Aula 4

Pré-processamento de dados

Classificação binária com dados não balanceados

Eduardo Lobo Lustosa Cabral

1. Objetivos

- Apresentar exemplo de pré-processamento de dados.

- Apresentar como classificar um conjunto de dados altamente desbalanceado no qual o número de exemplos de uma classe supera em muito os exemplos da outra.

  O conjunto de dados usados é o Detecção de fraude de cartão de crédito do Kaggle. O objetivo desses dados é detectar apenas 492 transações fraudulentas de um total de 284.807 transações.

- As tarefas realizadas para desenvolver a RNA para essa tarefa de classificação são as seguintes:

  - Carregar um arquivo tipo CSV usando o Pandas;
  - Criar conjuntos de treinamento, validação e teste;
  - Definir e treinar um modelo com definição de pesos de classe;
  - Avaliar o modelo usando várias métricas, incluindo precisão, revocação e F1;

2. Importação de bibliotecas

```
import tensorflow as tf
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import sklearn
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import train_test_split

tf.__version__

{"type":"string"}
```

3. Análise e processamento dos dados

Carregar dados ("Credit Card Fraud dataset")

Para carregar dados de arquivos tipo CSV a melhor ferramenta é o Pandas. O Pandas possui muitas funções úteis para processar dados estruturados.

```
raw_df =
pd.read_csv('https://storage.googleapis.com/download.tensorflow.org/
data/creditcard.csv')

# Mostra os 5 primeiros exemplos de dados
print(raw_df.shape)
raw_df.head(10)
```

```
(284807, 31)
```

{"type":"dataframe","variable_name":"raw_df"}

Estatística básica das colunas de características

```
raw_df.describe().T
```

{"summary":"{\n \"name\": \"raw_df\",\n \"rows\": 31,\n \"fields\":
[\n   {\n   \"column\": \"count\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 0.0,\n   \"min\":
284807.0,\n   \"max\": 284807.0,\n   \"num_unique_values\":
1,\n   \"samples\": [\n   284807.0\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"mean\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 17028.550324734668,\n
\"min\": -2.4063305498905906e-15,\n   \"max\":
94813.85957508067,\n   \"num_unique_values\": 31,\n
\"samples\": [\n   -3.6600908126037946e-16\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"std\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 8527.578758378544,\n
\"min\": 0.0415271896355952,\n   \"max\": 47488.14595456582,\n
\"num_unique_values\": 31,\n   \"samples\": [\n
0.4036324949650267\n   ],\n   \"semantic_type\": \"\",\n
\"description\": \"\"\n   }\n   },\n   {\n   \"column\":
\"min\",\n   \"properties\": {\n   \"dtype\": \"number\",\n
\"std\": 26.795994690128563,\n   \"min\": -113.743306711146,\n
\"max\": 0.0,\n   \"num_unique_values\": 29,\n
\"samples\": [\n   -22.5656793207827\n   ],\n
\"semantic_type\": \"\",\n   \"description\": \"\"\n   }\
n   },\n   {\n   \"column\": \"25%\",\n   \"properties\": {\n
\"dtype\": \"number\",\n   \"std\": 9734.925083048505,\n
\"min\": -0.920373384390322,\n   \"max\": 54201.5,\n
\"num_unique_values\": 31,\n   \"samples\": [\n   -

```
0.07083952930446921\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"50%\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 15211.001352467909,\n        \"min\": -0.274187076506651,\n
\"max\": 84692.0,\n        \"num_unique_values\": 31,\n
\"samples\": [\n          0.0013421459786502\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    },\n    {\n      \"column\": \"75%\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 25022.157936531003,\n
\"min\": 0.0,\n        \"max\": 139320.5,\n
\"num_unique_values\": 31,\n        \"samples\": [\n
0.09104511968580689\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"max\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 31218.887137498223,\n        \"min\": 1.0,\n        \"max\":
172792.0,\n        \"num_unique_values\": 31,\n        \"samples\": [\
n          31.6121981061363\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\
n}","type":"dataframe"}
```



```
print('Dimensões dos dados =', raw_df.shape)
```

```
Dimensões dos dados = (284807, 31)
```

Verificar o desbalanceamento dos dados

```
neg, pos = np.bincount(raw_df['Class'])
total = neg + pos
print('Examples:\n    Total: {}\n    Positive: {} ({:.2f}% of total)\
n'.format(
    total, pos, 100 * pos / total))
```

```
Examples:
    Total: 284807
    Positive: 492 (0.17% of total)
```

Esse resultado mostra a pequena fração dos dados da classe positiva (dados fraudulentos).

Limpeza inicial dos dados

Esses dados apresentam alguns problemas. Primeiro, as colunas `Time` e `Amount` apresentam grandes variações para serem usadas diretamente. Assim, vamos elimine a coluna `Time` (uma vez que não está claro o que significa) e vamos calcular o logaritmo da coluna `Amount` para reduzir seu intervalo de variação.

```
cleaned_df = raw_df.copy()

# Eliminação da coluna Time
cleaned_df.pop('Time')
```

```
# Cálculo do log da coluna Amount
eps=0.001 # deve-se somar um número pequeno para evitar calcular log
de zero
cleaned_df['LogAmmout'] = np.log(cleaned_df.pop('Amount')+eps)
cleaned_df.head()
```

{"type":"dataframe","variable_name":"cleaned_df"}

Divisão do conjunto de dados

Vamos dividir o conjunto de dados em conjuntos de treinamento, validação e teste. O conjunto de validação é usado durante o ajuste do modelo para avaliar a função de custo e outras métricas, no entanto, o modelo não se ajusta a esses dados. O conjunto de teste não é usado durante a fase de treinamento e só é usado no final para avaliar quão bem o modelo generaliza para novos dados. Isso é especialmente importante com conjuntos de dados desequilibrados, onde o sobreajuste é uma preocupação significativa devido à falta de dados de treinamento.

```
# Usaremos a função split da biblioteca sklearn para divir os dados
train_df, test_df = train_test_split(cleaned_df, test_size=0.3,
shuffle=True)
test_df, val_df = train_test_split(test_df, test_size=0.5)

# Separa as saídas dos dados de entrada e as transforma em tensores
Numpy
train_labels = np.array(train_df.pop('Class'))
val_labels = np.array(val_df.pop('Class'))
test_labels = np.array(test_df.pop('Class'))

# Transforma os dados de entrada em tensores Numpy
train_features = np.array(train_df)
val_features = np.array(val_df)
test_features = np.array(test_df)

print(train_features.shape, test_features.shape, val_features.shape)
print(train_labels.shape, test_labels.shape, val_labels.shape)
```

```
(199364, 29) (42721, 29) (42722, 29)
(199364,) (42721,) (42722,)
```

Normalização dos dados de entrada

Os dados de entrada serão normalizados para que cada característica (coluna) tenha média zero e desvio padrão igual a um.

