

DATASET RADIOGRAFÍAS

R

¿Cómo desarrollar un proyecto en Python con Machine Learning?

Autor: Jonathan Emmanuel Alvarez Salazar

BOOTCAMP: Ciencia de datos e inteligencia
artificial

Índice

- 1. Abstracto con motivación y audiencia**
- 2. Hipótesis/Preguntas a responder**
- 3. Resumen de Metadata**
- 4. Visualizaciones**
- 5. Insight y hallazgos**

Abstracto con motivación y audiencia

Nuestra audiencia puede definirse por clínicas o laboratorios que ofrecen servicios de rayos-X

Deseamos investigar 2 grupos de personas, unas personas que no presenta ninguna afectación por Neumonía y el otro grupo presente Neumonía por diferentes orígenes aunque muestran mucha similitud en sus respectivas radiografías

Los datos son extraídos de una [API publica de Kaggle](#)

Hipótesis/ preguntas a responder

¿Qué precisión tendrá un sistema de inteligencia artificial para clasificar las imágenes?

¿Se puede utilizar el aprendizaje automático para identificar la neumonía en radiografías de tórax?

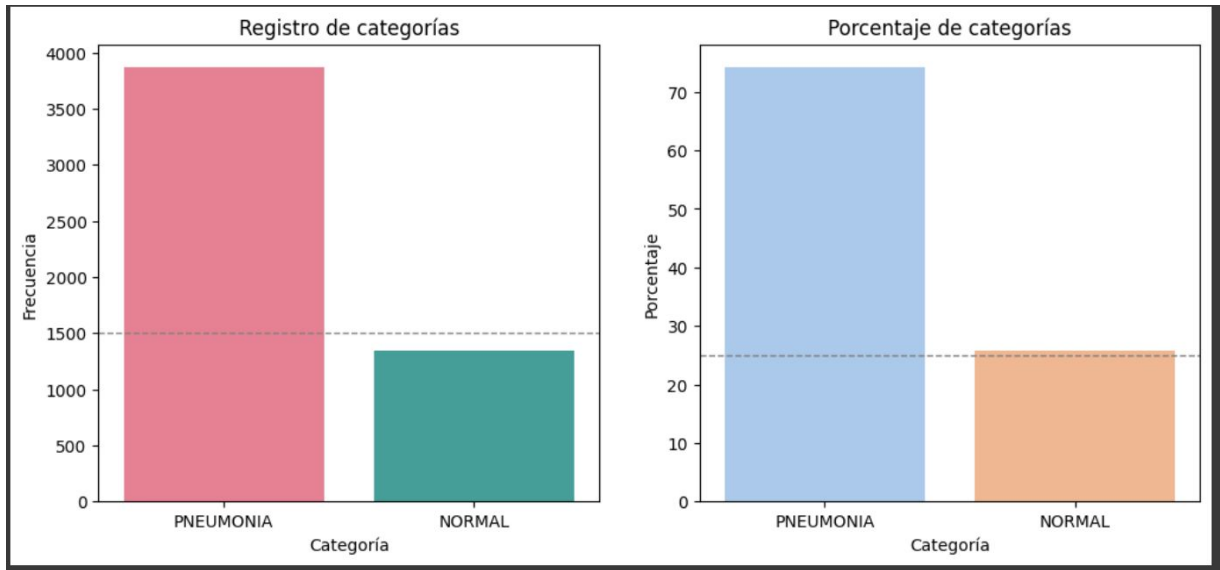
¿Se puede utilizar el aprendizaje automático para identificar diferentes tipos de neumonía en radiografías de tórax?

¿Se puede utilizar el aprendizaje automático para predecir la gravedad de la neumonía en pacientes con radiografías de tórax?

¿Se puede utilizar el aprendizaje automático para desarrollar herramientas de detección temprana de la neumonía que puedan ser utilizadas por personal no médico en zonas rurales o con acceso limitado a la atención médica?

- 1.-Con el conjunto de datos de radiografías de torax validadas por un profesional del área médica, con las librerías para desarrollar modelos con imágenes y TensorFlow
- 2.-Se puede desarrollar un modelo de aprendizaje automático que pueda favorecer el diagnóstico de neumonía con mayor precisión, con un conjunto de datos etiquetados "con neumonía" y "sin neumonía".
- 3.-Una increíble idea sería desarrollar un modelo de aprendizaje para cada tipo de neumonía (3 clases de neumonía), aunque es favorable auxiliarse de los análisis clínicos
- 4.-Necesitamos acceder a un conjunto de datos de radiografías de torax, etiquetas y evaluadas por profesionales definiendo la gravedad de la neumonía (leve, moderada o grave)
- 5.-Podríamos generar un API/página o interfaz para que una persona del área de salud, subir la imagen al "sistema/red/modelo" y facilitar su diagnóstico temprano o canalizarlo a laboratorio evitando la burocracia.

