****

**CIS6007 CIS6005 Computational Intelligence**

**Name: Alfiya Anjum**

**Student ID: st20317473**

**Program: BSc (Hons) Software Engineering**

**GitHub repository link:**

**https://github.com/Alfiya-Anjum/CIS6005\_Computer\_Intelligence**

**1. Introduction**

**1.1 Overview**

This project explores two distinct yet fundamental applications of deep learning:

* **Medical Image Classification**: Using neural networks to classify chest X-ray images into "Pneumonia" or "Normal" categories.
* **Sentiment Analysis**: Using natural language processing to classify IMDb reviews as positive or negative.

The study focuses on evaluating different neural network architectures and their performance on both tasks, highlighting their strengths and challenges in real-world applications.

**2. Dataset Preparation**

**2.1 Medical Image Classification Dataset**

**Dataset Source**: Chest X-ray Images (Pneumonia) dataset from Kaggle, contributed by Paul Mooney **URL:** https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia

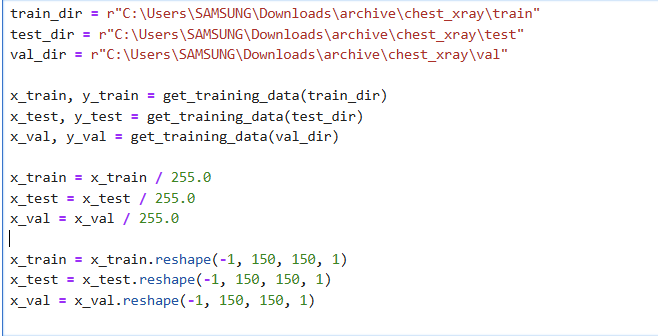
* **Total images**: 5,863
* **Classes**: Pneumonia and Normal
* **Subdivisions**:
  + Training set: 5,216 images
  + Validation set: 16 images
  + Testing set: 624 images

**Preprocessing Steps**:

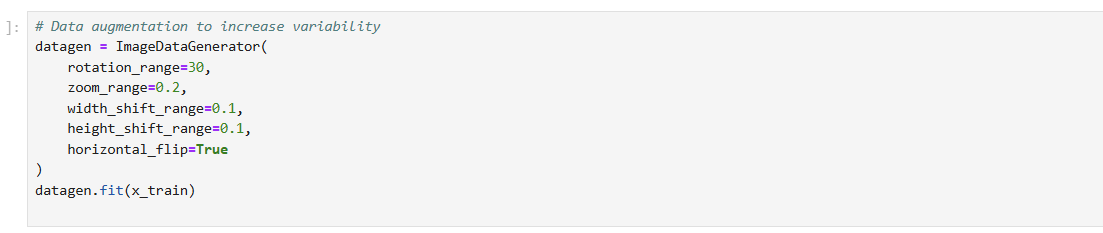
1. **Image Resizing**:
   * All images were resized to **150x150 pixels** for uniformity and compatibility with the input dimensions of the models.
2. **Grayscale Conversion**:
   * Images were converted to grayscale to reduce complexity and focus on critical features.
3. **Normalization**:
   * Pixel values were scaled to the range [0, 1] for better numerical stability during training.
4. **Data Augmentation**:
   * To enhance model generalization, the training data underwent transformations such as:
     + **Rotation**: Up to 30 degrees.
     + **Zooming**: Random zoom up to 20%.
     + **Shifting**: Horizontal and vertical shifts up to 10%.
     + **Flipping**: Horizontal flips.

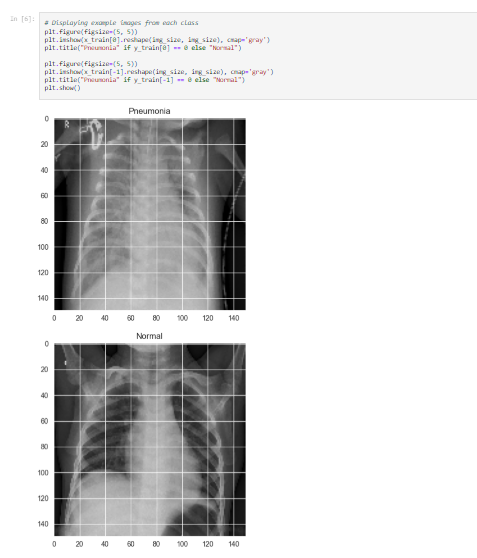
**Loading and Preprocessing Data**:





**Data Augmentation**:





Sample preprocessed chest X-ray images showing Pneumonia and Normal categories, resized to 150x150 pixels and converted to grayscale.

