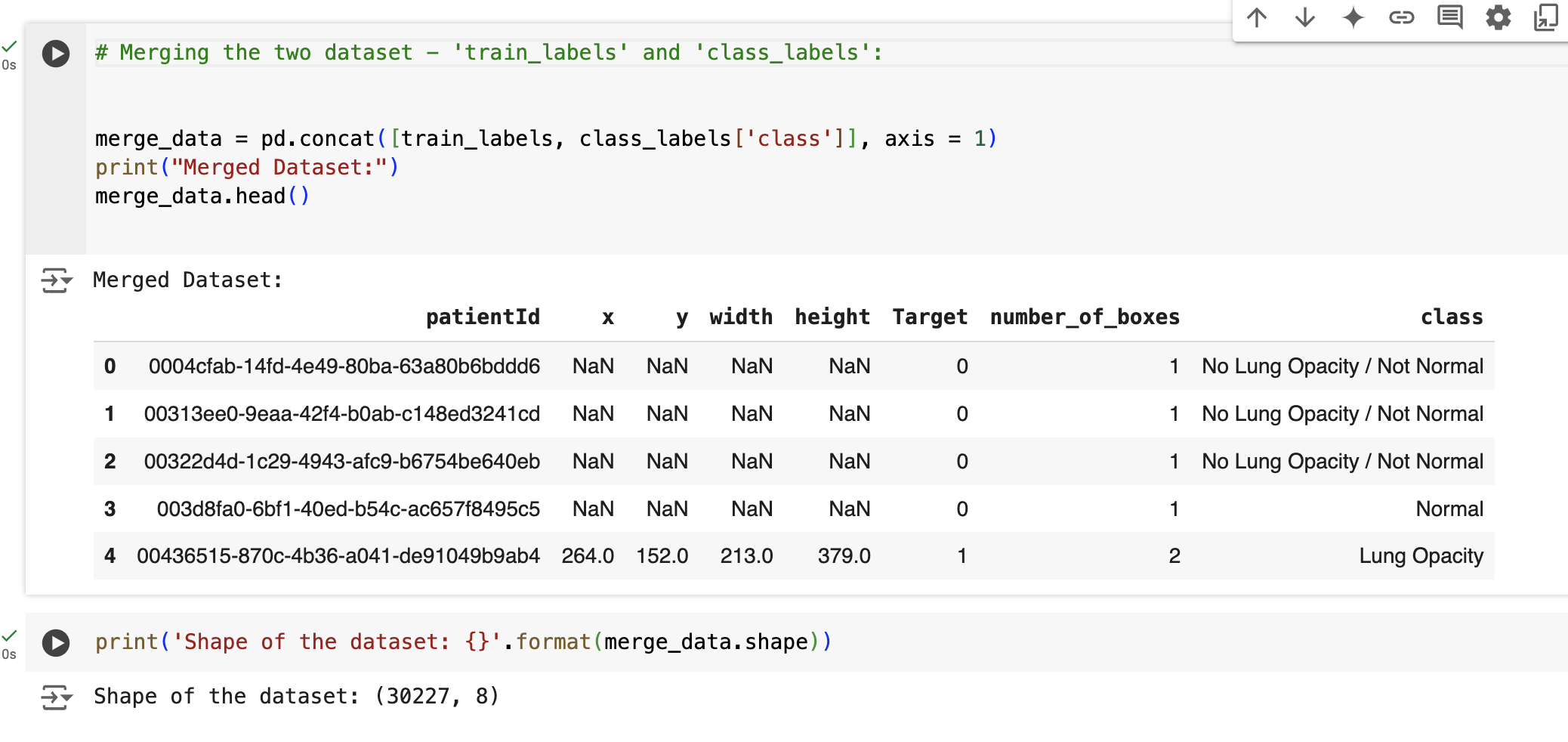
1. **Distribution of Bounding Boxes:**

A significant number of patient IDs (23,286 or 87%) have only one bounding box.

A few cases (13 patients) have as many as four bounding boxes, which may represent multiple findings within the same image.

