Modelos_Redes_Neuronales

April 14, 2025

<td><img src=https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTZcU0_2fgIIcOXf6T0U0AQq1Ld4v

<td><td style="font-size:3vw"><font color='OrangeRed'>Trabajo Final Master en Big Data<

<td><img src=https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTZcU0_2fgIIcOXf6T0U0AQq1Ld4v

<td><td style="font-size:3vw"><font color='OrangeRed'>Trabajo Final Master en Big Data<

1 Trabajo Final - Máster en Big Data - Fernández, García y Payovich

1.1 Modelos de Base: Redes Neuronales

1.1.1 Importación de Librerías

[1]:
```
import pandas as pd
import numpy as np
import os
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split # División del dataset
import matplotlib.pyplot as plt # Plotear
from sklearn.linear_model import LinearRegression # Modelo de ML
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.compose import make_column_transformer
from sklearn import metrics
from sklearn.metrics import precision_score, recall_score, f1_score,␣
↪accuracy_score, confusion_matrix, classification_report # Métricas
pd.options.display.max_columns= None
import warnings
from openpyxl import Workbook
```

[2]:
```
warnings.filterwarnings('ignore')
```

1.1.2 Importación y Transformación de las Bases

[3]:
```
df = pd.read_csv('base_final_EDA.csv', delimiter= ',')
```

[4]:
```
dfr = df.astype(int)
```

[5]:
```
dfpca = pd.read_csv('base_final_PCA.csv', delimiter= ';')
```

[6]:
```
x_r = dfr.drop(['detractor'], axis=1)
```

[7]:
```
y_r = dfr['detractor']
```

[8]:
```
x_pca = dfpca.drop(['detractor'], axis=1)
```

[9]:
```
y_pca = dfpca['detractor']
```

[10]:
```
len(x_r.columns)
```

[10]: 37

[11]:
```
len(x_pca.columns)
```

[11]: 23

1.1.3 Sampleo y Escalado de Datos:

[13]:
```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 42)
x_r_sm, y_r_sm = sm.fit_resample(x_r, y_r)
```

[14]:
```
from imblearn.over_sampling import SMOTE
sm = SMOTE(random_state = 42)
x_pca_sm, y_pca_sm = sm.fit_resample(x_pca, y_pca)
```

[15]:
```
x_train_r, x_test_r, y_train_r, y_test_r = train_test_split(x_r_sm, y_r_sm,
                                    random_state=104,
                                    test_size=0.20,
                                    shuffle=True)
```

[16]:
```
x_train_pca, x_test_pca, y_train_pca, y_test_pca = train_test_split(x_pca_sm,␣
↪y_pca_sm,
                                    random_state=104,
                                    test_size=0.20,
                                    shuffle=True)
```