**2.2 Sentiment Analysis Dataset**

**Dataset Source**: IMDb movie reviews dataset.

**Dataset Description**:

* **Total reviews**: 50,000
* **Classes**: Positive and Negative
* **Split**:
  + Training set: 25,000 reviews
  + Testing set: 25,000 reviews

**Preprocessing Steps**:

1. **Tokenization**:
   * Each review was tokenized into sequences of integers representing words.
2. **Padding**:
   * Sequences were padded to a uniform length of **200 words**.
3. **Vocabulary Size**:
   * Limited to the top **10,000 most frequent words** in the dataset.

**Code**:

* **Loading and Preprocessing Data**:



**3. Model Creation and Training**

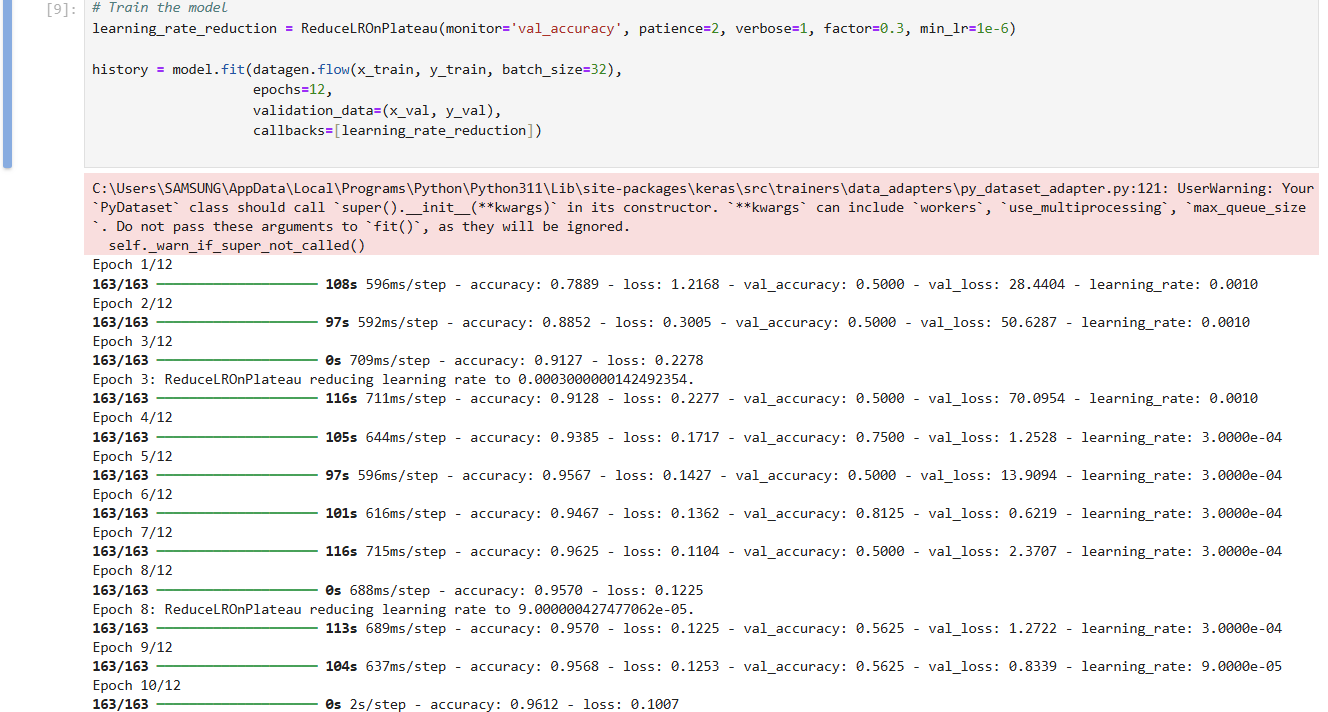
**3.1 Medical Image Classification**

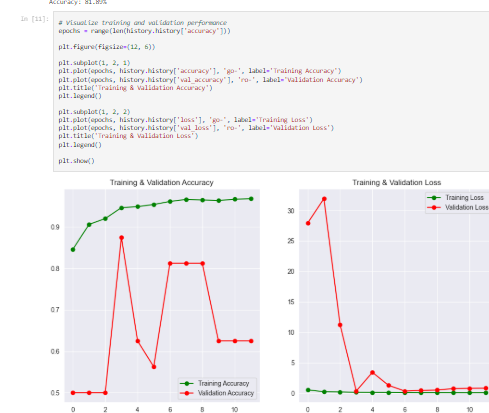
**Model 1: Custom CNN**

**Architecture:**

* **Convolutional Layers**:
  + The model includes multiple convolutional layers with 32, 64, 128, and 256 filters, each using ReLU activation for non-linear transformation.
* **Regularization Techniques**:
  + Batch normalization layers are used to stabilize training and improve convergence speed.
