Project

September 30, 2022

[1]:
```
import project_metrics.project_metrics as metrics
import numpy as np
import pandas as pd
import pickle
from os.path import exists
```

```
2022-09-30 00:18:09.283797: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
2022-09-30 00:18:09.635077: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open
shared object file: No such file or directory
2022-09-30 00:18:09.635100: I tensorflow/stream_executor/cuda/cudart_stub.cc:29]
Ignore above cudart dlerror if you do not have a GPU set up on your machine.
2022-09-30 00:18:09.679999: E tensorflow/stream_executor/cuda/cuda_blas.cc:2981]
Unable to register cuBLAS factory: Attempting to register factory for plugin
cuBLAS when one has already been registered
2022-09-30 00:18:10.636235: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libnvinfer.so.7'; dlerror: libnvinfer.so.7: cannot open shared
object file: No such file or directory
2022-09-30 00:18:10.636474: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libnvinfer_plugin.so.7'; dlerror: libnvinfer_plugin.so.7:
cannot open shared object file: No such file or directory
2022-09-30 00:18:10.636480: W
tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Cannot
dlopen some TensorRT libraries. If you would like to use Nvidia GPU with
TensorRT, please make sure the missing libraries mentioned above are installed
properly.
```

[2]:
```
df_train = pd.read_csv("Assets/train.csv")
df_val= pd.read_csv("Assets/val.csv")
df_test = pd.read_csv("Assets/test.csv")
```

[3]:
```
df_train.drop("INDEX", inplace=True, axis =1)
df_val.drop("INDEX", inplace=True, axis =1)
df_test.drop("INDEX", inplace=True, axis =1)

df_train.drop("Unnamed: 0", inplace=True, axis =1)
df_val.drop("Unnamed: 0", inplace=True, axis =1)
df_test.drop("Unnamed: 0", inplace=True, axis =1)

df_train.drop("IND_BOM_1_2", inplace=True, axis =1)
df_val.drop("IND_BOM_1_2", inplace=True, axis =1)
df_test.drop("IND_BOM_1_2", inplace=True, axis =1)
```

[4]:
```
y_train = df_train["IND_BOM_1_1"].values
y_val = df_val["IND_BOM_1_1"].values
y_test = df_test["IND_BOM_1_1"].values
```

[5]:
```
df_train.drop("IND_BOM_1_1", inplace=True, axis =1)
df_val.drop("IND_BOM_1_1", inplace=True, axis =1)
df_test.drop("IND_BOM_1_1", inplace=True, axis =1)
```

[6]:
```
x_train = df_train.values
x_val = df_val.values
x_test = df_test.values
```

[7]:
```
x_train.shape
```

[7]:
```
(255098, 243)
```

1 Final Project

1.1 MLP

[8]:
```
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping
import keras

MODELS_PATH = "Models/MLP"
HISTORY_PATH = f"{MODELS_PATH}/history"
MODEL_PATH = f"{MODELS_PATH}/model"
```

1.1.1 Experimento1

[9]:
```
# Número de features do nosso data set.
input_dim = x_train.shape[1]
```

```
# Aqui criamos o esboço da rede.
classifier = Sequential()

classifier.add(Dense(16, activation='relu', input_dim=input_dim))
classifier.add(Dense(1, activation='sigmoid'))
classifier.compile(optimizer='adam', loss='mean_squared_error')

history_file = HISTORY_PATH + "1.npy"
model_file = MODEL_PATH + "1"

if not exists(history_file) or  not exists(model_file):
    history = classifier.fit(x_train, y_train, batch_size=32, epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=10,verbose=1)], validation_data=(x_val,␣
 ↪y_val))
    np.save(history_file,history.history)
    classifier.save(model_file)
else:
    print("Model was already trained")

history=np.load(history_file, allow_pickle='TRUE').item()
classifier = keras.models.load_model(model_file)
```