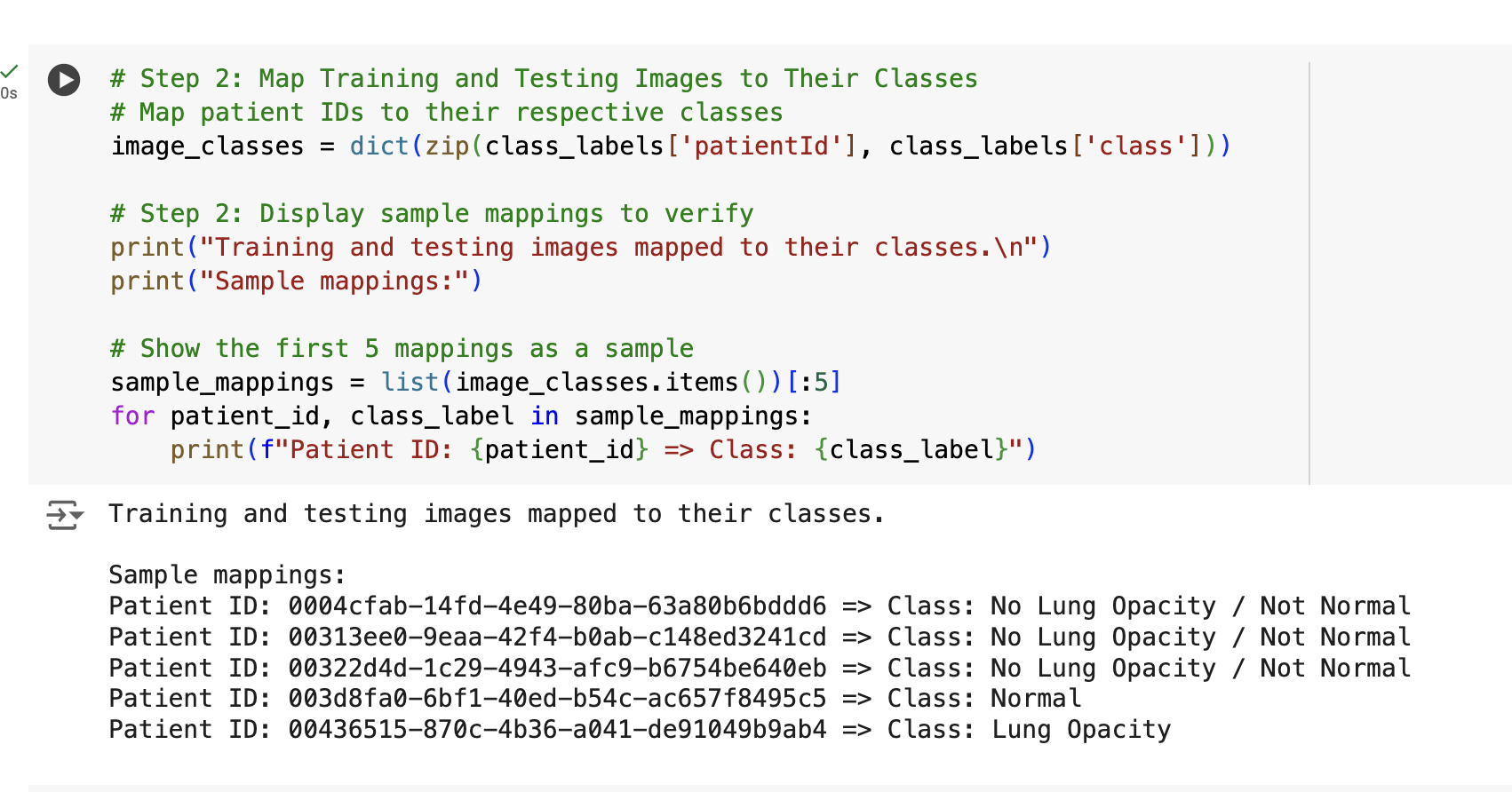
## Merge Data Sets

**Merge the two data frames with labels and class dataset**.



### 

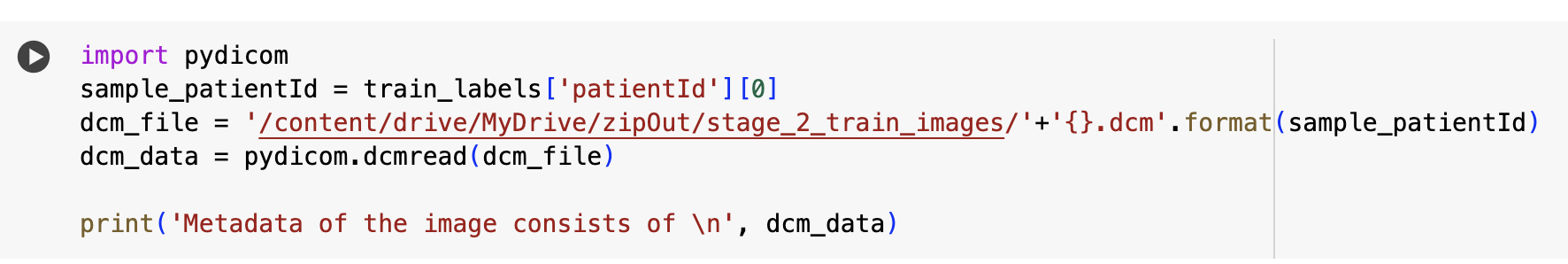
### Map Training and Testing Images to Their Classes:

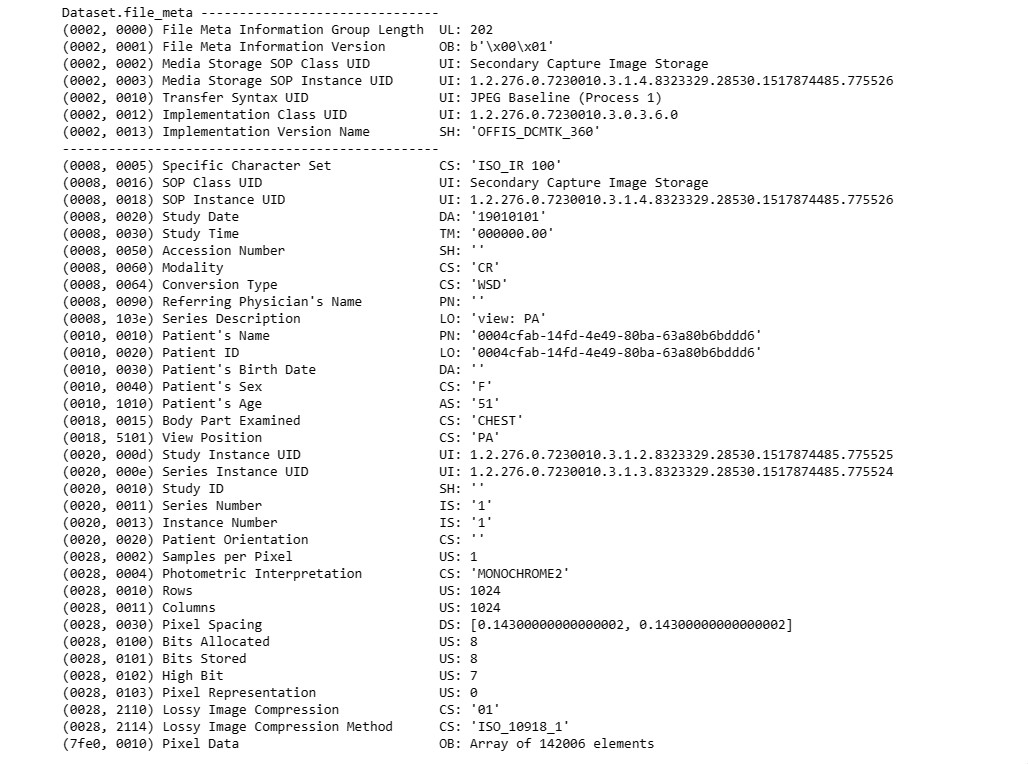


## Dealing with DICOM Images:

DICOM images are typically used for storing medical images and are rich in information. Alongside the image data, they include important patient details such as the patient's name, age, sex, and the physician's name, among others.

To preview DICOM images without extracting any information, you can utilize the following code. First, make sure to install the pydicom Python package by running pip install pydicom.





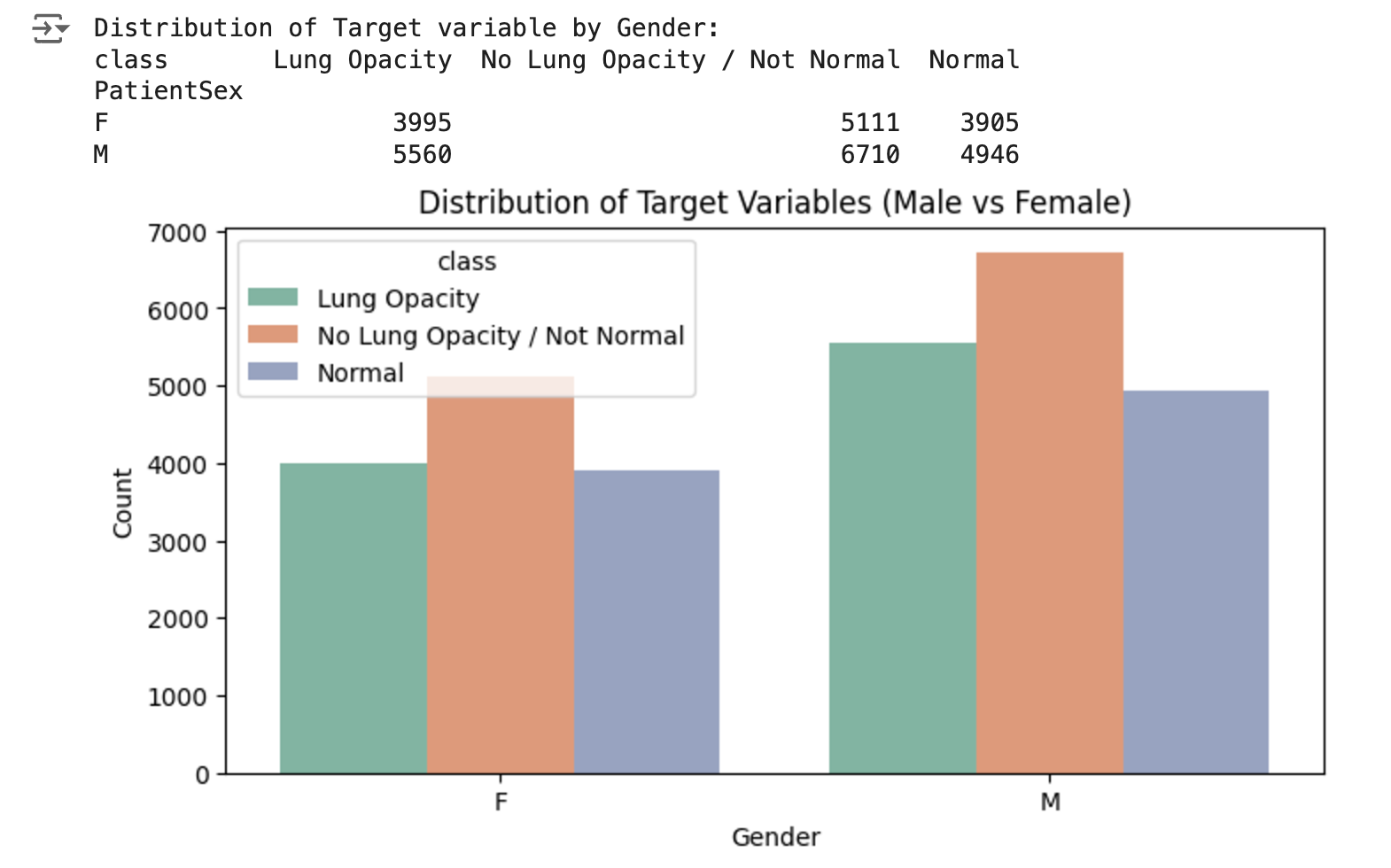
Here is the META Data from sample DICOM image file