1.1.4 Red de 2 Capas:

[17]:
```
import tensorflow.keras as keras
import tensorflow as tf
from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.metrics import Recall
from sklearn.metrics import accuracy_score, precision_score, recall_score,␣
 ↪f1_score, confusion_matrix, roc_auc_score

model_r = keras.Sequential([
      Dense(32, activation='relu', input_shape=(37,)),
      Dense(1, activation='sigmoid')
])

model_r.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=[Recall()])

model_r.fit(x_train_r, y_train_r, epochs=50, batch_size=8, verbose=1)

y_pred_r = model_r.predict(x_test_r).round()

loss, recall = model_r.evaluate(x_test_r, y_test_r)
print('Loss del modelo:{loss}')
print('Recall del modelo:{recall}')
print(classification_report(y_test_r, y_pred_r))
```

```
2025-04-14 15:51:37.004862: W
tensorflow/stream_executor/platform/default/dso_loader.cc:55] Could not load
dynamic library 'libcuda.so.1'; dlerror: /usr/lib/x86_64-linux-gnu/libcuda.so.1:
file too short; LD_LIBRARY_PATH: /usr/local/cuda/extras/CUPTI/lib64:/usr/local/c
uda/lib64:/usr/local/nvidia/lib:/usr/local/nvidia/lib64
2025-04-14 15:51:37.004899: E
tensorflow/stream_executor/cuda/cuda_driver.cc:313] failed call to cuInit:
UNKNOWN ERROR (303)
2025-04-14 15:51:37.004925: I
tensorflow/stream_executor/cuda/cuda_diagnostics.cc:163] no NVIDIA GPU device is
present: /dev/nvidia0 does not exist
2025-04-14 15:51:37.005068: I tensorflow/core/platform/cpu_feature_guard.cc:143]
Your CPU supports instructions that this TensorFlow binary was not compiled to
use: AVX2 AVX512F FMA
2025-04-14 15:51:37.034396: I
tensorflow/core/platform/profile_utils/cpu_utils.cc:102] CPU Frequency:
2300000000 Hz
2025-04-14 15:51:37.037592: I tensorflow/compiler/xla/service/service.cc:168]
XLA service 0x7f0eb4000b20 initialized for platform Host (this does not
guarantee that XLA will be used). Devices:
```

```
2025-04-14 15:51:37.037714: I tensorflow/compiler/xla/service/service.cc:176]
StreamExecutor device (0): Host, Default Version

Epoch 1/50
1963/1963 [==============================] - 3s 1ms/step - loss: 8.4027 -
recall: 0.5267
Epoch 2/50
1963/1963 [==============================] - 3s 1ms/step - loss: 3.5327 -
recall: 0.5674
Epoch 3/50
1963/1963 [==============================] - 3s 1ms/step - loss: 3.0493 -
recall: 0.5813
Epoch 4/50
1963/1963 [==============================] - 3s 1ms/step - loss: 3.2100 -
recall: 0.5854
Epoch 5/50
1963/1963 [==============================] - 3s 1ms/step - loss: 3.1410 -
recall: 0.6008
Epoch 6/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.9394 -
recall: 0.6131
Epoch 7/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.6396 -
recall: 0.6214
Epoch 8/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.9250 -
recall: 0.6163
Epoch 9/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.8484 -
recall: 0.6172
Epoch 10/50
1963/1963 [==============================] - 2s 1ms/step - loss: 2.7843 -
recall: 0.6255
Epoch 11/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.4562 -
recall: 0.6512
Epoch 12/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.6030 -
recall: 0.6265
Epoch 13/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.6203 -
recall: 0.6329
Epoch 14/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.6789 -
recall: 0.6397
Epoch 15/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.3543 -
recall: 0.6404
```

```
Epoch 16/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.5327 -
recall: 0.6408
Epoch 17/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.5743 -
recall: 0.6345
Epoch 18/50
1963/1963 [==============================] - 2s 1ms/step - loss: 2.5793 -
recall: 0.6452
Epoch 19/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.2804 -
recall: 0.6497
Epoch 20/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.3014 -
recall: 0.6474: 0s - loss: 2.2974 - recall: 0.64
Epoch 21/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.1945 -
recall: 0.6487
Epoch 22/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.2905 -
recall: 0.6475
Epoch 23/50
1963/1963 [==============================] - 2s 1ms/step - loss: 2.0188 -
recall: 0.6593
Epoch 24/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.0095 -
recall: 0.6614
Epoch 25/50
1963/1963 [==============================] - 3s 2ms/step - loss: 2.0071 -
recall: 0.6579
Epoch 26/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.9034 -
recall: 0.6565
Epoch 27/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.0267 -
recall: 0.6571
Epoch 28/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.8114 -
recall: 0.6608
Epoch 29/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.8925 -
recall: 0.6534
Epoch 30/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.8285 -
recall: 0.6598
Epoch 31/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.6550 -
recall: 0.6704
```

```
Epoch 32/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.6920 -
recall: 0.6642
Epoch 33/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.7239 -
recall: 0.6672
Epoch 34/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.6207 -
recall: 0.6665
Epoch 35/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.5570 -
recall: 0.6639
Epoch 36/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.4616 -
recall: 0.6625
Epoch 37/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.3925 -
recall: 0.6678
Epoch 38/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.4063 -
recall: 0.6716
Epoch 39/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.3318 -
recall: 0.6710
Epoch 40/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.2922 -
recall: 0.6713
Epoch 41/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.1732 -
recall: 0.6718
Epoch 42/50
1963/1963 [==============================] - 2s 1ms/step - loss: 1.1998 -
recall: 0.6697
Epoch 43/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.1969 -
recall: 0.6771
Epoch 44/50
1963/1963 [==============================] - 2s 1ms/step - loss: 1.0614 -
recall: 0.6770
Epoch 45/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.0213 -
recall: 0.6760
Epoch 46/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.9596 -
recall: 0.6790
Epoch 47/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.0443 -
recall: 0.6737
```

```
Epoch 48/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.0305 -
recall: 0.6817
Epoch 49/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.9959 -
recall: 0.6756
Epoch 50/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.9400 -
recall: 0.6776
123/123 [==============================] - 0s 941us/step - loss: 0.5606 -
recall: 0.7152
Loss del modelo:{loss}
Recall del modelo:{recall}
              precision    recall  f1-score   support

           0       0.72      0.74      0.73      1952
           1       0.74      0.72      0.73      1973

    accuracy                           0.73      3925
   macro avg       0.73      0.73      0.73      3925
weighted avg       0.73      0.73      0.73      3925
```