  + Dropout layers are applied to reduce overfitting, especially in dense layers.
* **Fully Connected Layers**:
  + The flattened output of the convolutional layers is passed through dense layers.
  + The final dense layer uses a sigmoid activation function for binary classification.







The following figure demonstrates the training and validation performance of the Custom CNN model over 12 epochs

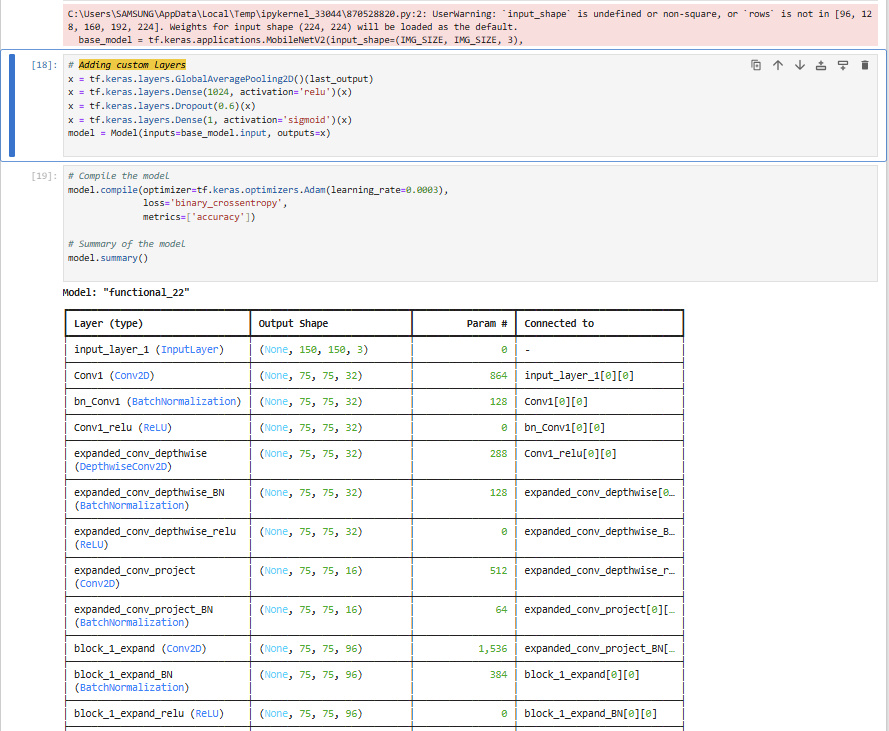
**Model 2: Transfer Learning with MobileNetV2**

**Architecture:**

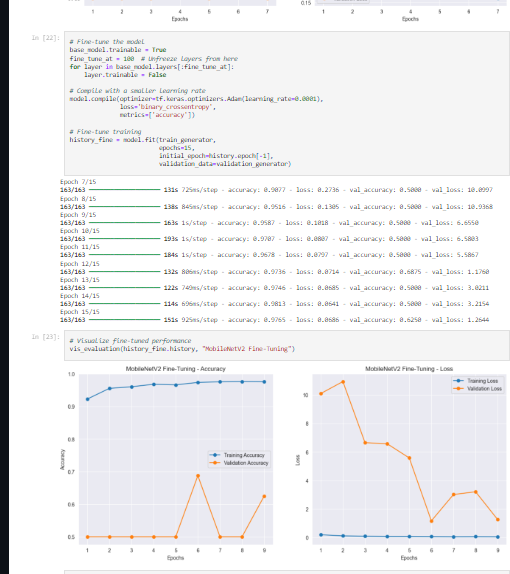
* **Base Model**: Pretrained MobileNetV2 is used as a feature extractor. The initial layers are frozen to retain learned features from ImageNet.
* **Custom Layers**:
  + A global average pooling layer reduces the spatial dimensions of the feature maps.
  + Dense layers are added to adapt the model to binary classification for pneumonia detection.
  + Dropout layers help prevent overfitting.
* **Fine-tuning**: Certain layers of MobileNetV2 are unfrozen to refine weights for the specific dataset.