## 

## Visualize Data

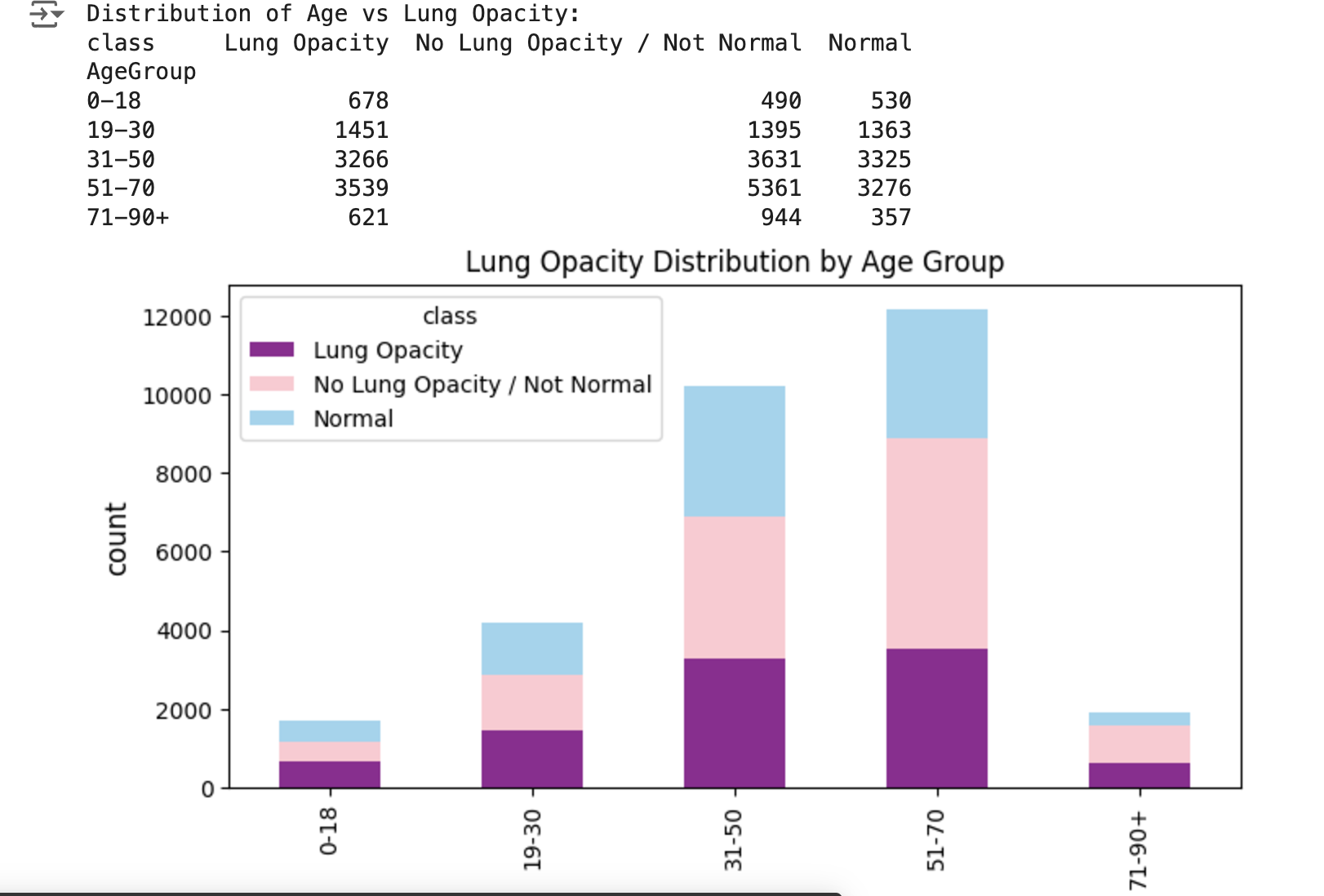
**Distribution of Target Variables Male vs Female**:

Within Male and Female group, the rate of Pneumonia risk is more in Male compared to Female

