New Section

New Section

```
from google.colab import files
files.upload()
```

Choose Files kaggle.json.json

kaggle.json.json(application/json) - 75 bytes, last modified: 11/25/2025 - 100% done

```
Saving kaggle.json.json to kaggle.json.json
{'kaggle.json.json':
```

```
# Install Kaggle API
!pip install -q kaggle
```

```
!mkdir -p ~/.kaggle
!cp kaggle.json.json ~/.kaggle/kaggle.json
!chmod 600 ~/.kaggle/kaggle.json
```

```
!kaggle datasets list
```

```
ref                                                              title
---------------------------------------------------------------  ----------------------------------
wardabilal/spotify-global-music-dataset-20092025                 Spotify Global Music Dataset (2009
sadiajavedd/students-academic-performance-dataset                Students_Academic_Performance_Data
prince7489/employee-salary-dataset                               Employee Salary Dataset
ayeshaimran123/social-media-and-mental-health-balance            Social Media and Mental Health Bal
zahranusrat/salary                                               Salary Insights by Job Role (2024)
hanif13/supermarket-sales                                        Supermarket Sales
shahzadi786/world-smartphone-market-2025                         World Smartphone Market 2025
khushikyad001/ai-impact-on-jobs-2030                             AI Impact on Jobs 2030
zubairamuti/bmw-car-sales-record-2010-2024                       BMW Car Sales Record (2010-2024)
kainatjamil12/niteee                                             Netflix Movies and TV Shows Compre
wardabilal/student-stress-analysis                               Student Stress Analysis
zubairdhuddi/shopping-dataset                                    Shopping Behavior Dataset
rockyt07/formula-1-championships-1950-2025                       Formula 1 Championships (1950-2025
ayeshasiddiqa123/cars-pre                                        Car Price Analysis Dataset
parthpatel2130/realistic-loan-approval-dataset-us-and-canada     Realistic Loan Approval Dataset |
alizabrand/school-performance-analysis                           School Performance Analysis
shaistashahid/youtube-analytics-data                             YouTube Analytics Data
saadaliyaseen/shopping-behaviour-dataset                         Shopping Behaviour Dataset
umuttuygurr/e-commerce-customer-behavior-and-sales-analysis-tr   E-Commerce Customer Behavior & Sal
ayeshaseherr/diamonds                                            diamonds
```

```
!kaggle datasets download -d paultimothymooney/chest-xray-pneumonia
```

```
Dataset URL: https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia
License(s): other
Downloading chest-xray-pneumonia.zip to /content
 99% 2.26G/2.29G [00:18<00:00, 227MB/s]
100% 2.29G/2.29G [00:18<00:00, 135MB/s]
```

```
!unzip chest-xray-pneumonia.zip
```

```
  inflating: chest_xray/train/PNEUMONIA/person981_bacteria_2907.j
  inflating: chest_xray/train/PNEUMONIA/person981_bacteria_2907.jpeg
  inflating: chest_xray/train/PNEUMONIA/person981_bacteria_2908.jpeg
  inflating: chest_xray/train/PNEUMONIA/person981_virus_1657.jpeg
  inflating: chest_xray/train/PNEUMONIA/person982_bacteria_2909.jpeg
  inflating: chest_xray/train/PNEUMONIA/person982_virus_1658.jpeg