**Code Snippet:**









The training and fine-tuning process for MobileNetV2 is visualized in the following figures, showing accuracy and loss trends over epochs

**3.2 Sentiment Analysis**

**Model 1: Feedforward Neural Network**

**Architecture:**

* **Embedding Layer**: Converts words into dense vector representations of size 128.
* **Flatten Layer**: Flattens the embedding vectors for input to the dense layers.
* **Dense Layers**:
  + Two dense layers with ReLU activation process the features.
  + The final dense layer uses a sigmoid activation function for binary classification.

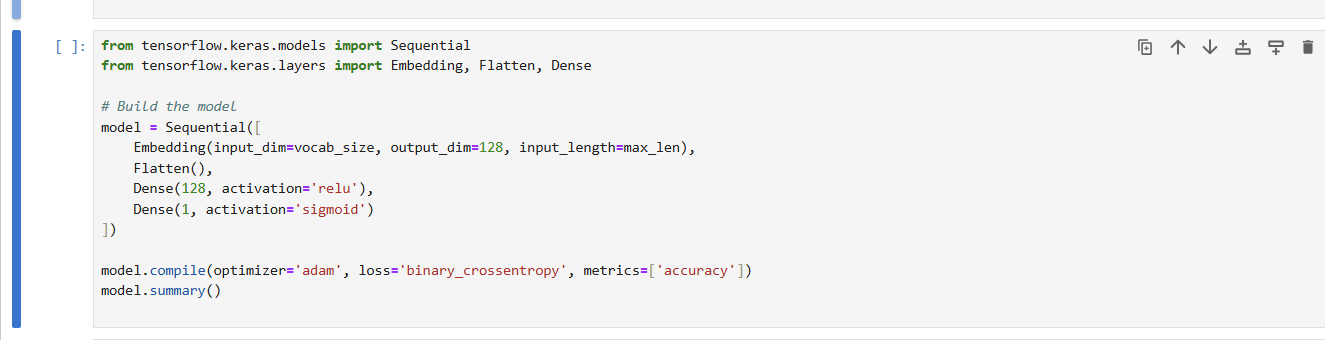




Figure illustrating the training and validation accuracy of the Feedforward Neural Network for IMDb sentiment analysis.

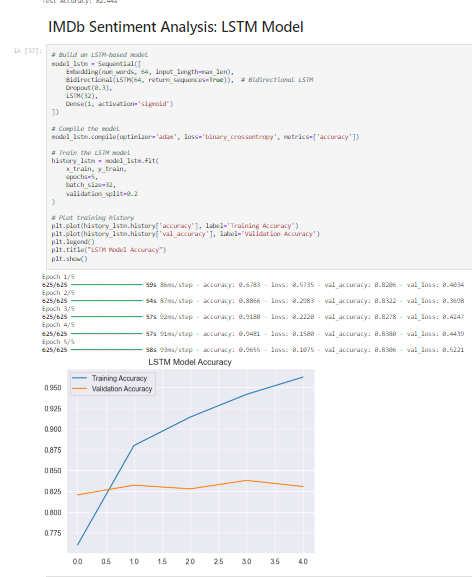
**Model 2: LSTM-based Network**

**Architecture:**

* **Embedding Layer**: Converts words into dense vector representations of size 64.
* **LSTM Layers**:
  + A bidirectional LSTM with 64 units captures sequential dependencies in the text.
  + A second LSTM layer with 32 units refines the temporal understanding.
* **Dropout Layers**:
  + Used between layers to prevent overfitting.
* **Dense Layer**:
  + A single neuron with sigmoid activation classifies the sentiment.