[18]:
```
model_pca = keras.Sequential([
      Dense(32, activation='relu', input_shape=(23,)),
      Dense(1, activation='sigmoid')
])

model_pca.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=[Recall()])

model_pca.fit(x_train_pca, y_train_pca, epochs=50, batch_size=8, verbose=1)

y_pred_pca = model_pca.predict(x_test_pca).round()

print(classification_report(y_test_pca, y_pred_pca))
```

```
Epoch 1/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6719 -
recall_1: 0.6272
Epoch 2/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6540 -
recall_1: 0.6426
Epoch 3/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6481 -
recall_1: 0.6554
Epoch 4/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6422 -
```

```
recall_1: 0.6730
Epoch 5/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6383 -
recall_1: 0.6673
Epoch 6/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6349 -
recall_1: 0.6654
Epoch 7/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6314 -
recall_1: 0.6679
Epoch 8/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6284 -
recall_1: 0.6741
Epoch 9/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6261 -
recall_1: 0.6811
Epoch 10/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6244 -
recall_1: 0.6808
Epoch 11/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6217 -
recall_1: 0.6875
Epoch 12/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6200 -
recall_1: 0.6859
Epoch 13/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6190 -
recall_1: 0.7024
Epoch 14/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6167 -
recall_1: 0.6926
Epoch 15/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6150 -
recall_1: 0.7003
Epoch 16/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6127 -
recall_1: 0.7005
Epoch 17/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6122 -
recall_1: 0.7093
Epoch 18/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6099 -
recall_1: 0.7005
Epoch 19/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6090 -
recall_1: 0.7033
Epoch 20/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6071 -
```

```
recall_1: 0.7063
Epoch 21/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6050 -
recall_1: 0.7114
Epoch 22/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6033 -
recall_1: 0.7095
Epoch 23/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6032 -
recall_1: 0.7177
Epoch 24/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6025 -
recall_1: 0.7168
Epoch 25/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6015 -
recall_1: 0.7086
Epoch 26/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6002 -
recall_1: 0.7168
Epoch 27/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5988 -
recall_1: 0.7131
Epoch 28/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5985 -
recall_1: 0.7211
Epoch 29/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5973 -
recall_1: 0.7141
Epoch 30/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5966 -
recall_1: 0.7208
Epoch 31/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5959 -
recall_1: 0.7261
Epoch 32/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5956 -
recall_1: 0.7220
Epoch 33/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5947 -
recall_1: 0.7192
Epoch 34/50
1963/1963 [==============================] - 2s 997us/step - loss: 0.5948 -
recall_1: 0.7204
Epoch 35/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5923 -
recall_1: 0.7233
Epoch 36/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5926 -
```

```
recall_1: 0.7225
Epoch 37/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5912 -
recall_1: 0.7307
Epoch 38/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5906 -
recall_1: 0.7256
Epoch 39/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5891 -
recall_1: 0.7293
Epoch 40/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5893 -
recall_1: 0.7253
Epoch 41/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5888 -
recall_1: 0.7386
Epoch 42/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5886 -
recall_1: 0.7299
Epoch 43/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5874 -
recall_1: 0.7290
Epoch 44/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5870 -
recall_1: 0.7312
Epoch 45/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5866 -
recall_1: 0.7333
Epoch 46/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5857 -
recall_1: 0.7373
Epoch 47/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5854 -
recall_1: 0.7310
Epoch 48/50
1963/1963 [==============================] - 2s 997us/step - loss: 0.5853 -
recall_1: 0.7329
Epoch 49/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5842 -
recall_1: 0.7331
Epoch 50/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5838 -
recall_1: 0.7284
              precision    recall  f1-score   support

       False       0.71      0.56      0.63      1952
        True       0.64      0.77      0.70      1973
```

```
    accuracy                           0.67      3925
   macro avg       0.67      0.67      0.66      3925
weighted avg       0.67      0.67      0.66      3925
```