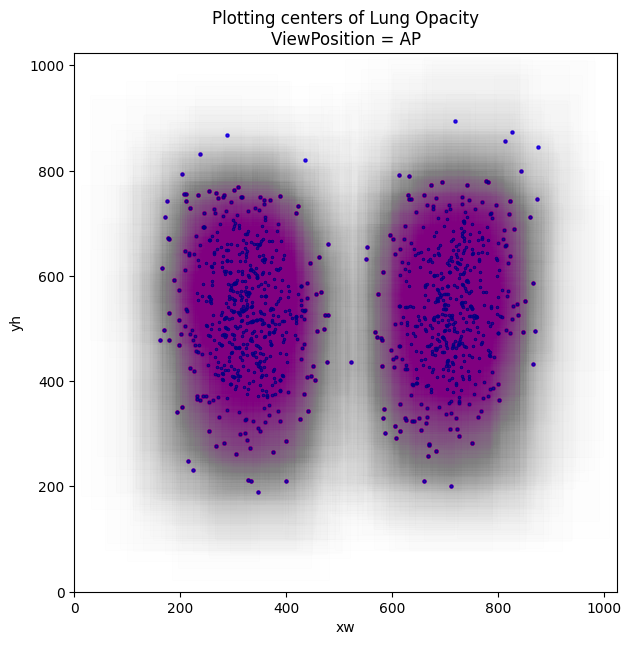


## Distribution of Age vs Lung Opacity:

* Lung opacity is more common in older age groups, particularly in the 51-70 age range, suggesting an age-related increase in prevalence or risk.
* Younger age groups (0-18 and 19-30) are predominantly "Normal," indicating lower susceptibility or fewer cases of lung opacity.
* As age increases, the proportion of "Lung Opacity" cases grows, while the proportion of "Normal" cases decreases.
* The "No Lung Opacity / Not Normal" class remains relatively stable across all age groups.
* Healthcare Implications: Screening and preventative measures should target middle-aged and older adults (31-70+) due to higher prevalence of lung opacity in these groups.



### Centers of Lung Opacity AP vs PA:



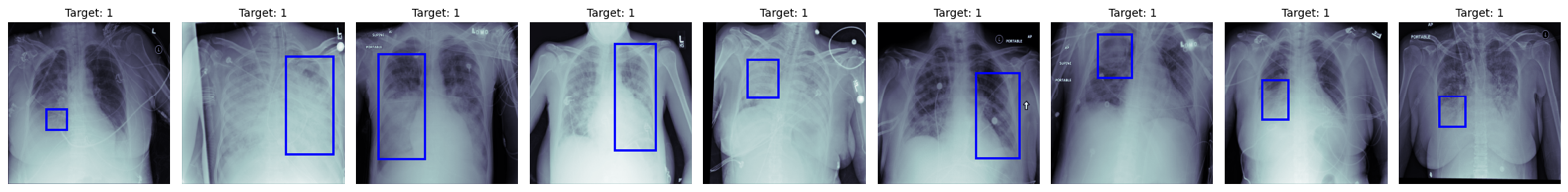
* BodyPart examined is unique for all cases and is CHEST in the training dataset and that was also expected.
* Unique in Modality is CR i.e. Computer Radiography
* Overall ViewPosition is almost equally distributed in the training dataset but for cases where. Target=1, most of the view position are AP.

## Sample images:

Here are some sample images from various categories, including Normal, Lung Opacity, and No Lung Opacity/Normal. For the objectives of this project, we will focus solely on predictions related to Normal and Lung Opacity.

Additionally, here are some random images from the training dataset. For the images depicting lung opacity, we have included bounding boxes to highlight the affected areas.