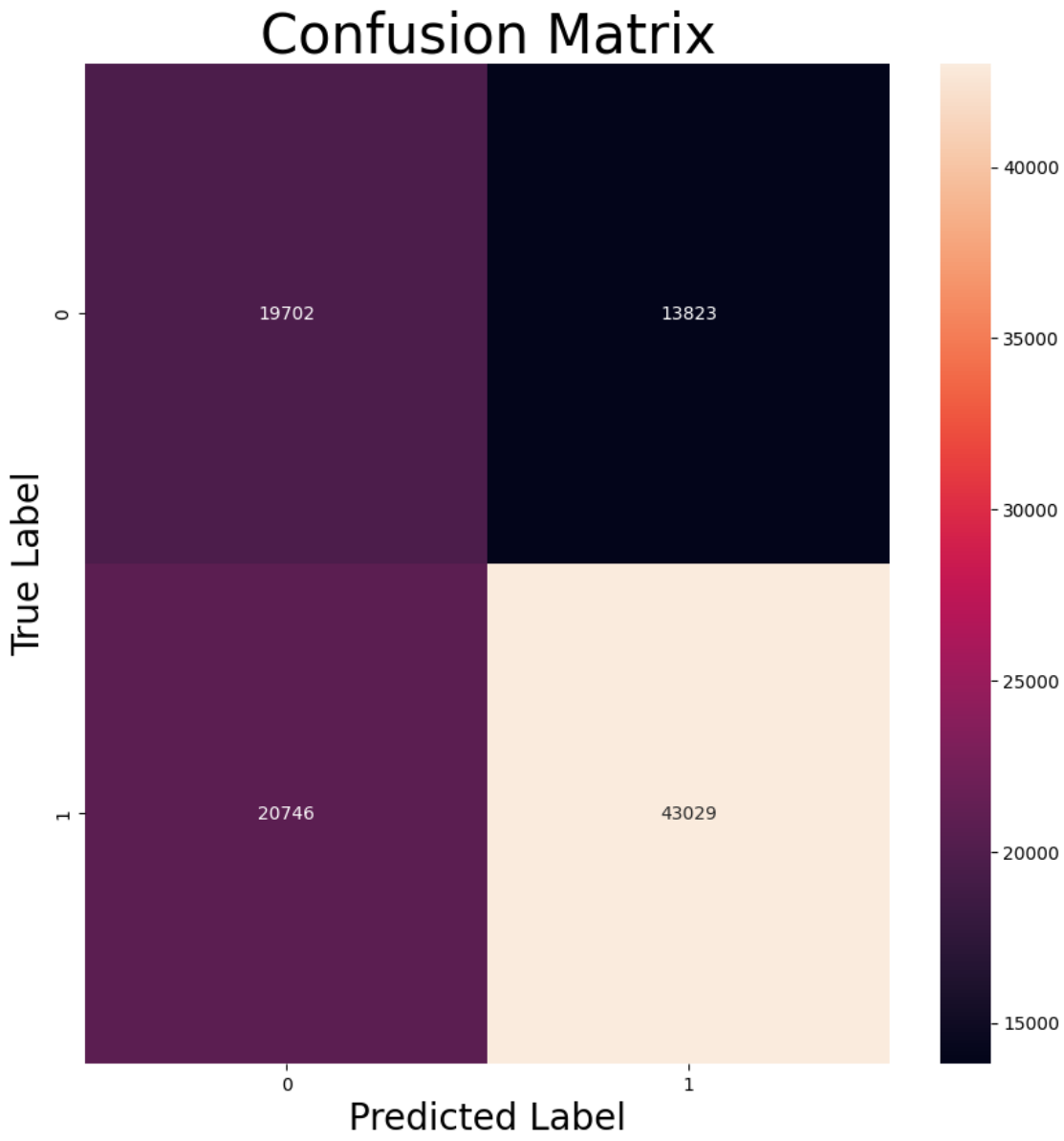
```
Model was already trained
```

```
2022-09-30 00:18:27.026734: I
tensorflow/stream_executor/cuda/cuda_gpu_executor.cc:980] successful NUMA node
read from SysFS had negative value (-1), but there must be at least one NUMA
node, so returning NUMA node zero
2022-09-30 00:18:27.026911: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudart.so.11.0'; dlerror: libcudart.so.11.0: cannot open
shared object file: No such file or directory
2022-09-30 00:18:27.027147: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcublas.so.11'; dlerror: libcublas.so.11: cannot open shared
object file: No such file or directory
2022-09-30 00:18:27.027196: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcublasLt.so.11'; dlerror: libcublasLt.so.11: cannot open
shared object file: No such file or directory
2022-09-30 00:18:27.027240: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcufft.so.10'; dlerror: libcufft.so.10: cannot open shared
object file: No such file or directory
2022-09-30 00:18:27.027287: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcurand.so.10'; dlerror: libcurand.so.10: cannot open shared
object file: No such file or directory
```

```
2022-09-30 00:18:27.027445: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcusolver.so.11'; dlerror: libcusolver.so.11: cannot open
shared object file: No such file or directory
2022-09-30 00:18:27.027492: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcusparse.so.11'; dlerror: libcusparse.so.11: cannot open
shared object file: No such file or directory
2022-09-30 00:18:27.027696: W
tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load
dynamic library 'libcudnn.so.8'; dlerror: libcudnn.so.8: cannot open shared
object file: No such file or directory
2022-09-30 00:18:27.027704: W
tensorflow/core/common_runtime/gpu/gpu_device.cc:1934] Cannot dlopen some GPU
libraries. Please make sure the missing libraries mentioned above are installed
properly if you would like to use GPU. Follow the guide at
https://www.tensorflow.org/install/gpu for how to download and setup the
required libraries for your platform.
Skipping registering GPU devices…
2022-09-30 00:18:27.028634: I tensorflow/core/platform/cpu_feature_guard.cc:193]
This TensorFlow binary is optimized with oneAPI Deep Neural Network Library
(oneDNN) to use the following CPU instructions in performance-critical
operations:  AVX2 FMA
To enable them in other operations, rebuild TensorFlow with the appropriate
compiler flags.
```

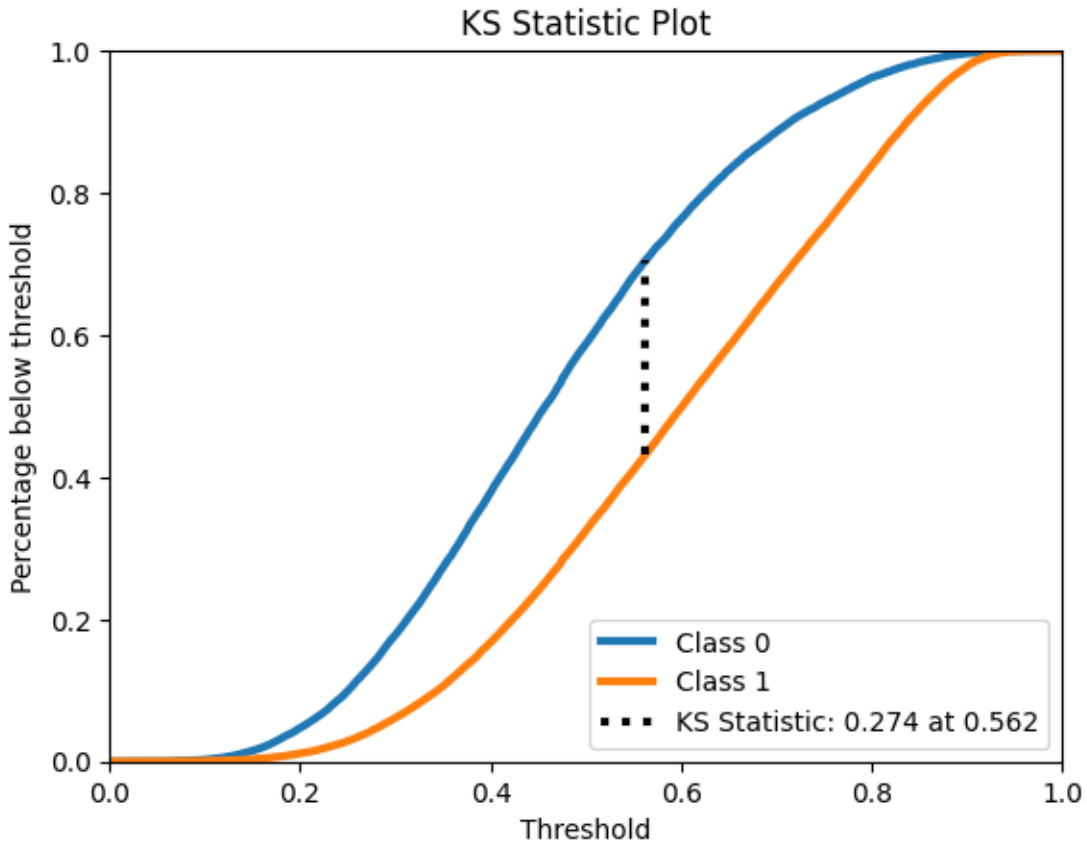
[10]:
```
# Fazer predições no conjunto de teste
y_pred_scores = classifier.predict(x_test)
y_pred_class = (y_pred_scores > 0.5).astype("int32")
y_pred_scores_0 = 1 - y_pred_scores
y_pred_scores = np.concatenate([y_pred_scores_0, y_pred_scores], axis=1)

## Matriz de confusão
print('Matriz de confusão no conjunto de teste:')
metrics.plot_confusion_matrix(y_test, y_pred_class)

## Resumo dos resultados
losses = metrics.extract_final_losses(history)
print()
print("{metric:<18}{value:.4f}".format(metric="Train Loss:",␣
 ↪value=losses['train_loss']))
print("{metric:<18}{value:.4f}".format(metric="Validation Loss:",␣
 ↪value=losses['val_loss']))
print('\nPerformance no conjunto de teste:')
accuracy, recall, precision, f1, auroc, aupr = metrics.
 ↪compute_performance_metrics(y_test, y_pred_class, y_pred_scores)
metrics.print_metrics_summary(accuracy, recall, precision, f1, auroc, aupr)
```

```
   1/3041 […] - ETA: 6:06

2022-09-30 00:18:27.450258: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 94575600
exceeds 10% of free system memory.

3041/3041 [==============================] - 2s 739us/step
Matriz de confusão no conjunto de teste:
```




```
Train Loss:       0.2187
Validation Loss:  0.2214
```