RESUMEN METADATA



Tenemos la visualización del conteo de las radiografías de tórax de personas "normal" y con "Pneumonia"

Nos muestra la distribución y porcentajes en cantidad del dataset de cada categoría respectivamente

Accede al siguiente notebook para reproducir el [análisis](#)

VISUALIZACIONES

Redes neuronales Convolucionales

Redes neuronales convolucionales:

Se utilizan para procesar datos espaciales, como imágenes. Contienen capas convolucionales que extraen características de los datos.

Se utilizan para tareas como clasificación de imágenes, detección de objetos y segmentación de imágenes. Esto es útil cuando se utiliza la técnica de transferencia de aprendizaje, donde se desea aprovechar el conocimiento preentrenado de un modelo como Xception.

Al congelar las capas del modelo base, se evita que se sobreajusten a su conjunto de datos específico, lo que puede mejorar el rendimiento general.

Batch Normalization es una técnica de normalización utilizada en redes neuronales profundas para acelerar el entrenamiento y mejorar la estabilidad.

```
#Armamos el modelo convulucional

model=Sequential()
model.add(modelo_base)
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(240,activation='relu'))
model.add(Dropout(0.25))

model.add(Dense(40,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
```



```
147/147 [=====] - 15s 99ms/step - loss: 0.1223 - accuracy: 0.9508 - val_loss: 0.0733 - val_accuracy: 0.9712
Epoch 19/20
147/147 [=====] - 15s 99ms/step - loss: 0.1289 - accuracy: 0.9478 - val_loss: 0.0710 - val_accuracy: 0.9750
Epoch 20/20
147/147 [=====] - 15s 97ms/step - loss: 0.1213 - accuracy: 0.9521 - val_loss: 0.0748 - val_accuracy: 0.9731
```

```
[54] validation_loss,validation_accuracy=model.evaluate(dataset_val)
```

```
print("validation_loss:",validation_loss)
print("validation_accuracy:",validation_accuracy)
```

```
17/17 [=====] - 2s 58ms/step - loss: 0.0748 - accuracy: 0.9731
validation_loss: 0.07482229173183441
validation_accuracy: 0.973128616809845
```

```
[55] mejor_epoch=history.history["val_accuracy"].index(max(history.history["val_accuracy"]))+1
print(f'El mejor epoch es el numero:{mejor_epoch}')
```

```
El mejor epoch es el numero:19
```

Verdadero: PNEUMONIA
Predicción: NORMAL



Verdadero: PNEUMONIA
Predicción: NORMAL



Verdadero: PNEUMONIA
Predicción: NORMAL



Verdadero: PNEUMONIA
Predicción: NORMAL



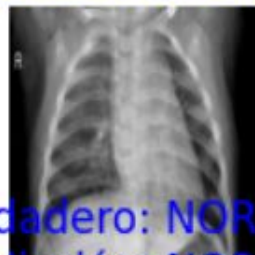
Verdadero: PNEUMONIA
Predicción: NORMAL



Verdadero: PNEUMONIA
Predicción: NORMAL



Verdadero: NORMAL
Predicción: NORMAL



Verdadero: NORMAL
Predicción: NORMAL



Verdadero: NORMAL
Predicción: NORMAL



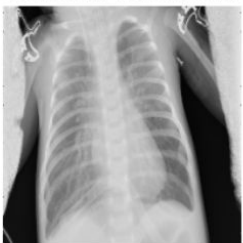
Predicted: PNEUMONIA



Predicted: PNEUMONIA



Predicted: PNEUMONIA



Predicted: PNEUMONIA



Predicted: PNEUMONIA



Predicted: PNEUMONIA



Predicted: PNEUMONIA



#Hacer predicciones

```
predicciones_2 = model.predict(img_array)
actual_prediction = (predicciones_2 > 0.5).astype(int)
```

#Mostar la imagen con su etiqueta de prediccion

```
axs[i].imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
axs[i].axis('off')
if actual_prediction[0][0] == 0:
    predicted_label = 'Normal'
else:
    predicted_label = 'PNEUMONIA'
axs[i].set_title(f'Predicted: {predicted_label}')
```