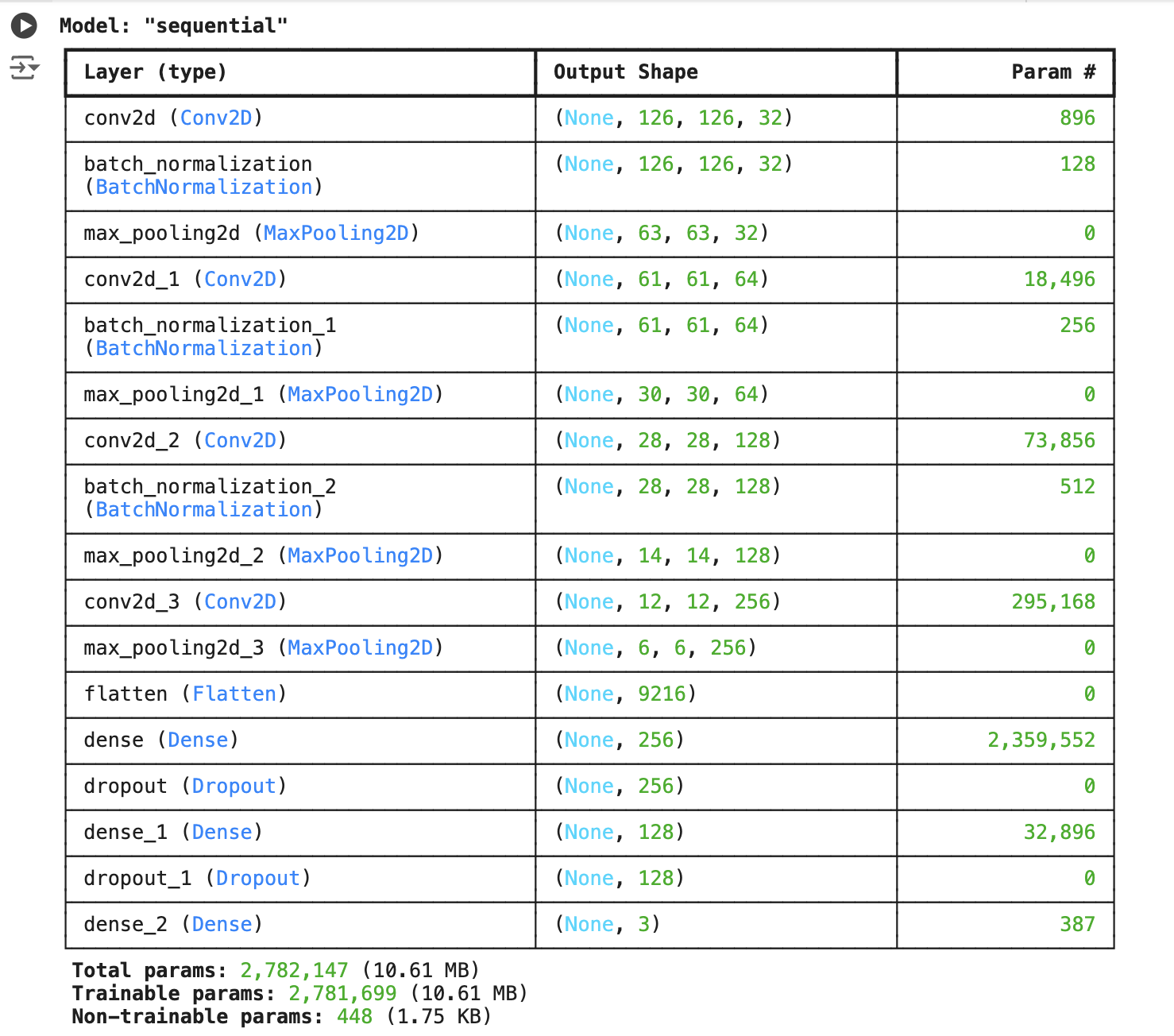


# Model and Model Building:

## Model Architecture:

* **Input Dimensions:**
  + Input Shape: (128, 128, 3) (image size is 128×128128 \times 128128×128 with RGB channels).
  + Output Classes: 3 (classification targets).
* **Architecture Overview:**
  + **Convolutional Layers:**
    - 3 convolutional layers with ReLU activation:
      * First layer: 323232 filters of size 3×33 \times 33×3.
      * Second layer: 646464 filters of size 3×33 \times 33×3.
      * Third layer: 128128128 filters of size 3×33 \times 33×3.
    - Each convolutional layer is followed by:
      * **Batch Normalization**: Stabilizes and accelerates training.
      * **MaxPooling**: Down samples feature maps by 2×22 \times 22×2.
  + **Flatten Layer:**
    - Converts the 3D output from the convolutional layers into a 1D vector.
  + **Dense (Fully Connected) Layers:**
    - Two fully connected layers with 128128128 neurons and ReLU activation.
    - Dropout (50%): Applied to both dense layers to mitigate overfitting.
  + **Output Layer:**
    - Dense layer with 333 neurons (for the 3 classes) and Softmax activation to output class probabilities.
* **Compilation Details:**
  + Optimizer: **Adam** (adaptive learning rate optimization).
  + Loss Function: **Categorical Crossentropy** (suitable for multi-class classification).
  + Metric: **Accuracy**.

## Model Summary:

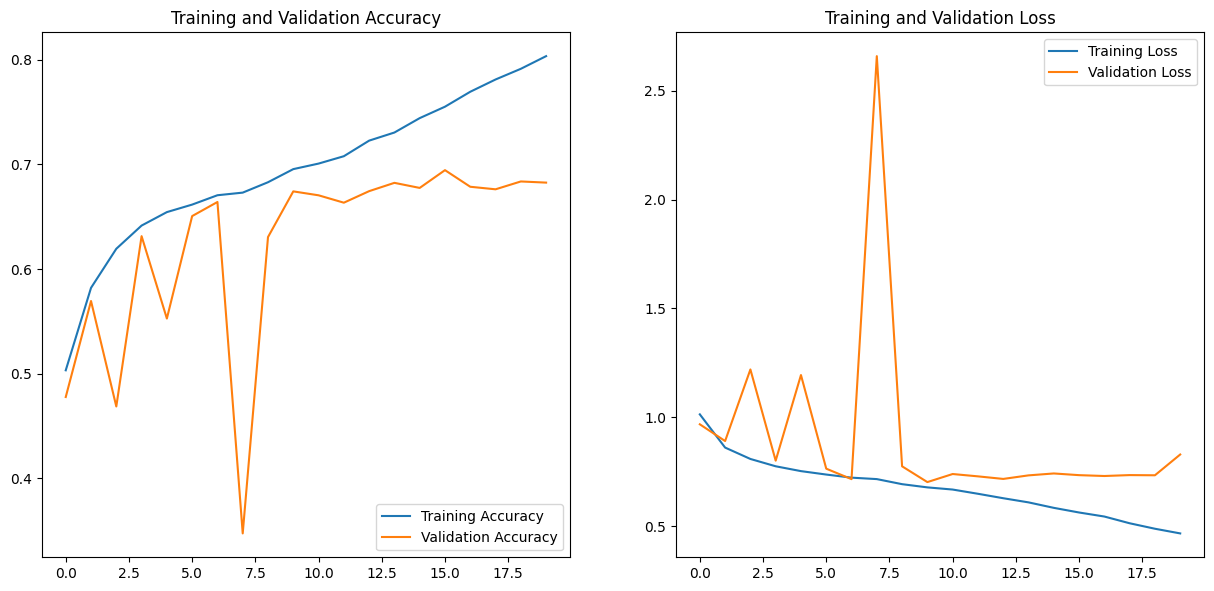


**Strengths of this Model:**

* Batch Normalization: Enhances model stability and performance by normalizing layer inputs.
* Dropout Layers: Reduces overfitting, especially with the higher dropout rate (50%50\%50%).
* Flexible Design: Modularized to accept any input dimensions and number of classes.