Performance no conjunto de teste:



Accuracy:	0.6447
Recall:	0.6747
Precision:	0.7569
F1:	0.7134
AUROC:	0.6887
AUPR:	0.8014

1.1.2 Experimento2

[11]:
```
# Número de features do nosso data set.
input_dim = x_train.shape[1]

# Aqui criamos o esboço da rede.
classifier = Sequential()

classifier.add(Dense(256, activation='relu', input_dim=input_dim))
```

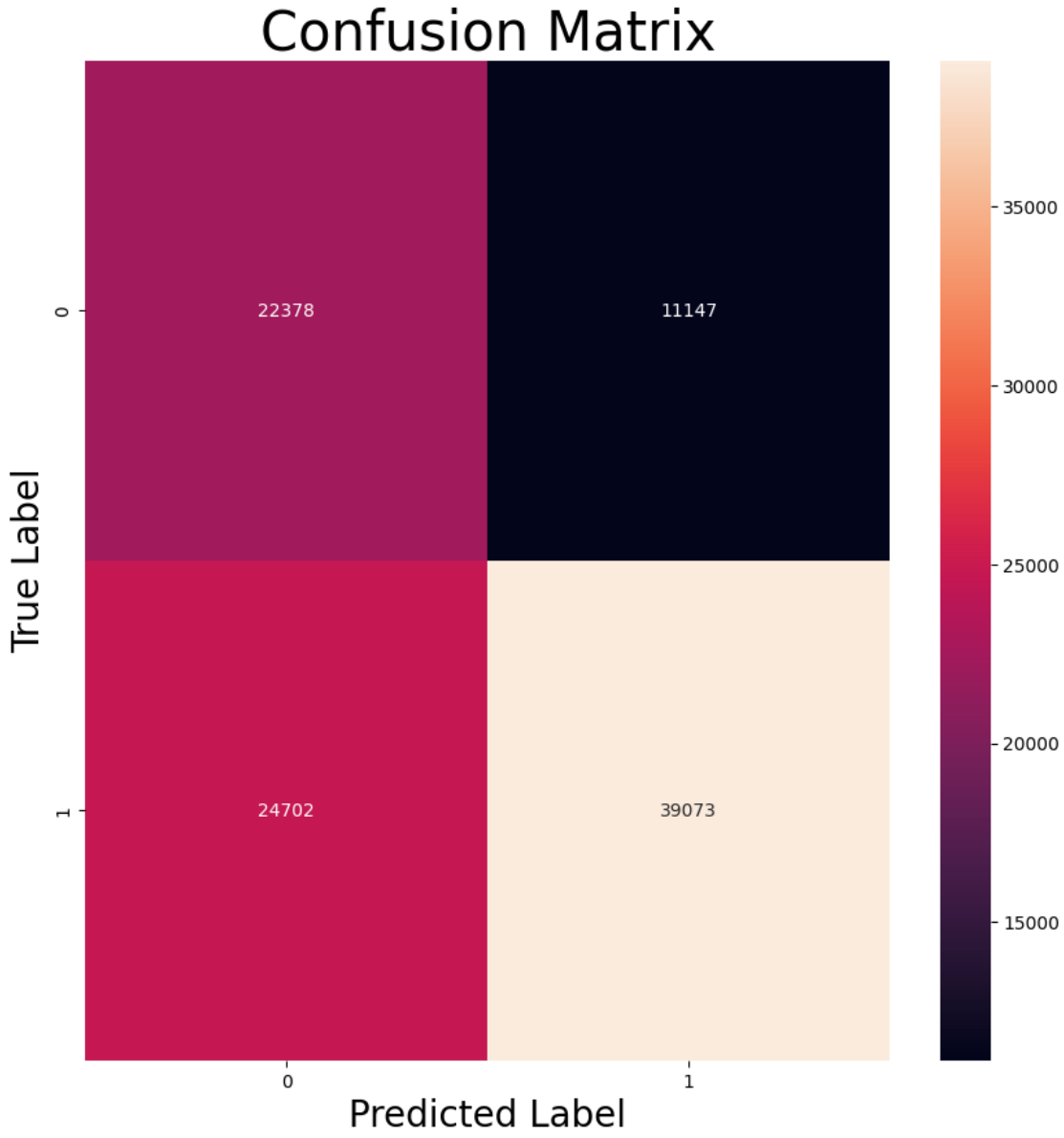
```
classifier.add(Dense(128, activation='relu', input_dim=input_dim))
classifier.add(Dense(1, activation='sigmoid'))
classifier.compile(optimizer='adam', loss='mean_squared_error')

history_file = HISTORY_PATH + "2.npy"
model_file = MODEL_PATH + "2"

if not exists(history_file) or  not exists(model_file):
    history = classifier.fit(x_train, y_train, batch_size=32, epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=20,verbose=1,restore_best_weights=True)],␣
 ↪validation_data=(x_val, y_val))
    np.save(history_file,history.history)
    classifier.save(model_file)
else:
    print("Model was already trained")

history=np.load(history_file, allow_pickle='TRUE').item()
classifier = keras.models.load_model(model_file)
```

```
Model was already trained
```