  inflating: chest_xray/train/PNEUMONIA/person983_bacteria_2910.jpeg
  inflating: chest_xray/train/PNEUMONIA/person983_virus_1660.jpeg
  inflating: chest_xray/train/PNEUMONIA/person984_bacteria_2911.jpeg
  inflating: chest_xray/train/PNEUMONIA/person985_bacteria_2912.jpeg
  inflating: chest_xray/train/PNEUMONIA/person986_bacteria_2913.jpeg
  inflating: chest_xray/train/PNEUMONIA/person987_bacteria_2914.jpeg
  inflating: chest_xray/train/PNEUMONIA/person988_bacteria_2915.jpeg
  inflating: chest_xray/train/PNEUMONIA/person988_virus_1666.jpeg
  inflating: chest_xray/train/PNEUMONIA/person989_virus_1667.jpeg
  inflating: chest_xray/train/PNEUMONIA/person98_virus_182.jpeg
  inflating: chest_xray/train/PNEUMONIA/person990_bacteria_2917.jpeg
  inflating: chest_xray/train/PNEUMONIA/person991_bacteria_2918.jpeg
  inflating: chest_xray/train/PNEUMONIA/person991_virus_1669.jpeg
  inflating: chest_xray/train/PNEUMONIA/person992_bacteria_2919.jpeg
  inflating: chest_xray/train/PNEUMONIA/person992_bacteria_2920.jpeg
  inflating: chest_xray/train/PNEUMONIA/person992_virus_1670.jpeg
  inflating: chest_xray/train/PNEUMONIA/person993_bacteria_2921.jpeg
  inflating: chest_xray/train/PNEUMONIA/person993_virus_1671.jpeg
  inflating: chest_xray/train/PNEUMONIA/person994_bacteria_2922.jpeg
  inflating: chest_xray/train/PNEUMONIA/person994_virus_1672.jpeg
  inflating: chest_xray/train/PNEUMONIA/person995_bacteria_2923.jpeg
  inflating: chest_xray/train/PNEUMONIA/person995_virus_1676.jpeg
  inflating: chest_xray/train/PNEUMONIA/person996_bacteria_2924.jpeg
  inflating: chest_xray/train/PNEUMONIA/person996_virus_1677.jpeg
  inflating: chest_xray/train/PNEUMONIA/person997_bacteria_2926.jpeg
  inflating: chest_xray/train/PNEUMONIA/person997_virus_1678.jpeg
  inflating: chest_xray/train/PNEUMONIA/person998_bacteria_2927.jpeg
  inflating: chest_xray/train/PNEUMONIA/person998_bacteria_2928.jpeg
  inflating: chest_xray/train/PNEUMONIA/person99_virus_183.jpeg
  inflating: chest_xray/train/PNEUMONIA/person9_bacteria_38.jpeg
  inflating: chest_xray/train/PNEUMONIA/person9_bacteria_39.jpeg
  inflating: chest_xray/train/PNEUMONIA/person9_bacteria_40.jpeg
  inflating: chest_xray/train/PNEUMONIA/person9_bacteria_41.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1427-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1430-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1431-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1436-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1437-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1438-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1440-0001.jpeg
  inflating: chest_xray/val/NORMAL/NORMAL2-IM-1442-0001.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1946_bacteria_4874.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1946_bacteria_4875.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1947_bacteria_4876.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1949_bacteria_4880.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1950_bacteria_4881.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1951_bacteria_4882.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1952_bacteria_4883.jpeg
  inflating: chest_xray/val/PNEUMONIA/person1954_bacteria_4886.jpeg
```