As médias e desvios padrões de cada característica são calculados usando somente o conjunto de dados de treinamento e esses valores são usados para normalizar também os dados de validação e teste. Isso deve ser feito porque nenhum ainformação dos dados de validação e teste devem ser utilizados no treinamento.

```python
# Calcula média e desvio padrão de cada coluna dos dados de treinamento
mean = np.mean(train_features, axis=0)
std = np.std(train_features, axis=0)

# Normaliza dados de treinamento, validação e teste usando média e desvio padrão dos dados de treinamento
train_features = (train_features - mean)/std
val_features = (val_features - mean)/std
test_features = (test_features - mean)/std

print('Dimensão das saídas de treinamento:', train_labels.shape)
print('Dimensão das saídas de validação:', val_labels.shape)
print('Dimensão das saídas de teste:', test_labels.shape)

print('Dimensão das entradas de treinamento:', train_features.shape)
print('Dimensão das entradas de validação:', val_features.shape)
print('Dimensão das entradas de teste:', test_features.shape)
```

```
Dimensão das saídas de treinamento: (199364,)
Dimensão das saídas de validação: (42722,)
Dimensão das saídas de teste: (42721,)
Dimensão das entradas de treinamento: (199364, 29)
Dimensão das entradas de validação: (42722, 29)
Dimensão das entradas de teste: (42721, 29)
```

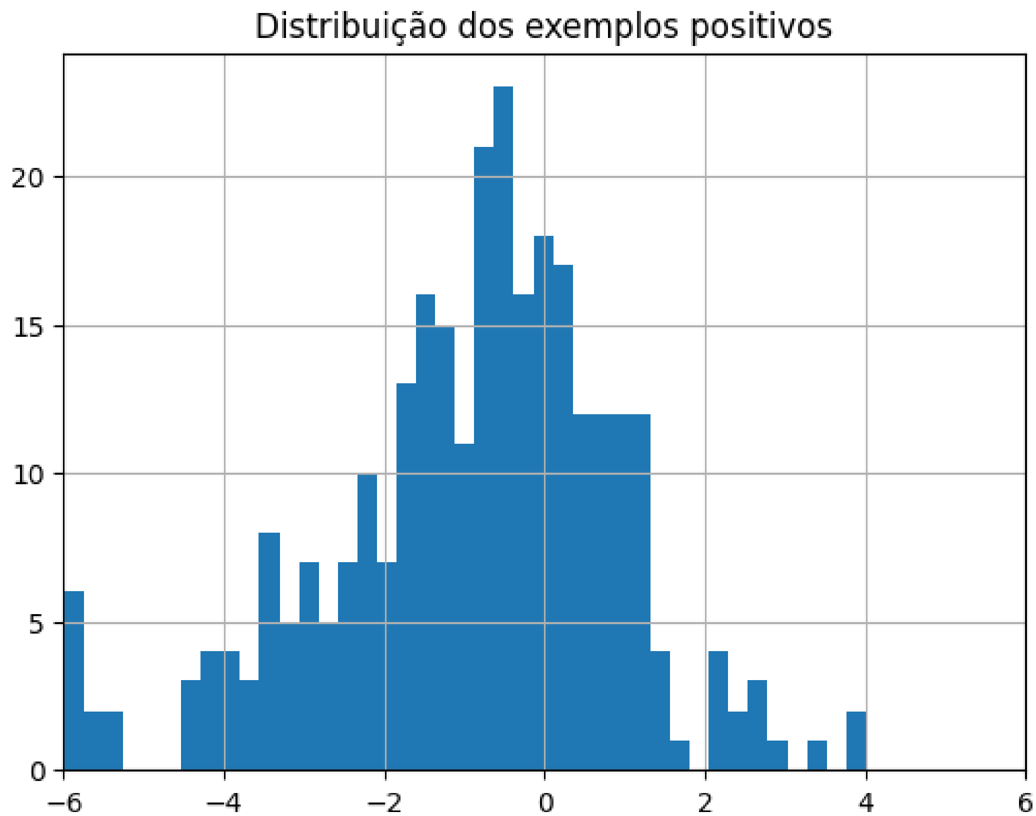
Distribuição dos dados normalizados

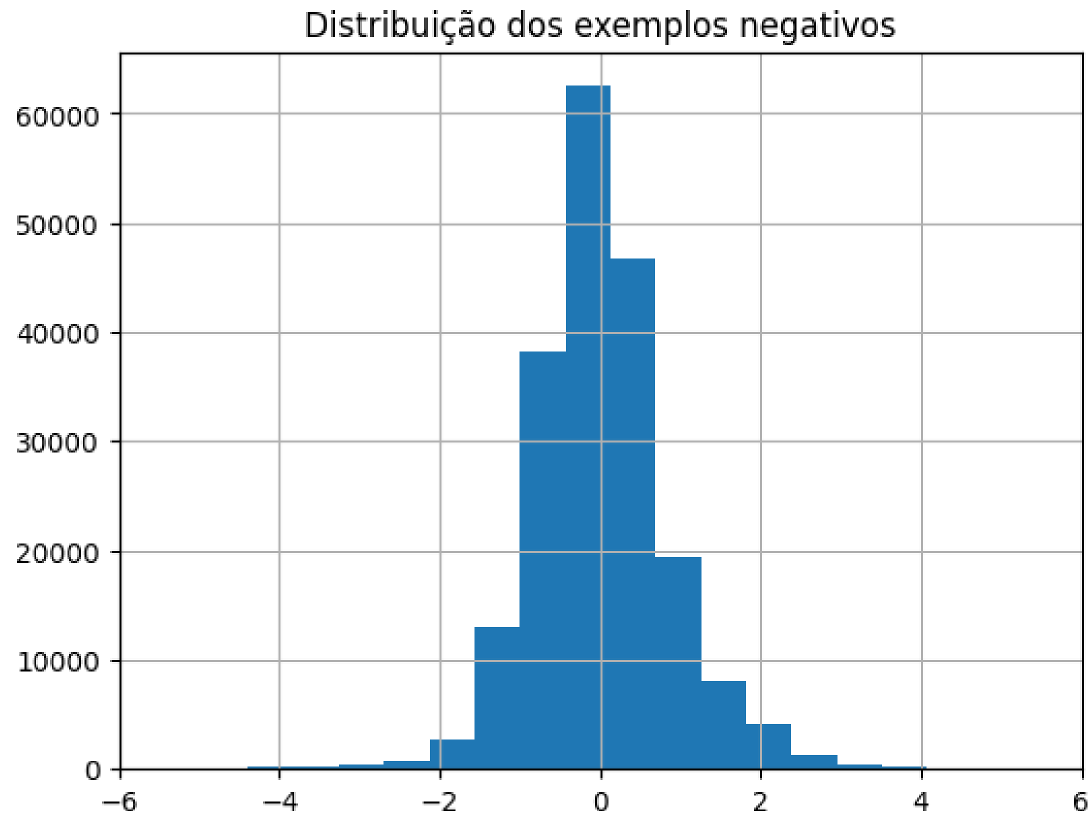
Vamos comparar as distribuições dos exemplos das classes positivo e negativo usando algumas características. As perguntas que se deve fazer neste momento são:

- Essas distribuições fazem sentido?
  - Sim. As entradas foram normalizadas e, portanto, os dados estão concentrados principalmente no intervalo `+/- 2`.
- Pode-se ver a diferença entre as distribuições?
  - Sim, os exemplos positivos contêm uma taxa muito maior de valores extremos.

```python
# Identifica exemplos positivos
bool_train_labels = train_labels != 0

# Separa exemplos positivos e negativos
pos_df = pd.DataFrame(train_features[ bool_train_labels], columns = train_df.columns)
neg_df = pd.DataFrame(train_features[~bool_train_labels], columns = train_df.columns)

# Faz gráficos da distribuição
plt.hist(pos_df['V5'], bins=100)
plt.title(("Distribuição dos exemplos positivos"))
plt.xlim([-6, +6])
```

```
plt.grid()
plt.show()
plt.hist(neg_df['V5'], bins=100)
plt.title("Distribuição dos exemplos negativos")
plt.xlim([-6, +6])
plt.grid()
plt.show()
```





4. Definição da RNA e das métricas

Vamos definir uma função que cria uma rede neural simples com uma camada oculta tipo densa e uma camada de saída com um único neurônio com função de ativação sigmóide, que retorna a probabilidade de uma transação ser fraudulenta.

```
# Importar do Keras classes de modelos e camadas
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense

# Define métricas
METRICS = [
      tf.keras.metrics.TruePositives(name='tp'),
      tf.keras.metrics.FalsePositives(name='fp'),
      tf.keras.metrics.TrueNegatives(name='tn'),
      tf.keras.metrics.FalseNegatives(name='fn'),
      tf.keras.metrics.BinaryAccuracy(name='accuracy'),
      tf.keras.metrics.Precision(name='precision'),
      tf.keras.metrics.Recall(name='recall'),
      tf.keras.metrics.AUC(name='auc')]


# Função que cria e compila a RN
def make_model(METRICS, INPUT_DIM):
```

```
    # Configuração da rede
    rna = Sequential()
    rna.add(Dense(units=32, activation='relu', input_dim=INPUT_DIM))
    rna.add(Dense(units=1, activation='sigmoid'))

    rna.compile(optimizer=tf.keras.optimizers.SGD(learning_rate=0.01),
                loss=tf.keras.losses.BinaryCrossentropy(),
                metrics=METRICS)

    return rna

# Determina número de carateríticas
features_shape = train_features.shape[1]
print('Dimensão dos dados de entrada =', features_shape)

# Cria RN já compilada
rna = make_model(METRICS, features_shape)
rna.summary()
```

```
Dimensão dos dados de entrada = 29
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/
dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)
```

```
Model: "sequential"
```

```
┏━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━┳
━━━━━━━━━━━━━━━━━┓
┃ Layer (type)                         ┃ Output Shape                 ┃
Param # ┃
┡━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━╇
━━━━━━━━━━━━━━━━━┩
│ dense (Dense)                        │ (None, 32)                   │
960 │
├──────────────────────────────────────┼──────────────────────────────┼
─────────────────┤
│ dense_1 (Dense)                      │ (None, 1)                    │
33 │
└──────────────────────────────────────┴──────────────────────────────┴
─────────────────┘
```

```
 Total params: 993 (3.88 KB)
```

```
 Trainable params: 993 (3.88 KB)
```

```
 Non-trainable params: 0 (0.00 B)
```

Métricas

- **Falsos negativos** e **falsos positivos** são exemplos classificados **incorretamente**.
- **Verdadeiros negativos** e **verdadeiros positivos** são exemplos classificados **corretamente**.
- **Exatidão ("accuracy")** é a porcentagem de exemplos classificados corretamente:

$$\frac{\text{exemplos classificados corretamente}}{\text{total de exemplos}}$$

- **Precisão ("precision")** é a porcentagem de exemplos positivos classificados corretamente:

$$\frac{\text{verdadeiros positivos}}{\text{verdadeiros positivos + falsos positivos}}$$

- **Revocação ("recall")** é a porcentagem de exemplos positivos reais que foram classificados corretamente:

$$\frac{\text{verdadeiros positivos}}{\text{verdadeiros positivos + falsos negativos}}$$

- **AUC** refere-se à área sob uma curva de característica de operação do receptor (ROC-AUC). Essa métrica é igual à probabilidade de que um classificador classifique uma amostra positiva mais alta do que uma amostra negativa ROC-AUC

Observação: a exatidão não é uma métrica útil para essa tarefa. Você pode ter uma precisão de 99,8% nesta tarefa prevendo Falso o tempo todo.

5. Resultado base

Treinamento da RN

Agora vamos treinar a RNA que foi definida anteriormente. Observe que o tamanho do lote de 4096 é bem maior do que o padrão de 32. Nesse tipo de problema isso é importante para garantir que cada lote tenha uma alguma chance de conter algumas amostras positivas. Se o tamanho do lote for muito pequeno, eles provavelmente não teriam transações fraudulentas com as quais aprender.