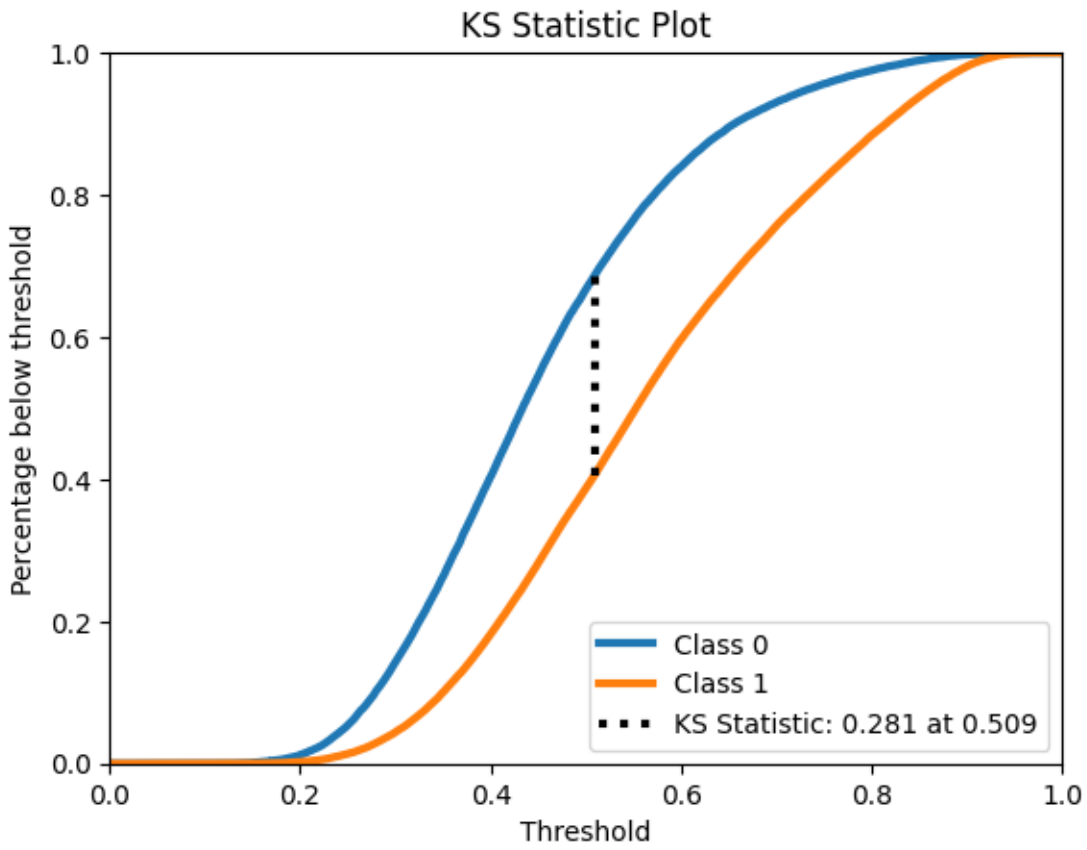
[12]:
```
# Fazer predições no conjunto de teste
y_pred_scores = classifier.predict(x_test)
y_pred_class = (y_pred_scores > 0.5).astype("int32")
y_pred_scores_0 = 1 - y_pred_scores
y_pred_scores = np.concatenate([y_pred_scores_0, y_pred_scores], axis=1)

## Matriz de confusão
print('Matriz de confusão no conjunto de teste:')
metrics.plot_confusion_matrix(y_test, y_pred_class)

## Resumo dos resultados
losses = metrics.extract_final_losses(history)
print()
print("{metric:<18}{value:.4f}".format(metric="Train Loss:",␣
 ↪value=losses['train_loss']))
print("{metric:<18}{value:.4f}".format(metric="Validation Loss:",␣
 ↪value=losses['val_loss']))
print('\nPerformance no conjunto de teste:')
accuracy, recall, precision, f1, auroc, aupr = metrics.
 ↪compute_performance_metrics(y_test, y_pred_class, y_pred_scores)
metrics.print_metrics_summary(accuracy, recall, precision, f1, auroc, aupr)
```

```
 110/3041 [>…] - ETA: 2s

2022-09-30 00:18:31.959175: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 94575600
exceeds 10% of free system memory.
```

```
3041/3041 [==============================] - 3s 926us/step
Matriz de confusão no conjunto de teste:
```




```
Train Loss:       0.2206
Validation Loss:  0.2213

Performance no conjunto de teste:
```




```
Accuracy:           0.6316
Recall:             0.6127
Precision:          0.7780
F1:                 0.6855
AUROC:              0.6921
AUPR:               0.8042
```