1.1.5 Red de 3 Capas:

[19]:
```
model_r3 = keras.Sequential([
      Dense(64, activation='relu', input_shape=(37,)),
      Dense(8, activation='relu'),
      Dense(1, activation='sigmoid')
])

model_r3.compile(optimizer='adam', loss='binary_crossentropy',␣
 ↪metrics=[Recall()])

model_r3.fit(x_train_r, y_train_r, epochs=50, batch_size=8, verbose=1)

y_pred_r3 = model_r3.predict(x_test_r).round()

loss, recall = model_r3.evaluate(x_test_r, y_test_r)
print(classification_report(y_test_r, y_pred_r))
```

```
Epoch 1/50
1963/1963 [==============================] - 3s 1ms/step - loss: 5.6116 -
recall_2: 0.5397
Epoch 2/50
1963/1963 [==============================] - 3s 1ms/step - loss: 3.3900 -
recall_2: 0.5714
Epoch 3/50
1963/1963 [==============================] - 2s 1ms/step - loss: 3.0896 -
recall_2: 0.5877
Epoch 4/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.1799 -
recall_2: 0.6030
Epoch 5/50
1963/1963 [==============================] - 3s 1ms/step - loss: 2.0505 -
recall_2: 0.6089
Epoch 6/50
1963/1963 [==============================] - 3s 2ms/step - loss: 1.4758 -
recall_2: 0.6279
Epoch 7/50
1963/1963 [==============================] - 3s 1ms/step - loss: 1.1847 -
recall_2: 0.6382
Epoch 8/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.8973 -
recall_2: 0.6357
```

```
Epoch 9/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.7013 -
recall_2: 0.6665
Epoch 10/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6612 -
recall_2: 0.6665
Epoch 11/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6689 -
recall_2: 0.6586
Epoch 12/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6721 -
recall_2: 0.7534: 0s - loss: 0.6733 - reca
Epoch 13/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6767 -
recall_2: 0.8180
Epoch 14/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6762 -
recall_2: 0.6852
Epoch 15/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6753 -
recall_2: 0.3757
Epoch 16/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6762 -
recall_2: 0.0885
Epoch 17/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6761 -
recall_2: 0.0857
Epoch 18/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6751 -
recall_2: 0.1994
Epoch 19/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6739 -
recall_2: 0.3170
Epoch 20/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6758 -
recall_2: 0.0929
Epoch 21/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6744 -
recall_2: 0.2077
Epoch 22/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6735 -
recall_2: 0.2065
Epoch 23/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6717 -
recall_2: 0.0865
Epoch 24/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6741 -
recall_2: 0.0890
```

```
Epoch 25/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6739 -
recall_2: 0.0916
Epoch 26/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6711 -
recall_2: 0.0890
Epoch 27/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6691 -
recall_2: 0.1143
Epoch 28/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6696 -
recall_2: 0.3212
Epoch 29/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6673 -
recall_2: 0.0910
Epoch 30/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6674 -
recall_2: 0.2484
Epoch 31/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6665 -
recall_2: 0.3310
Epoch 32/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6664 -
recall_2: 0.5822
Epoch 33/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6660 -
recall_2: 0.2602
Epoch 34/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6670 -
recall_2: 0.3985
Epoch 35/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6656 -
recall_2: 0.5478
Epoch 36/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6635 -
recall_2: 0.2447
Epoch 37/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6662 -
recall_2: 0.1156
Epoch 38/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6667 -
recall_2: 0.3633
Epoch 39/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6665 -
recall_2: 0.1142
Epoch 40/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6625 -
recall_2: 0.1346
```

```
Epoch 41/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6620 -
recall_2: 0.1533
Epoch 42/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6587 -
recall_2: 0.3139
Epoch 43/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6596 -
recall_2: 0.9329
Epoch 44/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6588 -
recall_2: 0.6790
Epoch 45/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6570 -
recall_2: 0.9259
Epoch 46/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6601 -
recall_2: 0.7438
Epoch 47/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6507 -
recall_2: 0.8529
Epoch 48/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.6505 -
recall_2: 0.8289
Epoch 49/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6435 -
recall_2: 0.7728
Epoch 50/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6386 -
recall_2: 0.7577
123/123 [==============================] - 0s 994us/step - loss: 0.6600 -
recall_2: 0.9346
              precision    recall  f1-score   support

           0       0.72      0.74      0.73      1952
           1       0.74      0.72      0.73      1973

    accuracy                           0.73      3925
   macro avg       0.73      0.73      0.73      3925
weighted avg       0.73      0.73      0.73      3925
```