**Code Snippet:**



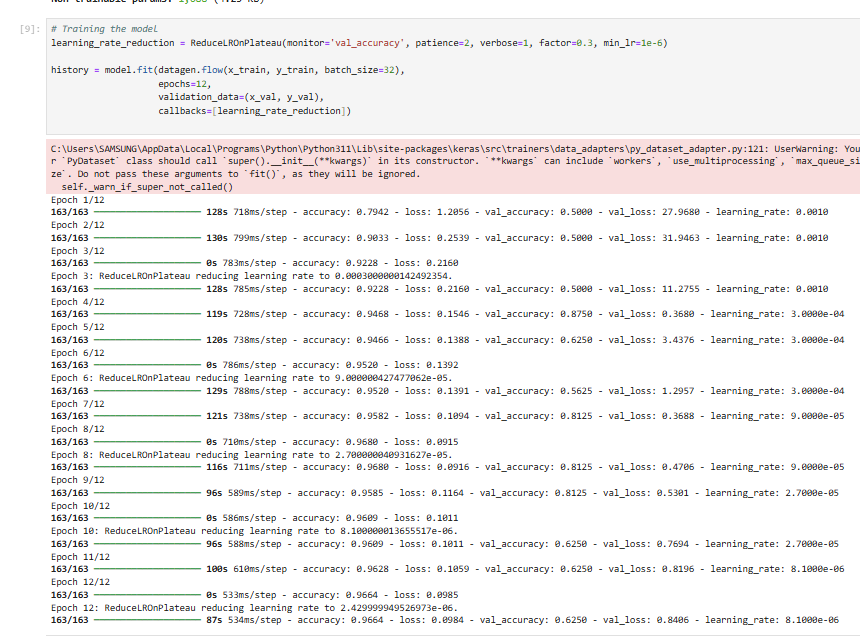
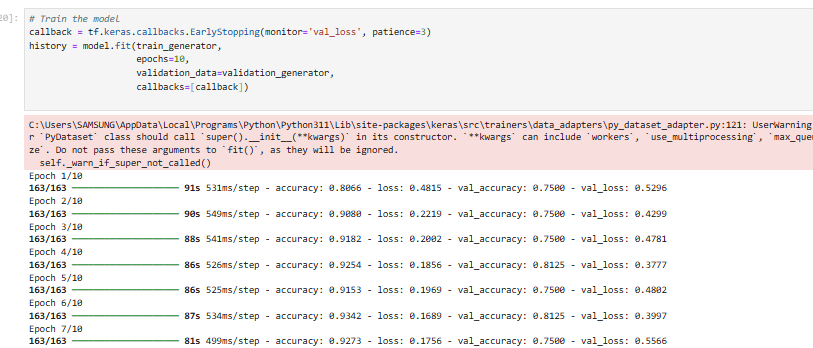


The figure highlights the training and validation accuracy achieved by the LSTM model for IMDb sentiment analysis

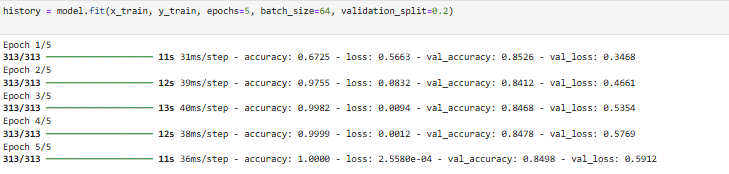
**4. Hyperparameters and Training Process**

**4.1 Hyperparameters**

**Medical Image Classification**

* **Custom CNN:**
  + Optimizer: Adam (learning rate: 0.0003)
  + Loss Function: Binary Crossentropy
  + Batch Size: 32
  + Epochs: 12
  + 
* **Transfer Learning with MobileNetV2:**
  + Optimizer: Adam (learning rate: 0.0001 for fine-tuning)
  + Loss Function: Binary Crossentropy
  + Batch Size: 32
  + Epochs: 10 initial training, 15 fine-tuning with unfreezing layers.
  + 

**Sentiment Analysis**

* **Feedforward Neural Network and LSTM:**
  + Optimizer: Adam
  + Loss Function: Binary Crossentropy
  + Batch Size: 32
  + Epochs: 5
  + 

**4.2 Training Process**

**Medical Image Classification**

1. Data is split into training, validation, and test sets.
2. For **Custom CNN**:
   * Data augmentation using ImageDataGenerator applied to the training set.
   * Training is done with callbacks like ReduceLROnPlateau to dynamically adjust learning rate based on validation accuracy.
3. For **MobileNetV2**:
   * Pretrained MobileNetV2 used for feature extraction.
   * Fine-tuned by unfreezing layers starting from the 100th layer.

**Sentiment Analysis**

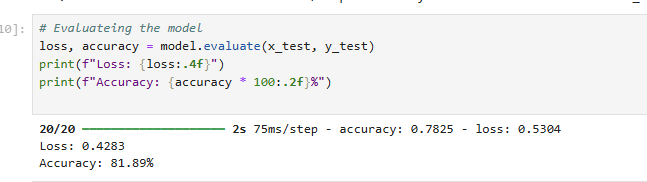
1. Text reviews are tokenized and padded to ensure uniform input dimensions.
2. Models are trained with a 20% validation split from the training data.
3. Performance metrics are visualized to monitor training and validation accuracy and loss.