1.2 Experimento3

[13]:
```
# Número de features do nosso data set.
input_dim = x_train.shape[1]

# Aqui criamos o esboço da rede.
classifier = Sequential()

classifier.add(Dense(512, activation='sigmoid', input_dim=input_dim))
classifier.add(Dense(256, activation='sigmoid', input_dim=input_dim))
classifier.add(Dense(1, activation='sigmoid'))
```

```
optimizer=keras.optimizers.SGD( learning_rate=0.01)
classifier.compile(optimizer=optimizer, loss='mean_squared_error')

history_file = HISTORY_PATH + "3.npy"
model_file = MODEL_PATH + "3"

if not exists(history_file) or  not exists(model_file):
    history = classifier.fit(x_train, y_train, batch_size=32, epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=20,verbose=1,restore_best_weights=True)],␣
 ↪validation_data=(x_val, y_val))
    np.save(history_file,history.history)
    classifier.save(model_file)
else:
    print("Model was already trained")

history=np.load(history_file, allow_pickle='TRUE').item()
classifier = keras.models.load_model(model_file)
```

```
Epoch 1/10000
```

```
2022-09-30 00:18:36.822806: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 247955256
exceeds 10% of free system memory.
```

```
7949/7972 [============================>.] - ETA: 0s - loss: 0.2488
```

```
2022-09-30 00:18:53.759515: W
tensorflow/core/framework/cpu_allocator_impl.cc:82] Allocation of 123976656
exceeds 10% of free system memory.
```

```
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2488 -
val_loss: 0.2468
Epoch 2/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2448 -
val_loss: 0.2441
Epoch 3/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2403 -
val_loss: 0.2395
Epoch 4/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2360 -
val_loss: 0.2337
Epoch 5/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2332 -
val_loss: 0.2321
Epoch 6/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2314 -
val_loss: 0.2312
Epoch 7/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2304 -
```

```
val_loss: 0.2293
Epoch 8/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2296 -
val_loss: 0.2287
Epoch 9/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2289 -
val_loss: 0.2288
Epoch 10/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2285 -
val_loss: 0.2276
Epoch 11/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2281 -
val_loss: 0.2300
Epoch 12/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2278 -
val_loss: 0.2276
Epoch 13/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2276 -
val_loss: 0.2269
Epoch 14/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2274 -
val_loss: 0.2269
Epoch 15/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2272 -
val_loss: 0.2301
Epoch 16/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2270 -
val_loss: 0.2267
Epoch 17/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2270 -
val_loss: 0.2265
Epoch 18/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2269 -
val_loss: 0.2266
Epoch 19/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2268 -
val_loss: 0.2263
Epoch 20/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2267 -
val_loss: 0.2282
Epoch 21/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2266 -
val_loss: 0.2269
Epoch 22/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2266 -
val_loss: 0.2261
Epoch 23/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2265 -
```

```
val_loss: 0.2262
Epoch 24/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2264 -
val_loss: 0.2262
Epoch 25/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2264 -
val_loss: 0.2269
Epoch 26/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2264 -
val_loss: 0.2263
Epoch 27/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2263 -
val_loss: 0.2263
Epoch 28/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2263 -
val_loss: 0.2259
Epoch 29/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2262 -
val_loss: 0.2258
Epoch 30/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2262 -
val_loss: 0.2263
Epoch 31/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2262 -
val_loss: 0.2262
Epoch 32/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2261 -
val_loss: 0.2265
Epoch 33/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2261 -
val_loss: 0.2257
Epoch 34/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2261 -
val_loss: 0.2278
Epoch 35/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2260 -
val_loss: 0.2257
Epoch 36/10000
7972/7972 [==============================] - 23s 3ms/step - loss: 0.2260 -
val_loss: 0.2267
Epoch 37/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2260 -
val_loss: 0.2263
Epoch 38/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2260 -
val_loss: 0.2256
Epoch 39/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2259 -
```

```
val_loss: 0.2256
Epoch 40/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2259 -
val_loss: 0.2259
Epoch 41/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2259 -
val_loss: 0.2263
Epoch 42/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2259 -
val_loss: 0.2256
Epoch 43/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2259 -
val_loss: 0.2256
Epoch 44/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2259 -
val_loss: 0.2277
Epoch 45/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2259 -
val_loss: 0.2273
Epoch 46/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2259 -
val_loss: 0.2270
Epoch 47/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2259 -
val_loss: 0.2261
Epoch 48/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2258 -
val_loss: 0.2255
Epoch 49/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
val_loss: 0.2255
Epoch 50/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2258 -
val_loss: 0.2264
Epoch 51/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2258 -
val_loss: 0.2257
Epoch 52/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2258 -
val_loss: 0.2254
Epoch 53/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
val_loss: 0.2258
Epoch 54/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2257 -
val_loss: 0.2255
Epoch 55/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
```

```
val_loss: 0.2259
Epoch 56/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
val_loss: 0.2255
Epoch 57/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
val_loss: 0.2258
Epoch 58/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2257 -
val_loss: 0.2254
Epoch 59/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2256 -
val_loss: 0.2254
Epoch 60/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2257 -
val_loss: 0.2255
Epoch 61/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2256 -
val_loss: 0.2256
Epoch 62/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2256 -
val_loss: 0.2257
Epoch 63/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2256 -
val_loss: 0.2257
Epoch 64/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2256 -
val_loss: 0.2260
Epoch 65/10000
7972/7972 [==============================] - 23s 3ms/step - loss: 0.2256 -
val_loss: 0.2270
Epoch 66/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2256 -
val_loss: 0.2259
Epoch 67/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2256 -
val_loss: 0.2258
Epoch 68/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2256 -
val_loss: 0.2263
Epoch 69/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
val_loss: 0.2315
Epoch 70/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2256 -
val_loss: 0.2259
Epoch 71/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
```

```
val_loss: 0.2256
Epoch 72/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2255 -
val_loss: 0.2259
Epoch 73/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
val_loss: 0.2256
Epoch 74/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
val_loss: 0.2253
Epoch 75/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
val_loss: 0.2253
Epoch 76/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2255 -
val_loss: 0.2277
Epoch 77/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2254 -
val_loss: 0.2257
Epoch 78/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2255 -
val_loss: 0.2269
Epoch 79/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2255 -
val_loss: 0.2256
Epoch 80/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2254 -
val_loss: 0.2252
Epoch 81/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2254 -
val_loss: 0.2258
Epoch 82/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2254 -
val_loss: 0.2252
Epoch 83/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2254 -
val_loss: 0.2257
Epoch 84/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2254 -
val_loss: 0.2253
Epoch 85/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2254 -
val_loss: 0.2285
Epoch 86/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2254 -
val_loss: 0.2254
Epoch 87/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2253 -
```

```
val_loss: 0.2252
Epoch 88/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2254 -
val_loss: 0.2252
Epoch 89/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2253 -
val_loss: 0.2253
Epoch 90/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2253 -
val_loss: 0.2254
Epoch 91/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2253 -
val_loss: 0.2265
Epoch 92/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2253 -
val_loss: 0.2251
Epoch 93/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2253 -
val_loss: 0.2264
Epoch 94/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2253 -
val_loss: 0.2252
Epoch 95/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2253 -
val_loss: 0.2251
Epoch 96/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2252 -
val_loss: 0.2260
Epoch 97/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2252 -
val_loss: 0.2290
Epoch 98/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2252 -
val_loss: 0.2266
Epoch 99/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2252 -
val_loss: 0.2253
Epoch 100/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2251 -
val_loss: 0.2266
Epoch 101/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2251 -
val_loss: 0.2251
Epoch 102/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2251 -
val_loss: 0.2253
Epoch 103/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2251 -
```

```
val_loss: 0.2252
Epoch 104/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2251 -
val_loss: 0.2250
Epoch 105/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2251 -
val_loss: 0.2249
Epoch 106/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2250 -
val_loss: 0.2250
Epoch 107/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2250 -
val_loss: 0.2254
Epoch 108/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2250 -
val_loss: 0.2250
Epoch 109/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2250 -
val_loss: 0.2248
Epoch 110/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2249 -
val_loss: 0.2249
Epoch 111/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2249 -
val_loss: 0.2249
Epoch 112/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2248 -
val_loss: 0.2260
Epoch 113/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2249 -
val_loss: 0.2250
Epoch 114/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2248 -
val_loss: 0.2251
Epoch 115/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2248 -
val_loss: 0.2251
Epoch 116/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2248 -
val_loss: 0.2247
Epoch 117/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2247 -
val_loss: 0.2251
Epoch 118/10000
7972/7972 [==============================] - 23s 3ms/step - loss: 0.2247 -
val_loss: 0.2259
Epoch 119/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2247 -
```

```
val_loss: 0.2245
Epoch 120/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2246 -
val_loss: 0.2246
Epoch 121/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2246 -
val_loss: 0.2248
Epoch 122/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2245 -
val_loss: 0.2244
Epoch 123/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2245 -
val_loss: 0.2257
Epoch 124/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2245 -
val_loss: 0.2250
Epoch 125/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2244 -
val_loss: 0.2243
Epoch 126/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2244 -
val_loss: 0.2252
Epoch 127/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2243 -
val_loss: 0.2243
Epoch 128/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2243 -
val_loss: 0.2242
Epoch 129/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2242 -
val_loss: 0.2243
Epoch 130/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2241 -
val_loss: 0.2244
Epoch 131/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2241 -
val_loss: 0.2248
Epoch 132/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2241 -
val_loss: 0.2241
Epoch 133/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2240 -
val_loss: 0.2240
Epoch 134/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2240 -
val_loss: 0.2239
Epoch 135/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2239 -
```

```
val_loss: 0.2238
Epoch 136/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2239 -
val_loss: 0.2246
Epoch 137/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2238 -