[20]:
```
model_p3 = keras.Sequential([
      Dense(64, activation='relu', input_shape=(23,)),
      Dense(8, activation='relu'),
      Dense(1, activation='sigmoid')
])
```

```
model_p3.compile(optimizer='adam', loss='binary_crossentropy',␣
↪metrics=[Recall()])

model_p3.fit(x_train_pca, y_train_pca, epochs=50, batch_size=8, verbose=1)

y_pred_p3 = model_p3.predict(x_test_pca).round()

loss, recall = model_p3.evaluate(x_test_pca, y_test_pca)
print(classification_report(y_test_pca, y_pred_pca))
```

```
Epoch 1/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6688 -
recall_3: 0.6183
Epoch 2/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6479 -
recall_3: 0.6609
Epoch 3/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6374 -
recall_3: 0.6653
Epoch 4/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.6277 -
recall_3: 0.6807
Epoch 5/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6180 -
recall_3: 0.6906
Epoch 6/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6100 -
recall_3: 0.7071
Epoch 7/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.6029 -
recall_3: 0.7183
Epoch 8/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5971 -
recall_3: 0.7250
Epoch 9/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5893 -
recall_3: 0.7352
Epoch 10/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5847 -
recall_3: 0.7375
Epoch 11/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5779 -
recall_3: 0.7447
Epoch 12/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5729 -
recall_3: 0.7521
```

```
Epoch 13/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5680 -
recall_3: 0.7589
Epoch 14/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5630 -
recall_3: 0.7622
Epoch 15/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5570 -
recall_3: 0.7622
Epoch 16/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5519 -
recall_3: 0.7690
Epoch 17/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5507 -
recall_3: 0.7634
Epoch 18/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5442 -
recall_3: 0.7740
Epoch 19/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5413 -
recall_3: 0.7788
Epoch 20/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5396 -
recall_3: 0.7719
Epoch 21/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5359 -
recall_3: 0.7770
Epoch 22/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5318 -
recall_3: 0.7815
Epoch 23/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5296 -
recall_3: 0.7834
Epoch 24/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5257 -
recall_3: 0.7879
Epoch 25/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5242 -
recall_3: 0.7859
Epoch 26/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5213 -
recall_3: 0.7945
Epoch 27/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5165 -
recall_3: 0.7900
Epoch 28/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5150 -
recall_3: 0.7981
```

```
Epoch 29/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5133 -
recall_3: 0.7977
Epoch 30/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5124 -
recall_3: 0.7960
Epoch 31/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5114 -
recall_3: 0.7944
Epoch 32/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.5072 -
recall_3: 0.8030
Epoch 33/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.5043 -
recall_3: 0.8070
Epoch 34/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.5039 -
recall_3: 0.8042
Epoch 35/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4998 -
recall_3: 0.8076
Epoch 36/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4976 -
recall_3: 0.8053:
Epoch 37/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4983 -
recall_3: 0.8103
Epoch 38/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4955 -
recall_3: 0.8111
Epoch 39/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4947 -
recall_3: 0.8090
Epoch 40/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4937 -
recall_3: 0.8112
Epoch 41/50
1963/1963 [==============================] - 2s 1ms/step - loss: 0.4913 -
recall_3: 0.8118
Epoch 42/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4887 -
recall_3: 0.8129
Epoch 43/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4905 -
recall_3: 0.8158
Epoch 44/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4876 -
recall_3: 0.8164
```

```
Epoch 45/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4869 -
recall_3: 0.8206
Epoch 46/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4838 -
recall_3: 0.8249
Epoch 47/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4844 -
recall_3: 0.8226
Epoch 48/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4840 -
recall_3: 0.8228: 0s - loss: 0.4800 - recall
Epoch 49/50
1963/1963 [==============================] - 3s 1ms/step - loss: 0.4811 -
recall_3: 0.8197
Epoch 50/50
1963/1963 [==============================] - 3s 2ms/step - loss: 0.4826 -
recall_3: 0.8254
123/123 [==============================] - 0s 973us/step - loss: 0.5942 -
recall_3: 0.8363
              precision    recall  f1-score   support

       False       0.71      0.56      0.63      1952
        True       0.64      0.77      0.70      1973

    accuracy                           0.67      3925
   macro avg       0.67      0.67      0.66      3925
weighted avg       0.67      0.67      0.66      3925
```