Observação: essa RNA não conseguirá lidar bem com o desequilíbrio de classe. Posteriormente vamos melhorar esse resultado usando pesos para as classes.

```
EPOCHS = 100
BATCH_SIZE = 4096

# Define callback para parada
early_stopping = tf.keras.callbacks.EarlyStopping(
    monitor='val_auc',
    verbose=1,
```

```
    patience=10,
    mode='max',
    restore_best_weights=True)

# Treinamento da RN
history = rna.fit(train_features, train_labels, epochs=EPOCHS,
batch_size=BATCH_SIZE,
                  validation_data=(val_features, val_labels),
verbose=1)#, callbacks=[early_stopping])
```

```
Epoch 1/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 9s 83ms/step - accuracy: 0.6188 - auc:
0.4420 - fn: 118.0600 - fp: 34627.8594 - loss: 0.6457 - precision:
0.0019 - recall: 0.4122 - tn: 69500.9375 - tp: 65.6200 - val_accuracy:
0.9297 - val_auc: 0.3524 - val_fn: 61.0000 - val_fp: 2943.0000 -
val_loss: 0.3681 - val_precision: 0.0030 - val_recall: 0.1286 -
val_tn: 39709.0000 - val_tp: 9.0000
Epoch 2/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9573 - auc:
0.3057 - fn: 171.3000 - fp: 3576.5400 - loss: 0.3265 - precision:
0.0054 - recall: 0.1132 - tn: 100543.7969 - tp: 20.8400 -
val_accuracy: 0.9941 - val_auc: 0.2964 - val_fn: 65.0000 - val_fp:
188.0000 - val_loss: 0.2295 - val_precision: 0.0259 - val_recall:
0.0714 - val_tn: 42464.0000 - val_tp: 5.0000
Epoch 3/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9964 - auc:
0.2683 - fn: 167.3400 - fp: 165.7400 - loss: 0.2113 - precision:
0.0445 - recall: 0.0506 - tn: 103970.7812 - tp: 8.6200 - val_accuracy:
0.9984 - val_auc: 0.2726 - val_fn: 69.0000 - val_fp: 1.0000 -
val_loss: 0.1654 - val_precision: 0.5000 - val_recall: 0.0143 -
val_tn: 42651.0000 - val_tp: 1.0000
Epoch 4/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9982 - auc:
0.2319 - fn: 184.2400 - fp: 2.1800 - loss: 0.1567 - precision: 0.2883
- recall: 0.0055 - tn: 104125.1797 - tp: 0.8800 - val_accuracy: 0.9984
- val_auc: 0.2627 - val_fn: 70.0000 - val_fp: 0.0000e+00 - val_loss:
0.1292 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00 - val_tn:
42652.0000 - val_tp: 0.0000e+00
Epoch 5/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9983 - auc:
0.2660 - fn: 178.3200 - fp: 0.0000e+00 - loss: 0.1236 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104134.1562 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2595 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.1062 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 6/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9980 - auc:
0.2548 - fn: 194.2400 - fp: 0.0000e+00 - loss: 0.1038 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104118.2422 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2616 - val_fn: 70.0000 - val_fp:
```

```
0.0000e+00 - val_loss: 0.0903 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 7/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9984 - auc:
0.2371 - fn: 174.7000 - fp: 0.0000e+00 - loss: 0.0878 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104137.7812 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2668 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0788 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 8/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.9981 - auc:
0.2334 - fn: 186.6000 - fp: 0.0000e+00 - loss: 0.0790 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104125.8828 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2740 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0700 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 9/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9983 - auc:
0.2720 - fn: 184.0400 - fp: 0.0000e+00 - loss: 0.0687 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104128.4375 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2865 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0630 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 10/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9982 - auc:
0.2812 - fn: 190.3800 - fp: 0.0000e+00 - loss: 0.0641 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104122.1016 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.2985 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0575 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 11/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.9983 - auc:
0.2927 - fn: 178.3000 - fp: 0.0000e+00 - loss: 0.0572 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104134.1797 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.3140 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0528 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 12/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9982 - auc:
0.2974 - fn: 185.2800 - fp: 0.0000e+00 - loss: 0.0529 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104127.2031 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.3317 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0490 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 13/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9984 - auc:
0.3298 - fn: 175.3200 - fp: 0.0000e+00 - loss: 0.0477 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104137.1562 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.3460 - val_fn: 70.0000 - val_fp:
```

```
0.0000e+00 - val_loss: 0.0457 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 14/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9983 - auc:
0.3576 - fn: 184.6800 - fp: 0.0000e+00 - loss: 0.0458 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104127.7969 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.3731 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0428 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 15/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9982 - auc:
0.3932 - fn: 190.1400 - fp: 0.0000e+00 - loss: 0.0436 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104122.3438 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.3901 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0403 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 16/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.9983 - auc:
0.3707 - fn: 180.8200 - fp: 0.0000e+00 - loss: 0.0413 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104131.6562 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.4156 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0381 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 17/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.9982 - auc:
0.4270 - fn: 184.8400 - fp: 0.0000e+00 - loss: 0.0385 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104127.6406 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.4371 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0361 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 18/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9983 - auc:
0.4544 - fn: 182.2400 - fp: 0.0000e+00 - loss: 0.0360 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104130.2422 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.4621 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0343 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 19/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.9982 - auc:
0.4681 - fn: 184.0800 - fp: 0.0000e+00 - loss: 0.0350 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104128.3984 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.4890 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0327 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 20/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.9982 - auc:
0.5136 - fn: 182.7200 - fp: 0.0000e+00 - loss: 0.0328 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104129.7578 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.5185 - val_fn: 70.0000 - val_fp:
```

```
0.0000e+00 - val_loss: 0.0312 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 21/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 8ms/step - accuracy: 0.9983 - auc:
0.5198 - fn: 186.5200 - fp: 0.0000e+00 - loss: 0.0311 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104125.9609 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.5321 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0298 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 22/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9983 - auc:
0.5750 - fn: 175.1200 - fp: 0.0000e+00 - loss: 0.0290 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104137.3594 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.5574 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0286 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 23/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9983 - auc:
0.5839 - fn: 178.3400 - fp: 0.0000e+00 - loss: 0.0285 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104134.1406 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.5799 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0274 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 24/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9981 - auc:
0.5960 - fn: 187.6200 - fp: 0.0000e+00 - loss: 0.0283 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104124.8594 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.6022 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0264 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 25/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9983 - auc:
0.6412 - fn: 179.0600 - fp: 0.0000e+00 - loss: 0.0260 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104133.4219 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.6206 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0254 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 26/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9982 - auc:
0.6668 - fn: 183.5000 - fp: 0.0000e+00 - loss: 0.0253 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104128.9766 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.6510 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0244 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 27/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9981 - auc:
0.6776 - fn: 191.7600 - fp: 0.0000e+00 - loss: 0.0243 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104120.7188 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.6704 - val_fn: 70.0000 - val_fp:
```

```
0.0000e+00 - val_loss: 0.0236 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 28/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9983 - auc:
0.7213 - fn: 178.2600 - fp: 0.0000e+00 - loss: 0.0232 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104134.2188 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.6859 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0228 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 29/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9982 - auc:
0.7180 - fn: 184.5600 - fp: 0.0000e+00 - loss: 0.0226 - precision:
0.0000e+00 - recall: 0.0000e+00 - tn: 104127.9219 - tp: 0.0000e+00 -
val_accuracy: 0.9984 - val_auc: 0.7094 - val_fn: 70.0000 - val_fp:
0.0000e+00 - val_loss: 0.0220 - val_precision: 0.0000e+00 -
val_recall: 0.0000e+00 - val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 30/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9982 - auc:
0.7375 - fn: 185.8400 - fp: 0.0000e+00 - loss: 0.0220 - precision:
0.4400 - recall: 0.0016 - tn: 104126.2031 - tp: 0.4400 - val_accuracy:
0.9984 - val_auc: 0.7240 - val_fn: 70.0000 - val_fp: 0.0000e+00 -
val_loss: 0.0213 - val_precision: 0.0000e+00 - val_recall: 0.0000e+00
- val_tn: 42652.0000 - val_tp: 0.0000e+00
Epoch 31/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9982 - auc:
0.7720 - fn: 189.6000 - fp: 0.0000e+00 - loss: 0.0212 - precision:
0.5800 - recall: 0.0025 - tn: 104122.2422 - tp: 0.6400 - val_accuracy:
0.9984 - val_auc: 0.7363 - val_fn: 68.0000 - val_fp: 0.0000e+00 -
val_loss: 0.0206 - val_precision: 1.0000 - val_recall: 0.0286 -
val_tn: 42652.0000 - val_tp: 2.0000
Epoch 32/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9982 - auc:
0.7383 - fn: 181.1800 - fp: 0.0000e+00 - loss: 0.0209 - precision:
0.7000 - recall: 0.0066 - tn: 104129.5391 - tp: 1.7600 - val_accuracy:
0.9984 - val_auc: 0.7400 - val_fn: 68.0000 - val_fp: 0.0000e+00 -
val_loss: 0.0200 - val_precision: 1.0000 - val_recall: 0.0286 -
val_tn: 42652.0000 - val_tp: 2.0000
Epoch 33/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9984 - auc:
0.8326 - fn: 177.5400 - fp: 0.0000e+00 - loss: 0.0187 - precision:
0.7600 - recall: 0.0152 - tn: 104131.1016 - tp: 3.8400 - val_accuracy:
0.9984 - val_auc: 0.7449 - val_fn: 67.0000 - val_fp: 1.0000 -
val_loss: 0.0194 - val_precision: 0.7500 - val_recall: 0.0429 -
val_tn: 42651.0000 - val_tp: 3.0000
Epoch 34/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9983 - auc:
0.7997 - fn: 175.0800 - fp: 1.7200 - loss: 0.0187 - precision: 0.7595
- recall: 0.0252 - tn: 104130.2812 - tp: 5.4000 - val_accuracy: 0.9984
- val_auc: 0.7485 - val_fn: 67.0000 - val_fp: 1.0000 - val_loss:
```

```
0.0188 - val_precision: 0.7500 - val_recall: 0.0429 - val_tn:
42651.0000 - val_tp: 3.0000
Epoch 35/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9983 - auc:
0.8064 - fn: 178.1400 - fp: 2.4800 - loss: 0.0189 - precision: 0.7837
- recall: 0.0429 - tn: 104122.7812 - tp: 9.0800 - val_accuracy: 0.9984
- val_auc: 0.7546 - val_fn: 66.0000 - val_fp: 1.0000 - val_loss:
0.0183 - val_precision: 0.8000 - val_recall: 0.0571 - val_tn:
42651.0000 - val_tp: 4.0000
Epoch 36/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9984 - auc:
0.8183 - fn: 170.5600 - fp: 4.4000 - loss: 0.0173 - precision: 0.7598
- recall: 0.0575 - tn: 104126.9375 - tp: 10.5800 - val_accuracy:
0.9984 - val_auc: 0.7531 - val_fn: 65.0000 - val_fp: 2.0000 -
val_loss: 0.0178 - val_precision: 0.7143 - val_recall: 0.0714 -
val_tn: 42650.0000 - val_tp: 5.0000
Epoch 37/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9982 - auc:
0.8149 - fn: 166.9000 - fp: 8.8200 - loss: 0.0175 - precision: 0.5927
- recall: 0.0751 - tn: 104123.4766 - tp: 13.2800 - val_accuracy:
0.9985 - val_auc: 0.7542 - val_fn: 61.0000 - val_fp: 2.0000 -
val_loss: 0.0174 - val_precision: 0.8182 - val_recall: 0.1286 -
val_tn: 42650.0000 - val_tp: 9.0000
Epoch 38/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9984 - auc:
0.8035 - fn: 159.7800 - fp: 10.3200 - loss: 0.0165 - precision: 0.5398
- recall: 0.0844 - tn: 104125.4219 - tp: 16.9600 - val_accuracy:
0.9986 - val_auc: 0.7566 - val_fn: 57.0000 - val_fp: 2.0000 -
val_loss: 0.0170 - val_precision: 0.8667 - val_recall: 0.1857 -
val_tn: 42650.0000 - val_tp: 13.0000
Epoch 39/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9985 - auc:
0.8262 - fn: 155.8600 - fp: 9.1400 - loss: 0.0160 - precision: 0.7247
- recall: 0.1267 - tn: 104127.3438 - tp: 20.1400 - val_accuracy:
0.9986 - val_auc: 0.7630 - val_fn: 56.0000 - val_fp: 4.0000 -
val_loss: 0.0166 - val_precision: 0.7778 - val_recall: 0.2000 -
val_tn: 42648.0000 - val_tp: 14.0000
Epoch 40/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9983 - auc:
0.8157 - fn: 157.1200 - fp: 14.0200 - loss: 0.0164 - precision: 0.6325
- recall: 0.1273 - tn: 104116.1016 - tp: 25.2400 - val_accuracy:
0.9986 - val_auc: 0.7653 - val_fn: 54.0000 - val_fp: 5.0000 -
val_loss: 0.0162 - val_precision: 0.7619 - val_recall: 0.2286 -
val_tn: 42647.0000 - val_tp: 16.0000
Epoch 41/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9985 - auc:
0.8284 - fn: 143.4200 - fp: 12.0000 - loss: 0.0151 - precision: 0.7053
- recall: 0.1790 - tn: 104126.2812 - tp: 30.7800 - val_accuracy:
0.9986 - val_auc: 0.7675 - val_fn: 54.0000 - val_fp: 5.0000 -
```

```
val_loss: 0.0159 - val_precision: 0.7619 - val_recall: 0.2286 -
val_tn: 42647.0000 - val_tp: 16.0000
Epoch 42/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9984 - auc:
0.8100 - fn: 146.7400 - fp: 15.7400 - loss: 0.0160 - precision: 0.7315
- recall: 0.2325 - tn: 104108.3594 - tp: 41.6400 - val_accuracy:
0.9986 - val_auc: 0.7698 - val_fn: 54.0000 - val_fp: 5.0000 -
val_loss: 0.0155 - val_precision: 0.7619 - val_recall: 0.2286 -
val_tn: 42647.0000 - val_tp: 16.0000
Epoch 43/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9986 - auc:
0.8147 - fn: 132.9000 - fp: 13.1200 - loss: 0.0149 - precision: 0.7627
- recall: 0.2569 - tn: 104120.8984 - tp: 45.5600 - val_accuracy:
0.9986 - val_auc: 0.7739 - val_fn: 53.0000 - val_fp: 5.0000 -
val_loss: 0.0152 - val_precision: 0.7727 - val_recall: 0.2429 -
val_tn: 42647.0000 - val_tp: 17.0000
Epoch 44/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9985 - auc:
0.8364 - fn: 137.5800 - fp: 13.6200 - loss: 0.0150 - precision: 0.8173
- recall: 0.2909 - tn: 104105.3594 - tp: 55.9200 - val_accuracy:
0.9987 - val_auc: 0.7760 - val_fn: 52.0000 - val_fp: 5.0000 -
val_loss: 0.0150 - val_precision: 0.7826 - val_recall: 0.2571 -
val_tn: 42647.0000 - val_tp: 18.0000
Epoch 45/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9987 - auc:
0.8410 - fn: 123.1600 - fp: 15.3600 - loss: 0.0140 - precision: 0.7650
- recall: 0.3109 - tn: 104120.3203 - tp: 53.6400 - val_accuracy:
0.9987 - val_auc: 0.7781 - val_fn: 51.0000 - val_fp: 5.0000 -
val_loss: 0.0147 - val_precision: 0.7917 - val_recall: 0.2714 -
val_tn: 42647.0000 - val_tp: 19.0000
Epoch 46/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9986 - auc:
0.8449 - fn: 129.0200 - fp: 14.1200 - loss: 0.0143 - precision: 0.8108
- recall: 0.3096 - tn: 104111.5391 - tp: 57.8000 - val_accuracy:
0.9987 - val_auc: 0.7830 - val_fn: 50.0000 - val_fp: 5.0000 -
val_loss: 0.0144 - val_precision: 0.8000 - val_recall: 0.2857 -
val_tn: 42647.0000 - val_tp: 20.0000
Epoch 47/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 9ms/step - accuracy: 0.9986 - auc:
0.8264 - fn: 127.7200 - fp: 14.4400 - loss: 0.0141 - precision: 0.8107
- recall: 0.2797 - tn: 104115.8594 - tp: 54.4600 - val_accuracy:
0.9987 - val_auc: 0.7852 - val_fn: 49.0000 - val_fp: 5.0000 -
val_loss: 0.0142 - val_precision: 0.8077 - val_recall: 0.3000 -
val_tn: 42647.0000 - val_tp: 21.0000
Epoch 48/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.9987 - auc:
0.8354 - fn: 119.8600 - fp: 16.2200 - loss: 0.0132 - precision: 0.7236
- recall: 0.3131 - tn: 104121.3438 - tp: 55.0600 - val_accuracy:
0.9987 - val_auc: 0.7873 - val_fn: 49.0000 - val_fp: 5.0000 -
```

```
val_loss: 0.0140 - val_precision: 0.8077 - val_recall: 0.3000 -
val_tn: 42647.0000 - val_tp: 21.0000
Epoch 49/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 10ms/step - accuracy: 0.9987 - auc:
0.8218 - fn: 122.0600 - fp: 14.8600 - loss: 0.0134 - precision: 0.8115
- recall: 0.3249 - tn: 104116.7422 - tp: 58.8200 - val_accuracy:
0.9987 - val_auc: 0.7924 - val_fn: 49.0000 - val_fp: 5.0000 -
val_loss: 0.0137 - val_precision: 0.8077 - val_recall: 0.3000 -
val_tn: 42647.0000 - val_tp: 21.0000
Epoch 50/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.9988 - auc:
0.8380 - fn: 113.3800 - fp: 15.2800 - loss: 0.0125 - precision: 0.7905
- recall: 0.3497 - tn: 104122.2031 - tp: 61.6200 - val_accuracy:
0.9988 - val_auc: 0.7944 - val_fn: 48.0000 - val_fp: 5.0000 -
val_loss: 0.0135 - val_precision: 0.8148 - val_recall: 0.3143 -
val_tn: 42647.0000 - val_tp: 22.0000
Epoch 51/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9986 - auc:
0.8537 - fn: 125.6400 - fp: 12.1800 - loss: 0.0133 - precision: 0.8517
- recall: 0.3711 - tn: 104103.5391 - tp: 71.1200 - val_accuracy:
0.9988 - val_auc: 0.7935 - val_fn: 47.0000 - val_fp: 5.0000 -
val_loss: 0.0133 - val_precision: 0.8214 - val_recall: 0.3286 -
val_tn: 42647.0000 - val_tp: 23.0000
Epoch 52/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9987 - auc:
0.8521 - fn: 118.3800 - fp: 13.7200 - loss: 0.0130 - precision: 0.8462
- recall: 0.3656 - tn: 104110.9375 - tp: 69.4400 - val_accuracy:
0.9988 - val_auc: 0.7952 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0131 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 53/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - accuracy: 0.9988 - auc:
0.8265 - fn: 113.6000 - fp: 11.8000 - loss: 0.0125 - precision: 0.8447
- recall: 0.3706 - tn: 104120.6172 - tp: 66.4600 - val_accuracy:
0.9988 - val_auc: 0.7924 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0129 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 54/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9988 - auc:
0.8279 - fn: 108.5000 - fp: 15.8600 - loss: 0.0125 - precision: 0.8446
- recall: 0.4202 - tn: 104112.0625 - tp: 76.0600 - val_accuracy:
0.9988 - val_auc: 0.7914 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0128 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 55/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8332 - fn: 107.6400 - fp: 15.6200 - loss: 0.0116 - precision: 0.8036
- recall: 0.3986 - tn: 104116.6172 - tp: 72.6000 - val_accuracy:
0.9988 - val_auc: 0.7930 - val_fn: 46.0000 - val_fp: 5.0000 -
```

```
val_loss: 0.0126 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 56/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9988 - auc:
0.8480 - fn: 108.6600 - fp: 14.3000 - loss: 0.0117 - precision: 0.8301
- recall: 0.3928 - tn: 104115.1562 - tp: 74.3600 - val_accuracy:
0.9988 - val_auc: 0.7863 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0124 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 57/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9988 - auc:
0.8547 - fn: 105.3600 - fp: 17.4400 - loss: 0.0114 - precision: 0.8064
- recall: 0.4236 - tn: 104110.0781 - tp: 79.6000 - val_accuracy:
0.9988 - val_auc: 0.7907 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0123 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 58/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9988 - auc:
0.8519 - fn: 107.1600 - fp: 16.6200 - loss: 0.0117 - precision: 0.8079
- recall: 0.4083 - tn: 104112.7422 - tp: 75.9600 - val_accuracy:
0.9988 - val_auc: 0.7921 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0121 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 59/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8436 - fn: 98.9200 - fp: 17.0200 - loss: 0.0111 - precision: 0.8115