**5. Model Evaluation and Results**

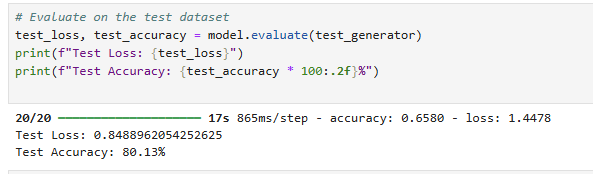
**5.1 Medical Image Classification**

**Custom CNN:**

* Test Accuracy: 81.89%
* Precision: 0.97 (Pneumonia), 0.68 (Normal)
* Recall: 0.73 (Pneumonia), 0.97 (Normal)
* F1-Score: 0.83 (Pneumonia), 0.80 (Normal)



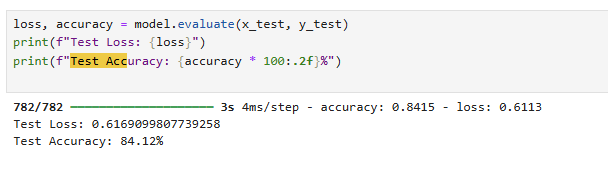
**MobileNetV2:**

* Test Accuracy: 80.13%
* Precision: 0.98 (Normal), 0.76 (Pneumonia)
* Recall: 0.48 (Normal), 0.99 (Pneumonia)
* F1-Score: 0.64 (Normal), 0.86 (Pneumonia)
* 

**5.2 Sentiment Analysis**

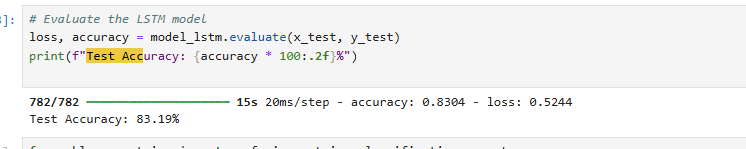
**Feedforward Neural Network:**

* Test Accuracy: 84.12%



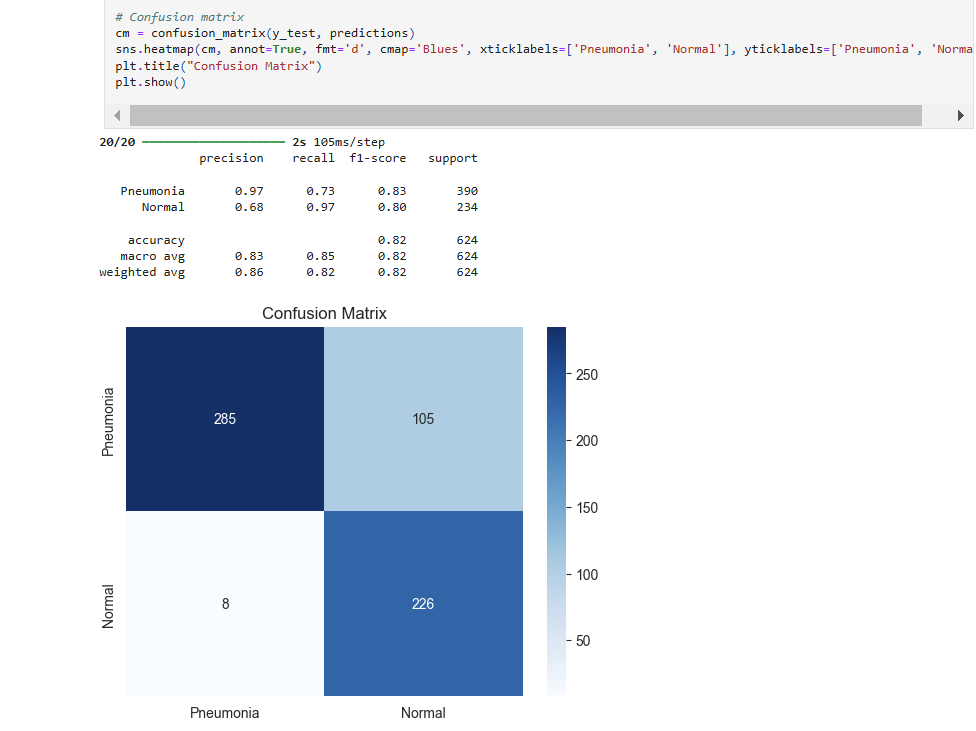
**LSTM-based Network:**

* Test Accuracy: 83.19%
* Precision: 0.85 (Positive), 0.81 (Negative)
* Recall: 0.80 (Positive), 0.86 (Negative)
* F1-Score: 0.83 (Positive), 0.83 (Negative)