1.1.6 Weights and Biases

[23]:
```
pip install wandb
```

```
Requirement already satisfied: wandb in /opt/conda/lib/python3.7/site-packages
(0.12.7)
Requirement already satisfied: sentry-sdk>=1.0.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (1.5.0)
Requirement already satisfied: yaspin>=1.0.0 in /opt/conda/lib/python3.7/site-
packages (from wandb) (2.1.0)
Requirement already satisfied: six>=1.13.0 in /opt/conda/lib/python3.7/site-
packages (from wandb) (1.16.0)
Requirement already satisfied: psutil>=5.0.0 in /opt/conda/lib/python3.7/site-
packages (from wandb) (5.8.0)
Requirement already satisfied: docker-pycreds>=0.4.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (0.4.0)
Requirement already satisfied: PyYAML in /opt/conda/lib/python3.7/site-packages
```

```
(from wandb) (6.0)
Requirement already satisfied: subprocess32>=3.5.3 in
/opt/conda/lib/python3.7/site-packages (from wandb) (3.5.4)
Requirement already satisfied: promise<3,>=2.0 in /opt/conda/lib/python3.7/site-
packages (from wandb) (2.3)
Requirement already satisfied: shortuuid>=0.5.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (1.0.8)
Requirement already satisfied: pathtools in /opt/conda/lib/python3.7/site-
packages (from wandb) (0.1.2)
Requirement already satisfied: protobuf>=3.12.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (3.18.1)
Requirement already satisfied: GitPython>=1.0.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (3.1.24)
Requirement already satisfied: configparser>=3.8.1 in
/opt/conda/lib/python3.7/site-packages (from wandb) (5.1.0)
Requirement already satisfied: python-dateutil>=2.6.1 in
/opt/conda/lib/python3.7/site-packages (from wandb) (2.8.2)
Requirement already satisfied: requests<3,>=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (2.26.0)
Requirement already satisfied: Click!=8.0.0,>=7.0 in
/opt/conda/lib/python3.7/site-packages (from wandb) (7.1.2)
Requirement already satisfied: typing-extensions>=3.7.4.3 in
/opt/conda/lib/python3.7/site-packages (from GitPython>=1.0.0->wandb) (4.0.0)
Requirement already satisfied: gitdb<5,>=4.0.1 in /opt/conda/lib/python3.7/site-
packages (from GitPython>=1.0.0->wandb) (4.0.9)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/opt/conda/lib/python3.7/site-packages (from requests<3,>=2.0.0->wandb) (2.0.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in
/opt/conda/lib/python3.7/site-packages (from requests<3,>=2.0.0->wandb) (1.26.7)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/lib/python3.7/site-
packages (from requests<3,>=2.0.0->wandb) (3.1)
Requirement already satisfied: certifi>=2017.4.17 in
/opt/conda/lib/python3.7/site-packages (from requests<3,>=2.0.0->wandb)
(2021.10.8)
Requirement already satisfied: termcolor<2.0.0,>=1.1.0 in
/opt/conda/lib/python3.7/site-packages (from yaspin>=1.0.0->wandb) (1.1.0)
Requirement already satisfied: smmap<6,>=3.0.1 in /opt/conda/lib/python3.7/site-
packages (from gitdb<5,>=4.0.1->GitPython>=1.0.0->wandb) (5.0.0)
Note: you may need to restart the kernel to use updated packages.
```

[27]:
```
import wandb
wandb.login()
```

```
Failed to detect the name of this notebook, you can set it manually with the
WANDB_NOTEBOOK_NAME environment variable to enable code saving.
wandb: Currently logged in as: ort_masterbigdata_2023 (use
`wandb login --relogin` to force relogin)
```