## Validation Loss and Validation Accuracy:

The following graph represent the validation loss and validation accuracy for the improved model.



**Training and Validation Accuracy:**

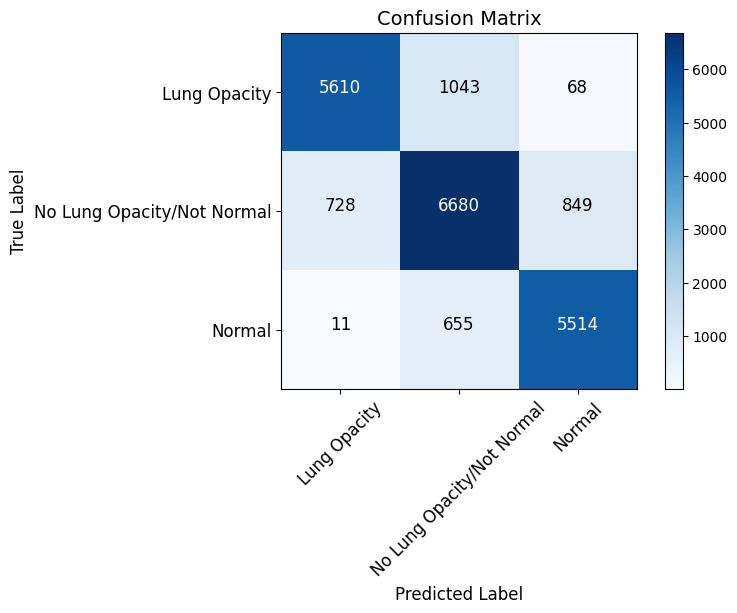
* Observation:
  + Training accuracy is steadily increasing over the epochs, indicating that the model is learning from the training data effectively.
  + Validation accuracy shows fluctuations but stabilizes at a slightly lower level compared to training accuracy. This may point to mild overfitting or the need for additional regularization techniques.

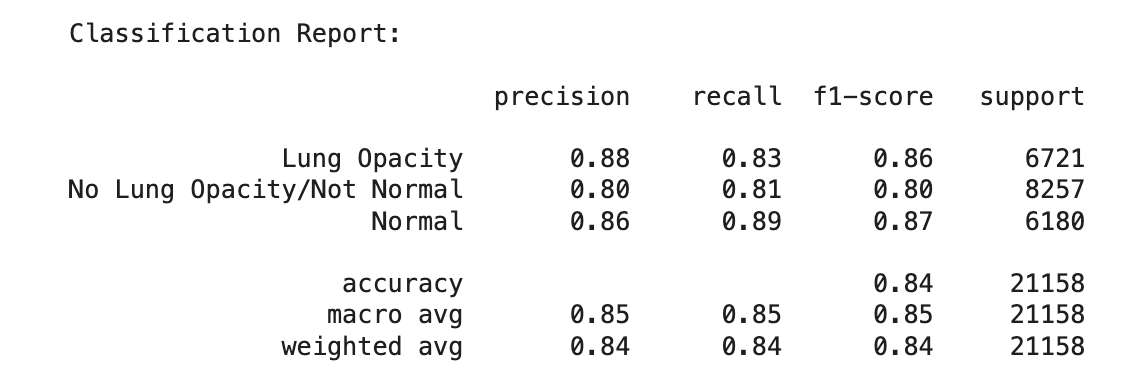
**Training and Validation Loss:**

* Observation:
  + Training loss is gradually decreasing, reflecting consistent improvement in the model's ability to minimize error on the training set.
  + Validation loss decreases initially but shows some large spikes and fluctuations, which could indicate:
    - Unstable learning: The model might struggle to generalize.
    - Overfitting: Validation loss diverging from training loss at later epochs.