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
import os
import matplotlib.pyplot as plt
```

```


DATA_DIR = 'chest_xray'

train_dir = os.path.join(DATA_DIR, 'train')
val_dir = os.path.join(DATA_DIR, 'val')
test_dir = os.path.join(DATA_DIR, 'test')

IMAGE_SIZE = (150, 150)
BATCH_SIZE = 32
```

```
train_datagen = ImageDataGenerator(
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True
)


test_datagen = ImageDataGenerator(rescale=1./255)


train_generator = train_datagen.flow_from_directory(
    train_dir,
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='binary'
)

validation_generator = test_datagen.flow_from_directory(
    val_dir,
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='binary'
)


test_generator = test_datagen.flow_from_directory(
    test_dir,
    target_size=IMAGE_SIZE,
    batch_size=BATCH_SIZE,
    class_mode='binary',
    shuffle=False
)
```

```
Found 5216 images belonging to 2 classes.
Found 16 images belonging to 2 classes.
Found 624 images belonging to 2 classes.
```

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2, 2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(128, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')
])

model.compile(optimizer='adam',
              loss='binary_crossentropy',
              metrics=['accuracy'])


history = model.fit(
    train_generator,
    steps_per_epoch=train_generator.samples // BATCH_SIZE,
    epochs=10,
    validation_data=validation_generator,
    validation_steps=validation_generator.samples // BATCH_SIZE
)
```

```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarni
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

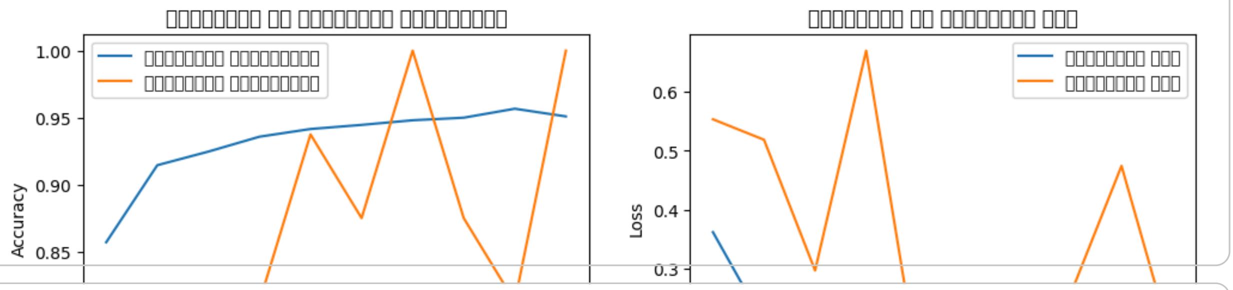
```
/usr/local/lib/python3.12/dist-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121
  self._warn_if_super_not_called()
Epoch 1/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 72s 414ms/step - accuracy: 0.7833 - loss: 0.6083 - val_accuracy: 0.75
Epoch 2/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 66s 402ms/step - accuracy: 0.9065 - loss: 0.2225 - val_accuracy: 0.75
Epoch 3/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 66s 406ms/step - accuracy: 0.9205 - loss: 0.2019 - val_accuracy: 0.81
Epoch 4/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 65s 398ms/step - accuracy: 0.9247 - loss: 0.1873 - val_accuracy: 0.81
Epoch 5/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 65s 400ms/step - accuracy: 0.9374 - loss: 0.1586 - val_accuracy: 0.93
Epoch 6/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 66s 404ms/step - accuracy: 0.9422 - loss: 0.1526 - val_accuracy: 0.87
Epoch 7/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 65s 398ms/step - accuracy: 0.9462 - loss: 0.1422 - val_accuracy: 1.00
Epoch 8/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 79s 488ms/step - accuracy: 0.9490 - loss: 0.1221 - val_accuracy: 0.87
Epoch 9/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 88s 538ms/step - accuracy: 0.9612 - loss: 0.1064 - val_accuracy: 0.81
Epoch 10/10
163/163 ━━━━━━━━━━━━━━━━━━━━ 86s 530ms/step - accuracy: 0.9494 - loss: 0.1288 - val_accuracy: 1.00
```

```
print("टेस्ट डेटा पर मॉडल का मूल्यांकन शुरू हो रहा है...")


loss, accuracy = model.evaluate(test_generator)

print(f"\n✅ टेस्ट लॉस (Test Loss): {loss:.4f}")
print(f"✅ टेस्ट एक्यूरेसी (Test Accuracy): {accuracy*100:.2f}%")
```

```
टेस्ट डेटा पर मॉडल का मूल्यांकन शुरू हो रहा है...
20/20 ━━━━━━━━━━━━━━━━━━━━ 5s 171ms/step - accuracy: 0.8058 - loss: 0.5383

✅ टेस्ट लॉस (Test Loss): 0.3438
✅ टेस्ट एक्यूरेसी (Test Accuracy): 87.18%
```

```
import matplotlib.pyplot as plt


plt.figure(figsize=(12, 4))
plt.subplot(1, 2, 1)
plt.plot(history.history['accuracy'], label='ट्रेनिंग एक्यूरेसी')
plt.plot(history.history['val_accuracy'], label='वैलिडेशन एक्यूरेसी')
plt.title('ट्रेनिंग और वैलिडेशन एक्यूरेसी')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()


plt.subplot(1, 2, 2)
plt.plot(history.history['loss'], label='ट्रेनिंग लॉस')
plt.plot(history.history['val_loss'], label='वैलिडेशन लॉस')
plt.title('ट्रेनिंग और वैलिडेशन लॉस')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()

plt.show()
```

```
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2335
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Matplotlib
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2381
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2352
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2375
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2344
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2367
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2306
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2327
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2324
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2357
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2376
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2354
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2337
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2358
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2319
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2325
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2351
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2370
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2360
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2368
  fig.canvas.print_figure(bytes_io, **kw)
/usr/local/lib/python3.12/dist-packages/IPython/core/pylabtools.py:151: UserWarning: Glyph 2377
  fig.canvas.print_figure(bytes_io, **kw)
```



```
model.save('pneumonia_detection_model.h5')
print("मॉडल सफलतापूर्वक 'pneumonia_detection_model.h5' नाम से सेव हो गया है।")
```

```
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_mo
मॉडल सफलतापूर्वक 'pneumonia_detection_model.h5' नाम से सेव हो गया है।
```