[27]: True

[28]:
```
project = 'trabajo_final_nps_itau'
wandb.init(project=project)

# Hiperparámetros
config = wandb.config
config.learning_rate = 0.01
config.epochs = 50
config.batch_size = 32

recall = recall_score(y_test_r, y_pred_r)
wandb.log({"recall": recall})

# Ejemplo de entrenamiento dummy
for epoch in range(config.epochs):
    loss = 0.01 * epoch  # Simulación de pérdida
    acc = 1 - loss       # Simulación de accuracy
    recall = recall
    wandb.log({"epoch": epoch, "loss": loss, "accuracy": acc, "recall":recall})
```

```
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

<IPython.core.display.HTML object>

[29]:
```
wandb.save('red_3capas.h5')
```

[29]: []

Optimización de Parámetros

[30]:
```
def train():
    # Inicializar wandb y acceder a los hiperparámetros
    with wandb.init() as run:
        config = wandb.config

        for epoch in range(10):
            loss = 1 / (config.learning_rate * (epoch + 1))
            wandb.log({"epoch": epoch, "loss": loss, "recall":recall})
```

[31]:
```
sweep_config = {
    'method': 'random',  # también puede ser 'grid' o 'bayes'
    'metric': {
        'name': 'loss',
        'goal': 'minimize'
    },
```

```
    'parameters': {
        'learning_rate': {
            'min': 0.0001,
            'max': 0.1
        },
        'batch_size': {
            'values': [16, 32, 64]
        }
    }
}
```

[32]:
```
sweep_id = wandb.sweep(sweep=sweep_config, project=project)
```

```
wandb: WARNING Calling wandb.login() after wandb.init()
has no effect.

Create sweep with ID: frjklye1
Sweep URL:
https://wandb.ai/ort_masterbigdata_2023/trabajo_final_nps_itau/sweeps/frjklye1
```

[33]:
```
wandb.agent(sweep_id, function=train, count=20)
```

```
wandb: WARNING Calling wandb.login() after wandb.init()
has no effect.
wandb: Agent Starting Run: 9iwk84nk with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.002860281813977048
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: m1bhpthg with config:
wandb:     batch_size: 32
wandb:     learning_rate: 0.020796526213385503
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…
```

```
<IPython.core.display.HTML object>

wandb: Agent Starting Run: aex8v7ol with config:
wandb:     batch_size: 32
wandb:     learning_rate: 0.018208359917665905
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: bfjdsclq with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.09248747925890678
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 7gh8lfav with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.04266239770153933
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: qgfob2kz with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.027517625244927516
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: hohx6sut with config:
wandb:     batch_size: 32
wandb:     learning_rate: 0.08134934005265824
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: iw2wf28e with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.002887532327763576
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 74u6n58i with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.03656111298503896
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: hiz2i4n1 with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.06265314245478716
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 9z5tv4fu with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.015729622983009152
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 874d3rzo with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.06420059641336172
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 7ie6tzn5 with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.03666642823682552
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade

<IPython.core.display.HTML object>

<IPython.core.display.HTML object>

VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…

<IPython.core.display.HTML object>

wandb: Agent Starting Run: 5axsar3z with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.09056518332141192
```

```
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: 64msyqrn with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.0643353035678868
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: vj9p6fdk with config:
wandb:     batch_size: 32
wandb:     learning_rate: 0.06844088868538194
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: lcrhez96 with config:
wandb:     batch_size: 32
wandb:     learning_rate: 0.05094818987681296
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: gve7dp91 with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.004157729739572018
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=0.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: a1r3e6e2 with config:
wandb:     batch_size: 64
wandb:     learning_rate: 0.003408442400140216
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```

```
wandb: Agent Starting Run: j5mzbdvb with config:
wandb:     batch_size: 16
wandb:     learning_rate: 0.09978527213100873
wandb: wandb version 0.19.9 is available!  To upgrade, please run:
wandb:  $ pip install wandb --upgrade
```

```
<IPython.core.display.HTML object>
```

```
<IPython.core.display.HTML object>
```

```
VBox(children=(Label(value=' 0.00MB of 0.00MB uploaded (0.00MB deduped)\r'),␣
 ↪FloatProgress(value=1.0, max=1.0)…
```

```
<IPython.core.display.HTML object>
```