val_loss: 0.2242
Epoch 138/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2238 -
val_loss: 0.2238
Epoch 139/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2237 -
val_loss: 0.2237
Epoch 140/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2237 -
val_loss: 0.2239
Epoch 141/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2236 -
val_loss: 0.2238
Epoch 142/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2235 -
val_loss: 0.2239
Epoch 143/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2235 -
val_loss: 0.2235
Epoch 144/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2234 -
val_loss: 0.2236
Epoch 145/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2234 -
val_loss: 0.2234
Epoch 146/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2233 -
val_loss: 0.2233
Epoch 147/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2233 -
val_loss: 0.2233
Epoch 148/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2232 -
val_loss: 0.2232
Epoch 149/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2232 -
val_loss: 0.2237
Epoch 150/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2231 -
val_loss: 0.2233
Epoch 151/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2231 -
```

```
val_loss: 0.2245
Epoch 152/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2230 -
val_loss: 0.2232
Epoch 153/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2230 -
val_loss: 0.2234
Epoch 154/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2229 -
val_loss: 0.2229
Epoch 155/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2228 -
val_loss: 0.2236
Epoch 156/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2228 -
val_loss: 0.2256
Epoch 157/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2228 -
val_loss: 0.2228
Epoch 158/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2227 -
val_loss: 0.2228
Epoch 159/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2227 -
val_loss: 0.2227
Epoch 160/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2226 -
val_loss: 0.2230
Epoch 161/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2226 -
val_loss: 0.2263
Epoch 162/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2226 -
val_loss: 0.2228
Epoch 163/10000
7972/7972 [==============================] - 23s 3ms/step - loss: 0.2225 -
val_loss: 0.2232
Epoch 164/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2225 -
val_loss: 0.2226
Epoch 165/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2224 -
val_loss: 0.2234
Epoch 166/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2224 -
val_loss: 0.2226
Epoch 167/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2223 -
```

```
val_loss: 0.2227
Epoch 168/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2223 -
val_loss: 0.2226
Epoch 169/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2223 -
val_loss: 0.2227
Epoch 170/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2223 -
val_loss: 0.2231
Epoch 171/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2222 -
val_loss: 0.2224
Epoch 172/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2222 -
val_loss: 0.2223
Epoch 173/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2221 -
val_loss: 0.2224
Epoch 174/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2221 -
val_loss: 0.2225
Epoch 175/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2221 -
val_loss: 0.2228
Epoch 176/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2220 -
val_loss: 0.2223
Epoch 177/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2220 -
val_loss: 0.2248
Epoch 178/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2219 -
val_loss: 0.2227
Epoch 179/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2219 -
val_loss: 0.2225
Epoch 180/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2219 -
val_loss: 0.2222
Epoch 181/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2219 -
val_loss: 0.2221
Epoch 182/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2218 -
val_loss: 0.2221
Epoch 183/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2218 -
```

```
val_loss: 0.2223
Epoch 184/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2217 -
val_loss: 0.2220
Epoch 185/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2217 -
val_loss: 0.2223
Epoch 186/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2217 -
val_loss: 0.2224
Epoch 187/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2217 -
val_loss: 0.2226
Epoch 188/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2216 -
val_loss: 0.2219
Epoch 189/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2216 -
val_loss: 0.2219
Epoch 190/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2216 -
val_loss: 0.2223
Epoch 191/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2216 -
val_loss: 0.2220
Epoch 192/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2215 -
val_loss: 0.2222
Epoch 193/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2215 -
val_loss: 0.2219
Epoch 194/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2215 -
val_loss: 0.2223
Epoch 195/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2214 -
val_loss: 0.2218
Epoch 196/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2214 -
val_loss: 0.2224
Epoch 197/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2214 -
val_loss: 0.2237
Epoch 198/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2214 -
val_loss: 0.2224
Epoch 199/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2213 -
```

```
val_loss: 0.2219
Epoch 200/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2213 -
val_loss: 0.2219
Epoch 201/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2212 -
val_loss: 0.2222
Epoch 202/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2212 -
val_loss: 0.2217
Epoch 203/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2213 -
val_loss: 0.2218
Epoch 204/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2212 -
val_loss: 0.2218
Epoch 205/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2211 -
val_loss: 0.2217
Epoch 206/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2211 -
val_loss: 0.2216
Epoch 207/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2211 -
val_loss: 0.2217
Epoch 208/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2211 -
val_loss: 0.2216
Epoch 209/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2211 -
val_loss: 0.2216
Epoch 210/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2211 -
val_loss: 0.2217
Epoch 211/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2210 -
val_loss: 0.2221
Epoch 212/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2210 -
val_loss: 0.2218
Epoch 213/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2210 -
val_loss: 0.2251
Epoch 214/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2209 -
val_loss: 0.2215
Epoch 215/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2209 -
```

```
val_loss: 0.2232
Epoch 216/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2209 -
val_loss: 0.2215
Epoch 217/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2209 -
val_loss: 0.2223
Epoch 218/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2209 -
val_loss: 0.2215
Epoch 219/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2208 -
val_loss: 0.2215
Epoch 220/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2208 -
val_loss: 0.2214
Epoch 221/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2208 -
val_loss: 0.2225
Epoch 222/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2208 -
val_loss: 0.2221
Epoch 223/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2208 -
val_loss: 0.2219
Epoch 224/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2207 -
val_loss: 0.2214
Epoch 225/10000
7972/7972 [==============================] - 24s 3ms/step - loss: 0.2207 -
val_loss: 0.2215
Epoch 226/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2206 -
val_loss: 0.2223
Epoch 227/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2206 -
val_loss: 0.2215
Epoch 228/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2206 -
val_loss: 0.2220
Epoch 229/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2205 -
val_loss: 0.2217
Epoch 230/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2206 -
val_loss: 0.2220
Epoch 231/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2205 -
```

```
val_loss: 0.2216
Epoch 232/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2205 -
val_loss: 0.2214
Epoch 233/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2205 -
val_loss: 0.2212
Epoch 234/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2205 -
val_loss: 0.2213
Epoch 235/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2205 -
val_loss: 0.2213
Epoch 236/10000
7972/7972 [==============================] - 22s 3ms/step - loss: 0.2204 -
val_loss: 0.2224
Epoch 237/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2204 -
val_loss: 0.2219
Epoch 238/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2204 -
val_loss: 0.2220
Epoch 239/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2203 -
val_loss: 0.2212
Epoch 240/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2203 -
val_loss: 0.2212
Epoch 241/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2203 -
val_loss: 0.2218
Epoch 242/10000
7972/7972 [==============================] - 15983s 2s/step - loss: 0.2203 -
val_loss: 0.2224
Epoch 243/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2203 -
val_loss: 0.2225
Epoch 244/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2202 -
val_loss: 0.2216
Epoch 245/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2202 -
val_loss: 0.2224
Epoch 246/10000
7972/7972 [==============================] - 20s 3ms/step - loss: 0.2202 -
val_loss: 0.2211
Epoch 247/10000
7972/7972 [==============================] - 20s 2ms/step - loss: 0.2202 -
```

```
val_loss: 0.2211
Epoch 248/10000
7972/7972 [==============================] - 21s 3ms/step - loss: 0.2201 -
val_loss: 0.2212
Epoch 249/10000
7972/7972 [==============================] - 19s 2ms/step - loss: 0.2201 -
val_loss: 0.2212
Epoch 250/10000
7972/7972 [==============================] - 18s 2ms/step - loss: 0.2201 -
val_loss: 0.2211
Epoch 251/10000
7972/7972 [==============================] - 18s 2ms/step - loss: 0.2201 -
val_loss: 0.2221
Epoch 252/10000
7972/7972 [==============================] - 17s 2ms/step - loss: 0.2200 -
val_loss: 0.2215
Epoch 253/10000
7972/7972 [==============================] - 15s 2ms/step - loss: 0.2200 -
val_loss: 0.2214
Epoch 254/10000
7972/7972 [==============================] - 15s 2ms/step - loss: 0.2200 -
val_loss: 0.2211
Epoch 255/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2200 -
val_loss: 0.2214
Epoch 256/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2200 -
val_loss: 0.2210
Epoch 257/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2199 -
val_loss: 0.2211
Epoch 258/10000
7972/7972 [==============================] - 17s 2ms/step - loss: 0.2199 -
val_loss: 0.2216
Epoch 259/10000
7972/7972 [==============================] - 17s 2ms/step - loss: 0.2199 -
val_loss: 0.2210
Epoch 260/10000
7972/7972 [==============================] - 15s 2ms/step - loss: 0.2199 -
val_loss: 0.2226
Epoch 261/10000
7972/7972 [==============================] - 15s 2ms/step - loss: 0.2199 -
val_loss: 0.2211
Epoch 262/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2198 -
val_loss: 0.2211
Epoch 263/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2198 -
```

```
val_loss: 0.2212
Epoch 264/10000
7972/7972 [==============================] - 16s 2ms/step - loss: 0.2198 -
val_loss: 0.2210
Epoch 265/10000
7972/7972 [==============================] - 17s 2ms/step - loss: 0.2198 -
val_loss: 0.2210
Epoch 266/10000
7972/7972 [==============================] - 19s 2ms/step - loss: 0.2197 -
val_loss: 0.2212
Epoch 267/10000
7833/7972 [============================>.] - ETA: 0s - loss: 0.2198
```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
Cell In [13], line 18
     15 model_file = MODEL_PATH + "3"
     17 if not exists(history_file) or  not exists(model_file):
---> 18     history = classifier.fit(x_train, y_train, batch_size=32,␣
 ↪epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=20,verbose=1,restore_best_weights=True)],␣
 ↪validation_data=(x_val, y_val))
     19     np.save(history_file,history.history)
     20     classifier.save(model_file)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪keras/utils/traceback_utils.py:65, in filter_traceback.<locals>.
 ↪error_handler(*args, **kwargs)
     63 filtered_tb = None
     64 try:
---> 65     return fn(*args, **kwargs)
     66 except Exception as e:
     67     filtered_tb = _process_traceback_frames(e.__traceback__)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪keras/engine/training.py:1564, in Model.fit(self, x, y, batch_size, epochs,␣
 ↪verbose, callbacks, validation_split, validation_data, shuffle, class_weight,␣
 ↪sample_weight, initial_epoch, steps_per_epoch, validation_steps,␣
 ↪validation_batch_size, validation_freq, max_queue_size, workers,␣
 ↪use_multiprocessing)
   1556 with tf.profiler.experimental.Trace(
   1557     "train",
   1558     epoch_num=epoch,
   (…)
   1561     _r=1,
   1562 ):
   1563     callbacks.on_train_batch_begin(step)
-> 1564     tmp_logs = self.train_function(iterator)
   1565     if data_handler.should_sync:
```