```
import numpy as np
from tensorflow.keras.preprocessing import image
import os

def predict_pneumonia(image_path, model):

    img = image.load_img(image_path, target_size=(150, 150))

    img_array = image.img_to_array(img)
    img_array = np.expand_dims(img_array, axis=0)

    img_array /= 255.0


    prediction = model.predict(img_array)


    if prediction[0][0] >= 0.5:
        result = "PNEUMONIA"
        probability = prediction[0][0] * 100
    else:
        result = "NORMAL"
        probability = (1 - prediction[0][0]) * 100

    return result, probability, img
```

```
test_image_path = os.path.join(test_dir, 'PNEUMONIA', 'person1_testing_10.jpeg')




if os.path.exists(test_image_path):
    result, prob, img = predict_pneumonia(test_image_path, model)


    plt.imshow(img)
    plt.title(f"Prediction: {result} (Confidence: {prob:.2f}%)")
    plt.axis('off')
    plt.show()
else:
    print(f"Error: फ़ाइल नहीं मिली -> {test_image_path}")
```

```
Error: फ़ाइल नहीं मिली -> chest_xray/test/PNEUMONIA/person1_testing_10.jpeg
```

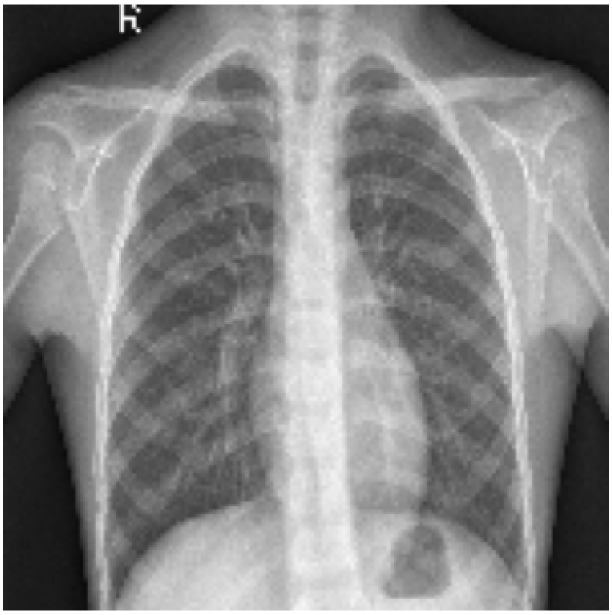
```
import os


DATA_DIR = 'chest_xray'


test_dir = os.path.join(DATA_DIR, 'test')
```

```
test_images, test_labels = next(test_generator)


predictions = model.predict(test_images)


predicted_class_index = (predictions[0][0] > 0.5).astype(int)


class_labels = list(test_generator.class_indices.keys())
predicted_label = class_labels[predicted_class_index]
actual_label = class_labels[int(test_labels[0])]


image_to_plot = (test_images[0] * 255).astype(np.uint8)


plt.imshow(image_to_plot)
plt.title(f"Predicted: {predicted_label} (Actual: {actual_label})")
```

```
    plt.axis('off')
    plt.show()
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 309ms/step
```

Predicted: NORMAL (Actual: NORMAL)



```
import tensorflow as tf


loaded_model = tf.keras.models.load_model('pneumonia_detection_model.h5')

print("मॉडल सफलतापूर्वक लोड हो गया है।")
```

```
WARNING:absl:Compiled the loaded model, but the compiled metrics have yet to be built. `model.comp
मॉडल सफलतापूर्वक लोड हो गया है।
```

```
from sklearn.metrics import classification_report, confusion_matrix
import numpy as np


test_generator.reset()
Y_pred = loaded_model.predict(test_generator, steps=test_generator.samples // test_generator.batch


y_pred_classes = (Y_pred > 0.5).astype(int)


true_classes = test_generator.classes
class_labels = list(test_generator.class_indices.keys())


print("### कन्फ्यूजन मैट्रिक्स ###")
conf_matrix = confusion_matrix(true_classes, y_pred_classes)
print(conf_matrix)


print("\n### क्लासिफिकेशन रिपोर्ट ###")
report = classification_report(true_classes, y_pred_classes, target_names=class_labels)
print(report)
```

```
20/20 ━━━━━━━━━━━━━━━━━━━━ 5s 189ms/step
### कन्फ्यूजन मैट्रिक्स ###
[[167  67]
 [ 13 377]]
```

```
### क्लासिफिकेशन रिपोर्ट ###
              precision    recall  f1-score   support

      NORMAL       0.93      0.71      0.81       234
   PNEUMONIA       0.85      0.97      0.90       390

    accuracy                           0.87       624
   macro avg       0.89      0.84      0.86       624
weighted avg       0.88      0.87      0.87       624
```