## Confusion Matrix:

**Train set:**

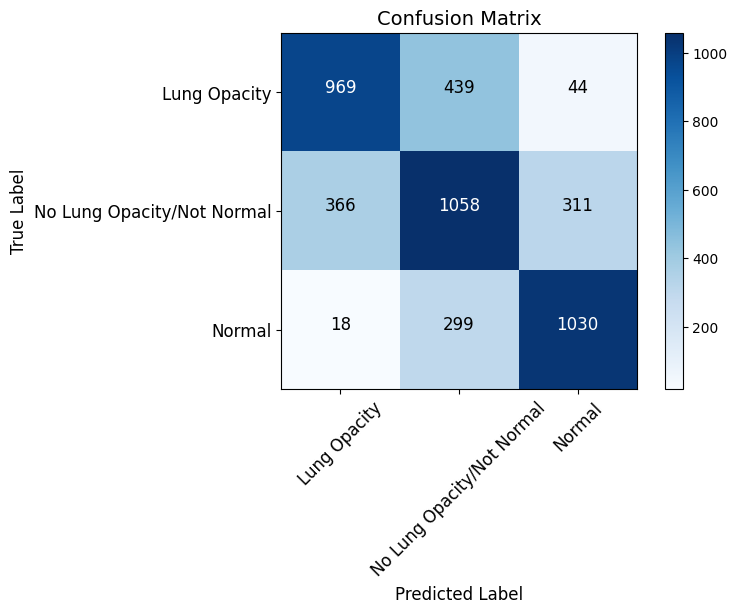


****

**Observations:**

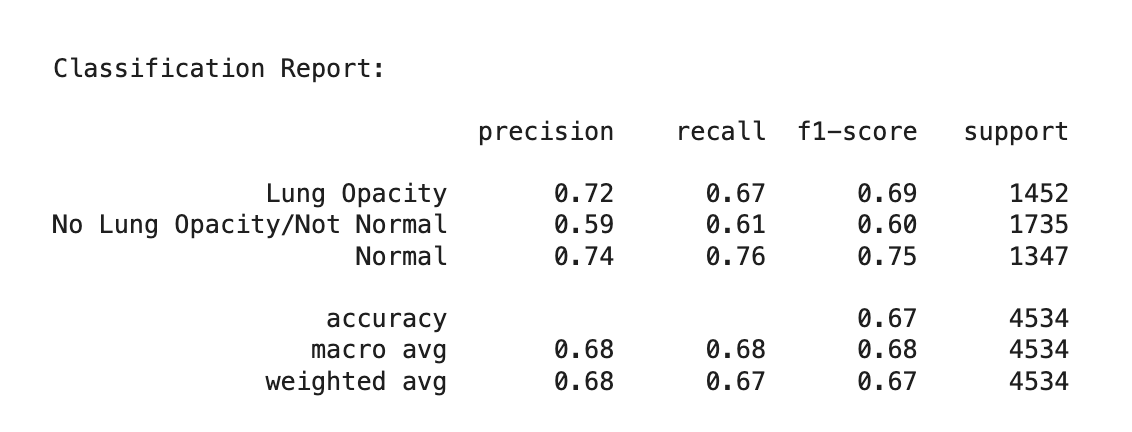
* The No Lung Opacity/Not Normal class shows slightly lower precision and recall compared to the others. This could be due to:
  + Overlap or similarity between the No Lung Opacity/Not Normal and other classes, leading to confusion.
  + Potential class imbalance or inherent complexity in this label.
* Normal and Lung Opacity classes are performing well, with higher precision and recall, indicating good differentiation by the model.

**Test Set:**



**Observations and Insights:**

1. Highest Confusion:
   * The model struggles most between Lung Opacity and No Lung Opacity/Not Normal, with a significant number of samples being misclassified in both directions (439 and 366, respectively).
   * There is moderate confusion between No Lung Opacity/Not Normal and Normal (311 samples misclassified).
2. Normal Class:
   * The Normal class has the highest accuracy, with relatively fewer misclassifications.
3. Class Imbalance or Ambiguity:
   * The high confusion between Lung Opacity and No Lung Opacity/Not Normal might indicate overlapping features between these classes or an imbalance in the training data.

****

**Observations**

1. Performance :
   * The "Normal" class performs the best, with the highest F1-score (0.75).
   * The "No Lung Opacity/Not Normal" class performs the worst, with the lowest F1-score (0.60). This aligns with confusion observed in the confusion matrix.
2. Precision vs Recall:
   * For "Lung Opacity" and "Normal," precision and recall are balanced.
   * For "No Lung Opacity/Not Normal," precision is slightly lower than recall, indicating false positives are more common.
3. Class Support:
   * The "No Lung Opacity/Not Normal" class has the highest support (1735 samples), but its performance metrics are the weakest, suggesting potential challenges in feature extraction or data quality for this class.

**Strategies to Improve Model Performance**

1. Data Preparation

* Diversify the Dataset: Use data augmentation techniques like flipping, rotating, scaling, or cropping to increase variability without collecting new samples.
* Normalize and Scale: Ensure data consistency by scaling image pixel values to a range like [0, 1] or applying normalization with mean-zero and unit variance.

1. Enhance Model Architecture

* Optimize Network Design: Experiment with increasing or reducing the number of layers and neurons to find the optimal balance.
* Incorporate Regularization:
  + Add dropout layers to mitigate overfitting.
  + Use weight regularization (e.g., L1/L2 penalties) to control model complexity.

1. Refine Optimization Techniques

* Tune Learning Rates:
  + Use learning rate schedulers, such as ReduceLROnPlateau or cosine annealing, for dynamic adjustment.
  + Implement warm-up strategies by starting with a low learning rate and gradually increasing it.
* Choose Effective Optimizers: Test optimizers like Adam, RMSprop, SGD with momentum, or AdamW to identify the best fit for your task.

1. Hyperparameter Tuning

* Conduct systematic searches using methods like grid search or random search to optimize key parameters.

1. Improve Training Techniques

* Adjust Batch Sizes: Test different batch sizes to find a balance between performance and generalization.
* Shuffle Data: Randomize the training order to eliminate biases from sequential data.
* Apply Early Stopping: Stop training once validation loss stabilizes to avoid overfitting.

1. Leverage Pretrained Models

* Transfer Learning: Fine-tune models trained on large datasets for your specific problem.
* Feature Extraction: Freeze initial layers of a pretrained model and only train the task-specific layers.

1. Monitor and Evaluate Performance

* Use cross-validation to split data into training and validation sets, ensuring your model generalizes well.

1. Debug and Fine-Tune

* Control Gradients: Apply gradient clipping to manage exploding gradients.
* Optimize Weight Initialization: Experiment with methods like Xavier or He initialization.
* Validate Code: Check dataset pipelines, loss functions, and optimizer updates for errors.

1. Use Ensemble Methods

* Combine predictions from multiple models to improve accuracy and robustness.