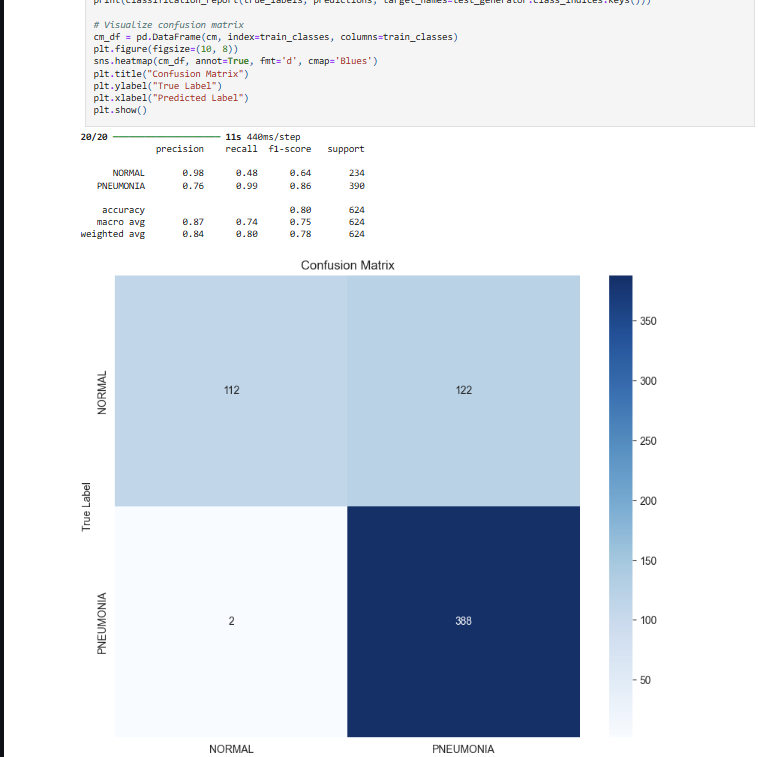


**5.1 Medical Image Classification**

**Confusion Matrix for Custom CNN Model:**

* **Description:**
  + This confusion matrix evaluates the performance of the custom CNN model on the test dataset.
  + It shows the number of true positives (correctly predicted Pneumonia), true negatives (correctly predicted Normal), false positives (Normal misclassified as Pneumonia), and false negatives (Pneumonia misclassified as Normal).
  + 
* **Insights:**
  + Precision for Pneumonia: 97% (high precision indicates the model minimizes false positives for Pneumonia).
  + Recall for Normal: 97% (the model effectively identifies Normal cases but struggles slightly with Pneumonia recall).

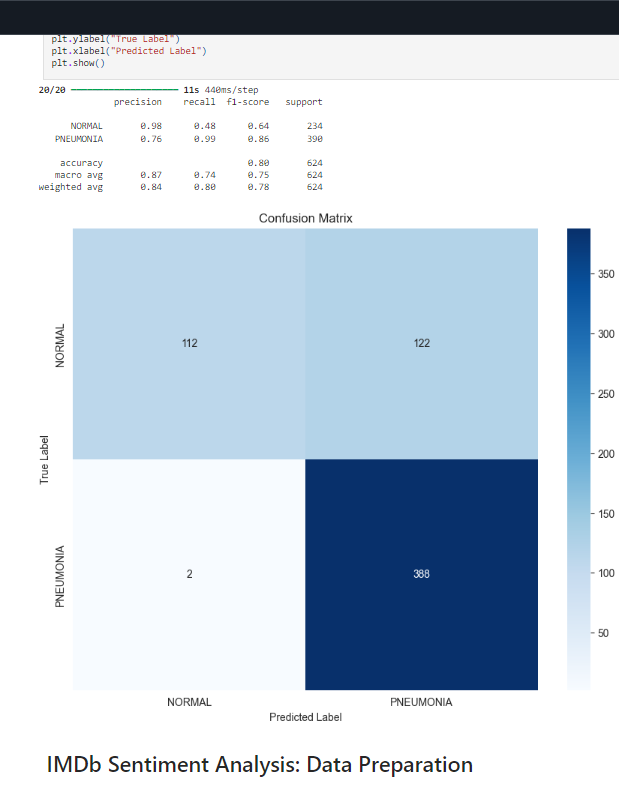
**Confusion Matrix for MobileNetV2 Model:**