```
   1566         context.async_wait()

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/util/traceback_utils.py:150, in filter_traceback.<locals>.
 ↪error_handler(*args, **kwargs)
    148 filtered_tb = None
    149 try:
--> 150   return fn(*args, **kwargs)
    151 except Exception as e:
    152   filtered_tb = _process_traceback_frames(e.__traceback__)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/def_function.py:915, in Function.__call__(self, *args␣
 ↪**kwds)
    912 compiler = "xla" if self._jit_compile else "nonXla"
    914 with OptionalXlaContext(self._jit_compile):
--> 915   result = self._call(*args, **kwds)
    917 new_tracing_count = self.experimental_get_tracing_count()
    918 without_tracing = (tracing_count == new_tracing_count)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/def_function.py:947, in Function._call(self, *args,␣
 ↪**kwds)
    944   self._lock.release()
    945   # In this case we have created variables on the first call, so we run␣
 ↪the
    946   # defunned version which is guaranteed to never create variables.
--> 947   return self._stateless_fn(*args, **kwds)  # pylint:␣
 ↪disable=not-callable
    948 elif self._stateful_fn is not None:
    949   # Release the lock early so that multiple threads can perform the cal
    950   # in parallel.
    951   self._lock.release()

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/function.py:2496, in Function.__call__(self, *args,␣
 ↪**kwargs)
   2493 with self._lock:
   2494   (graph_function,
   2495    filtered_flat_args) = self._maybe_define_function(args, kwargs)
-> 2496 return graph_function._call_flat(
   2497     filtered_flat_args, captured_inputs=graph_function.captured_inputs)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/function.py:1862, in ConcreteFunction._call_flat(self␣
 ↪args, captured_inputs, cancellation_manager)
   1858 possible_gradient_type = gradients_util.PossibleTapeGradientTypes(args)
   1859 if (possible_gradient_type == gradients_util.POSSIBLE_GRADIENT_TYPES_NO E
   1860     and executing_eagerly):
```

```
   1861   # No tape is watching; skip to running the function.
-> 1862   return self._build_call_outputs(self._inference_function.call(
   1863       ctx, args, cancellation_manager=cancellation_manager))
   1864 forward_backward = self._select_forward_and_backward_functions(
   1865     args,
   1866     possible_gradient_type,
   1867     executing_eagerly)
   1868 forward_function, args_with_tangents = forward_backward.forward()