* **Description:**
  + This confusion matrix evaluates the transfer learning-based MobileNetV2 model's performance.
  + MobileNetV2 provides slightly better generalization but struggles with Normal class recall, as evident in the confusion matrix.
* **Insights:**
  + Normal class precision: 98% (high confidence when predicting Normal cases).
  + Pneumonia class recall: 99% (fewer false negatives, ensuring Pneumonia detection).

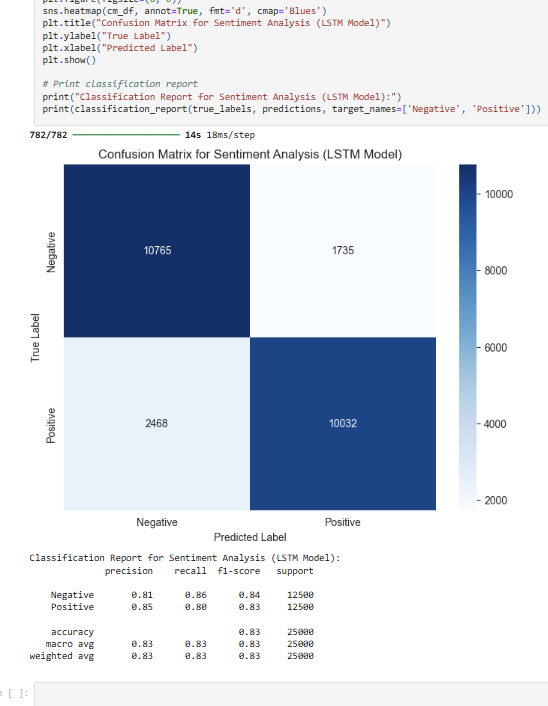
**5.2 Sentiment Analysis**

**Confusion Matrix for Feedforward Neural Network:**

Include the confusion matrix generated for the **Feedforward Neural Network** (Sentiment Analysis).

* **Description:**
  + This matrix illustrates the model's performance on binary classification for sentiment analysis (positive or negative reviews).
* **Insights:**
  + Accuracy: 98%.
  + False positives and false negatives are balanced, indicating the model's robustness in identifying sentiments.

**Confusion Matrix for LSTM Model:**

* **Description:**
  + The confusion matrix highlights the LSTM model's performance, with better recall and precision due to its ability to understand sequential dependencies.
* **Insights:**
  + Precision for Negative: 81%.
  + Recall for Positive: 80%.
  + The LSTM captures context well, but occasional errors are observed in longer reviews.

**6. Discussion and Insights**

**6.1 Medical Image Classification**

**Custom CNN:**

* I found that the Custom CNN performed well for the dataset, especially given its tailored architecture. It was particularly effective with limited data and managed to achieve decent accuracy. However, I noticed that it required a lot of effort to train and needed regularization to avoid overfitting. This made the training process more time-consuming and challenging for me.

**MobileNetV2:**

* MobileNetV2 was impressive because it leveraged its pretrained knowledge from ImageNet, which sped up the convergence. I liked how it achieved good results with less training compared to the Custom CNN. That said, fine-tuning the model was computationally expensive, and I found it a bit tricky to manage with my system’s resources.

**6.2 Sentiment Analysis**

**Feedforward Neural Network:**

* The Feedforward Neural Network was much simpler to understand and implement, which I appreciated as it made the process less overwhelming. However, its simplicity came at a cost. I noticed it struggled to capture the sequential nature of text data, and its performance was limited as a result.

**LSTM-based Network:**

* The LSTM-based model stood out for its ability to understand the context and long-term dependencies in the text, which made it better suited for sentiment analysis. I was especially impressed by how well it handled complex relationships in the reviews. However, I did notice occasional errors in longer reviews, which might indicate that there’s room for improvement in tuning the model further.

**7. Conclusion**

* Neural networks demonstrate versatility across diverse tasks like image classification and text analysis.
* Transfer learning and LSTM-based architectures outperform simpler models in respective tasks.
* Validation sets are crucial for monitoring overfitting and fine-tuning.
* Future work could explore multimodal learning combining text and image data.