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/function.py:499, in _EagerDefinedFunction.call(self,
 ↪ctx, args, cancellation_manager)
    497 with _InterpolateFunctionError(self):
    498   if cancellation_manager is None:
--> 499     outputs = execute.execute(
    500         str(self.signature.name),
    501         num_outputs=self._num_outputs,
    502         inputs=args,
    503         attrs=attrs,
    504         ctx=ctx)
    505   else:
    506     outputs = execute.execute_with_cancellation(
    507         str(self.signature.name),
    508         num_outputs=self._num_outputs,
   (…)
    511         ctx=ctx,
    512         cancellation_manager=cancellation_manager)

File ~/Desktop/neural-networks/neural-networks/lib/python3.10/site-packages/
 ↪tensorflow/python/eager/execute.py:54, in quick_execute(op_name, num_outputs,
 ↪inputs, attrs, ctx, name)
     52 try:
     53   ctx.ensure_initialized()
---> 54   tensors = pywrap_tfe.TFE_Py_Execute(ctx._handle, device_name, op_name
     55                                       inputs, attrs, num_outputs)
     56 except core._NotOkStatusException as e:
     57   if name is not None:

KeyboardInterrupt:
```

[ ]:
```
# Fazer predições no conjunto de teste
y_pred_scores = classifier.predict(x_test)
y_pred_class = (y_pred_scores > 0.5).astype("int32")
y_pred_scores_0 = 1 - y_pred_scores
y_pred_scores = np.concatenate([y_pred_scores_0, y_pred_scores], axis=1)

## Matriz de confusão
```

```
print('Matriz de confusão no conjunto de teste:')
metrics.plot_confusion_matrix(y_test, y_pred_class)

## Resumo dos resultados
losses = metrics.extract_final_losses(history)
print()
print("{metric:<18}{value:.4f}".format(metric="Train Loss:",␣
 ↪value=losses['train_loss']))
print("{metric:<18}{value:.4f}".format(metric="Validation Loss:",␣
 ↪value=losses['val_loss']))
print('\nPerformance no conjunto de teste:')
accuracy, recall, precision, f1, auroc, aupr = metrics.
 ↪compute_performance_metrics(y_test, y_pred_class, y_pred_scores)
metrics.print_metrics_summary(accuracy, recall, precision, f1, auroc, aupr)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In [1], line 2
      1 # Fazer predições no conjunto de teste
----> 2 y_pred_scores = classifier.predict(x_test)
      3 y_pred_class = (y_pred_scores > 0.5).astype("int32")
      4 y_pred_scores_0 = 1 - y_pred_scores

NameError: name 'classifier' is not defined
```

2 Experimento4

[ ]:
```
# Número de features do nosso data set.
input_dim = x_train.shape[1]

# Aqui criamos o esboço da rede.
classifier = Sequential()

classifier.add(Dense(512, activation='tanh', input_dim=input_dim))
classifier.add(Dense(256, activation='tanh', input_dim=input_dim))
classifier.add(Dense(1, activation='sigmoid'))

optimizer=keras.optimizers.SGD( learning_rate=0.01)
classifier.compile(optimizer=optimizer, loss='mean_squared_error')

history_file = HISTORY_PATH + "4.npy"
model_file = MODEL_PATH + "4"

if not exists(history_file) or  not exists(model_file):
```

```
    history = classifier.fit(x_train, y_train, batch_size=32, epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=20,verbose=1,restore_best_weights=True)],␣
 ↪validation_data=(x_val, y_val))
    np.save(history_file,history.history)
    classifier.save(model_file)
else:
    print("Model was already trained")

history=np.load(history_file, allow_pickle='TRUE').item()
classifier = keras.models.load_model(model_file)
```

[ ]:
```
# Fazer predições no conjunto de teste
y_pred_scores = classifier.predict(x_test)
y_pred_class = (y_pred_scores > 0.5).astype("int32")
y_pred_scores_0 = 1 - y_pred_scores
y_pred_scores = np.concatenate([y_pred_scores_0, y_pred_scores], axis=1)

## Matriz de confusão
print('Matriz de confusão no conjunto de teste:')
metrics.plot_confusion_matrix(y_test, y_pred_class)

## Resumo dos resultados
losses = metrics.extract_final_losses(history)
print()
print("{metric:<18}{value:.4f}".format(metric="Train Loss:",␣
 ↪value=losses['train_loss']))
print("{metric:<18}{value:.4f}".format(metric="Validation Loss:",␣
 ↪value=losses['val_loss']))
print('\nPerformance no conjunto de teste:')
accuracy, recall, precision, f1, auroc, aupr = metrics.
 ↪compute_performance_metrics(y_test, y_pred_class, y_pred_scores)
metrics.print_metrics_summary(accuracy, recall, precision, f1, auroc, aupr)
```

3 Experimento5

[ ]:
```
# Número de features do nosso data set.
input_dim = x_train.shape[1]

# Aqui criamos o esboço da rede.
classifier = Sequential()

classifier.add(Dense(1024, activation='tanh', input_dim=input_dim))
classifier.add(Dense(1, activation='sigmoid'))
classifier.compile(optimizer='adam', loss='mean_squared_error')

history_file = HISTORY_PATH + "5.npy"
```

```
model_file = MODEL_PATH + "5"

if not exists(history_file) or  not exists(model_file):
    history = classifier.fit(x_train, y_train, batch_size=32, epochs=10_000,␣
 ↪callbacks=[EarlyStopping(patience=20,verbose=1)], validation_data=(x_val,␣
 ↪y_val))
    np.save(history_file,history.history)
    classifier.save(model_file)
else:
    print("Model was already trained")

history=np.load(history_file, allow_pickle='TRUE').item()
classifier = keras.models.load_model(model_file)
```

[ ]:
```
# Fazer predições no conjunto de teste
y_pred_scores = classifier.predict(x_test)
y_pred_class = (y_pred_scores > 0.5).astype("int32")
y_pred_scores_0 = 1 - y_pred_scores
y_pred_scores = np.concatenate([y_pred_scores_0, y_pred_scores], axis=1)

## Matriz de confusão
print('Matriz de confusão no conjunto de teste:')
metrics.plot_confusion_matrix(y_test, y_pred_class)

## Resumo dos resultados
losses = metrics.extract_final_losses(history)
print()
print("{metric:<18}{value:.4f}".format(metric="Train Loss:",␣
 ↪value=losses['train_loss']))
print("{metric:<18}{value:.4f}".format(metric="Validation Loss:",␣
 ↪value=losses['val_loss']))
print('\nPerformance no conjunto de teste:')
accuracy, recall, precision, f1, auroc, aupr = metrics.
 ↪compute_performance_metrics(y_test, y_pred_class, y_pred_scores)
metrics.print_metrics_summary(accuracy, recall, precision, f1, auroc, aupr)
```