- recall: 0.4258 - tn: 104122.3438 - tp: 74.2000 - val_accuracy:
0.9988 - val_auc: 0.7935 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0120 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 60/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8409 - fn: 96.1800 - fp: 14.3800 - loss: 0.0107 - precision: 0.8588
- recall: 0.4711 - tn: 104124.2188 - tp: 77.7000 - val_accuracy:
0.9988 - val_auc: 0.7950 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0118 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 61/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8346 - fn: 99.5400 - fp: 15.8200 - loss: 0.0113 - precision: 0.8336
- recall: 0.4314 - tn: 104118.4219 - tp: 78.7000 - val_accuracy:
0.9988 - val_auc: 0.7934 - val_fn: 46.0000 - val_fp: 5.0000 -
val_loss: 0.0117 - val_precision: 0.8276 - val_recall: 0.3429 -
val_tn: 42647.0000 - val_tp: 24.0000
Epoch 62/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8691 - fn: 94.5000 - fp: 13.7200 - loss: 0.0101 - precision: 0.8663
- recall: 0.4554 - tn: 104124.2969 - tp: 79.9600 - val_accuracy:
0.9989 - val_auc: 0.7949 - val_fn: 44.0000 - val_fp: 5.0000 -
```

```
val_loss: 0.0116 - val_precision: 0.8387 - val_recall: 0.3714 -
val_tn: 42647.0000 - val_tp: 26.0000
Epoch 63/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8571 - fn: 98.1400 - fp: 15.4600 - loss: 0.0106 - precision: 0.8591
- recall: 0.4843 - tn: 104111.9766 - tp: 86.9000 - val_accuracy:
0.9989 - val_auc: 0.7962 - val_fn: 42.0000 - val_fp: 5.0000 -
val_loss: 0.0114 - val_precision: 0.8485 - val_recall: 0.4000 -
val_tn: 42647.0000 - val_tp: 28.0000
Epoch 64/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8692 - fn: 94.8800 - fp: 14.3600 - loss: 0.0102 - precision: 0.8498
- recall: 0.4523 - tn: 104122.8984 - tp: 80.3400 - val_accuracy:
0.9989 - val_auc: 0.8035 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0113 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 65/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8728 - fn: 98.2800 - fp: 12.4800 - loss: 0.0099 - precision: 0.8959
- recall: 0.4589 - tn: 104116.2188 - tp: 85.5000 - val_accuracy:
0.9989 - val_auc: 0.8049 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0112 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 66/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9988 - auc:
0.8643 - fn: 102.7000 - fp: 15.8600 - loss: 0.0106 - precision: 0.8161
- recall: 0.4186 - tn: 104108.5391 - tp: 85.3800 - val_accuracy:
0.9989 - val_auc: 0.8061 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0111 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 67/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9991 - auc:
0.8679 - fn: 92.1800 - fp: 13.0200 - loss: 0.0101 - precision: 0.8915
- recall: 0.5306 - tn: 104116.7969 - tp: 90.4800 - val_accuracy:
0.9989 - val_auc: 0.8074 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0110 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 68/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9990 - auc:
0.8667 - fn: 93.9200 - fp: 14.1000 - loss: 0.0100 - precision: 0.8576
- recall: 0.4783 - tn: 104120.2969 - tp: 84.1600 - val_accuracy:
0.9989 - val_auc: 0.8088 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0109 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 69/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9988 - auc:
0.8614 - fn: 96.4800 - fp: 16.6400 - loss: 0.0102 - precision: 0.8177
- recall: 0.4616 - tn: 104112.2188 - tp: 87.1400 - val_accuracy:
0.9989 - val_auc: 0.8101 - val_fn: 40.0000 - val_fp: 5.0000 -
```

```
val_loss: 0.0108 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 70/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8824 - fn: 94.5400 - fp: 18.2400 - loss: 0.0097 - precision: 0.8092
- recall: 0.4780 - tn: 104112.5000 - tp: 87.2000 - val_accuracy:
0.9989 - val_auc: 0.8113 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0107 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 71/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8616 - fn: 91.6600 - fp: 15.3400 - loss: 0.0096 - precision: 0.8589
- recall: 0.4862 - tn: 104118.6172 - tp: 86.8600 - val_accuracy:
0.9989 - val_auc: 0.8126 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0106 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 72/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8650 - fn: 90.3400 - fp: 15.4600 - loss: 0.0093 - precision: 0.8268
- recall: 0.4977 - tn: 104119.7812 - tp: 86.9000 - val_accuracy:
0.9989 - val_auc: 0.8137 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0105 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 73/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8793 - fn: 84.1800 - fp: 13.7000 - loss: 0.0089 - precision: 0.8858
- recall: 0.5455 - tn: 104122.6797 - tp: 91.9200 - val_accuracy:
0.9989 - val_auc: 0.8148 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0104 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 74/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8681 - fn: 89.2600 - fp: 15.6200 - loss: 0.0096 - precision: 0.8586
- recall: 0.4996 - tn: 104120.6406 - tp: 86.9600 - val_accuracy:
0.9989 - val_auc: 0.8167 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0103 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 75/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8849 - fn: 87.1800 - fp: 15.4400 - loss: 0.0091 - precision: 0.8599
- recall: 0.5445 - tn: 104118.1797 - tp: 91.6800 - val_accuracy:
0.9989 - val_auc: 0.8179 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0102 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 76/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8905 - fn: 92.1400 - fp: 15.1200 - loss: 0.0088 - precision: 0.8662
- recall: 0.5238 - tn: 104107.5234 - tp: 97.7000 - val_accuracy:
0.9989 - val_auc: 0.8189 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0101 - val_precision: 0.8571 - val_recall: 0.4286 -
```

```
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 77/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8730 - fn: 85.3200 - fp: 16.0400 - loss: 0.0091 - precision: 0.8612
- recall: 0.5270 - tn: 104112.8438 - tp: 98.2800 - val_accuracy:
0.9989 - val_auc: 0.8200 - val_fn: 40.0000 - val_fp: 5.0000 -
val_loss: 0.0100 - val_precision: 0.8571 - val_recall: 0.4286 -
val_tn: 42647.0000 - val_tp: 30.0000
Epoch 78/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8405 - fn: 86.2600 - fp: 14.6200 - loss: 0.0095 - precision: 0.8811
- recall: 0.5331 - tn: 104115.5625 - tp: 96.0400 - val_accuracy:
0.9990 - val_auc: 0.8211 - val_fn: 39.0000 - val_fp: 5.0000 -
val_loss: 0.0100 - val_precision: 0.8611 - val_recall: 0.4429 -
val_tn: 42647.0000 - val_tp: 31.0000
Epoch 79/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - accuracy: 0.9990 - auc:
0.8616 - fn: 84.3400 - fp: 17.7000 - loss: 0.0090 - precision: 0.8297
- recall: 0.5146 - tn: 104119.0234 - tp: 91.4200 - val_accuracy:
0.9990 - val_auc: 0.8222 - val_fn: 39.0000 - val_fp: 5.0000 -
val_loss: 0.0099 - val_precision: 0.8611 - val_recall: 0.4429 -
val_tn: 42647.0000 - val_tp: 31.0000
Epoch 80/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9989 - auc:
0.8746 - fn: 96.4600 - fp: 12.1400 - loss: 0.0095 - precision: 0.9114
- recall: 0.5091 - tn: 104107.1406 - tp: 96.7400 - val_accuracy:
0.9990 - val_auc: 0.8234 - val_fn: 39.0000 - val_fp: 5.0000 -
val_loss: 0.0098 - val_precision: 0.8611 - val_recall: 0.4429 -
val_tn: 42647.0000 - val_tp: 31.0000
Epoch 81/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9990 - auc:
0.8726 - fn: 85.5200 - fp: 19.4600 - loss: 0.0089 - precision: 0.8047
- recall: 0.5235 - tn: 104113.4375 - tp: 94.0600 - val_accuracy:
0.9990 - val_auc: 0.8243 - val_fn: 39.0000 - val_fp: 5.0000 -
val_loss: 0.0097 - val_precision: 0.8611 - val_recall: 0.4429 -
val_tn: 42647.0000 - val_tp: 31.0000
Epoch 82/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8814 - fn: 92.0400 - fp: 15.1800 - loss: 0.0087 - precision: 0.8872
- recall: 0.5229 - tn: 104105.7031 - tp: 99.5600 - val_accuracy:
0.9990 - val_auc: 0.8253 - val_fn: 38.0000 - val_fp: 5.0000 -
val_loss: 0.0097 - val_precision: 0.8649 - val_recall: 0.4571 -
val_tn: 42647.0000 - val_tp: 32.0000
Epoch 83/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.9072 - fn: 79.6600 - fp: 15.3000 - loss: 0.0079 - precision: 0.8736
- recall: 0.5828 - tn: 104114.0391 - tp: 103.4800 - val_accuracy:
0.9990 - val_auc: 0.8264 - val_fn: 38.0000 - val_fp: 5.0000 -
val_loss: 0.0096 - val_precision: 0.8649 - val_recall: 0.4571 -
```

```
val_tn: 42647.0000 - val_tp: 32.0000
Epoch 84/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8871 - fn: 77.1800 - fp: 16.9000 - loss: 0.0082 - precision: 0.8337
- recall: 0.5510 - tn: 104123.8438 - tp: 94.5600 - val_accuracy:
0.9990 - val_auc: 0.8273 - val_fn: 38.0000 - val_fp: 5.0000 -
val_loss: 0.0095 - val_precision: 0.8649 - val_recall: 0.4571 -
val_tn: 42647.0000 - val_tp: 32.0000
Epoch 85/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9990 - auc:
0.8550 - fn: 87.6800 - fp: 18.3200 - loss: 0.0094 - precision: 0.8516
- recall: 0.5299 - tn: 104107.7188 - tp: 98.7600 - val_accuracy:
0.9990 - val_auc: 0.8284 - val_fn: 37.0000 - val_fp: 5.0000 -
val_loss: 0.0095 - val_precision: 0.8684 - val_recall: 0.4714 -
val_tn: 42647.0000 - val_tp: 33.0000
Epoch 86/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9991 - auc:
0.8846 - fn: 87.2800 - fp: 13.4400 - loss: 0.0081 - precision: 0.8990
- recall: 0.5316 - tn: 104114.9375 - tp: 96.8200 - val_accuracy:
0.9990 - val_auc: 0.8292 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0094 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 87/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9991 - auc:
0.9082 - fn: 75.7600 - fp: 17.1000 - loss: 0.0080 - precision: 0.8618
- recall: 0.5834 - tn: 104117.2422 - tp: 102.3800 - val_accuracy:
0.9990 - val_auc: 0.8303 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0093 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 88/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9991 - auc:
0.8970 - fn: 80.1400 - fp: 17.1200 - loss: 0.0076 - precision: 0.8566
- recall: 0.5533 - tn: 104119.3594 - tp: 95.8600 - val_accuracy:
0.9990 - val_auc: 0.8332 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0093 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 89/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9990 - auc:
0.8903 - fn: 83.1000 - fp: 17.4800 - loss: 0.0081 - precision: 0.8523
- recall: 0.5531 - tn: 104110.7188 - tp: 101.1800 - val_accuracy:
0.9990 - val_auc: 0.8341 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0092 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 90/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9989 - auc:
0.8811 - fn: 89.3800 - fp: 19.1200 - loss: 0.0088 - precision: 0.8416
- recall: 0.5254 - tn: 104098.7969 - tp: 105.1800 - val_accuracy:
0.9990 - val_auc: 0.8349 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0092 - val_precision: 0.8718 - val_recall: 0.4857 -
```

```
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 91/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8806 - fn: 81.1400 - fp: 15.8800 - loss: 0.0082 - precision: 0.8794
- recall: 0.5280 - tn: 104113.8828 - tp: 101.5800 - val_accuracy:
0.9990 - val_auc: 0.8359 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0091 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 92/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9992 - auc:
0.8806 - fn: 75.5000 - fp: 17.0400 - loss: 0.0077 - precision: 0.8555
- recall: 0.5922 - tn: 104120.6172 - tp: 99.3200 - val_accuracy:
0.9990 - val_auc: 0.8368 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0090 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 93/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8936 - fn: 85.5800 - fp: 15.9800 - loss: 0.0081 - precision: 0.8857
- recall: 0.5490 - tn: 104111.7188 - tp: 99.2000 - val_accuracy:
0.9990 - val_auc: 0.8376 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0090 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 94/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9991 - auc:
0.8887 - fn: 80.2200 - fp: 16.3400 - loss: 0.0082 - precision: 0.8795
- recall: 0.5880 - tn: 104103.1406 - tp: 112.7800 - val_accuracy:
0.9990 - val_auc: 0.8383 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0089 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 95/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8956 - fn: 87.4200 - fp: 19.3800 - loss: 0.0082 - precision: 0.8398
- recall: 0.5303 - tn: 104102.9219 - tp: 102.7600 - val_accuracy:
0.9990 - val_auc: 0.8391 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0089 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 96/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.9114 - fn: 79.6200 - fp: 16.1200 - loss: 0.0074 - precision: 0.8784
- recall: 0.5583 - tn: 104117.2031 - tp: 99.5400 - val_accuracy:
0.9990 - val_auc: 0.8398 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0088 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 97/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8972 - fn: 78.5400 - fp: 19.7800 - loss: 0.0078 - precision: 0.8286
- recall: 0.6041 - tn: 104106.8984 - tp: 107.2600 - val_accuracy:
0.9990 - val_auc: 0.8405 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0088 - val_precision: 0.8718 - val_recall: 0.4857 -
```

```
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 98/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8859 - fn: 84.4800 - fp: 17.2000 - loss: 0.0078 - precision: 0.8459
- recall: 0.5154 - tn: 104112.2031 - tp: 98.6000 - val_accuracy:
0.9990 - val_auc: 0.8345 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0087 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 99/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9990 - auc:
0.8846 - fn: 84.1000 - fp: 20.2200 - loss: 0.0083 - precision: 0.8288
- recall: 0.5728 - tn: 104101.2188 - tp: 106.9400 - val_accuracy:
0.9990 - val_auc: 0.8352 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0087 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
Epoch 100/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9991 - auc:
0.8884 - fn: 79.2200 - fp: 18.1600 - loss: 0.0080 - precision: 0.8380
- recall: 0.5711 - tn: 104117.4766 - tp: 97.6200 - val_accuracy:
0.9990 - val_auc: 0.8359 - val_fn: 36.0000 - val_fp: 5.0000 -
val_loss: 0.0086 - val_precision: 0.8718 - val_recall: 0.4857 -
val_tn: 42647.0000 - val_tp: 34.0000
```

Teste da RNA com os exemplos de teste

```
rna.predict(test_features[:10])
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 227ms/step
```

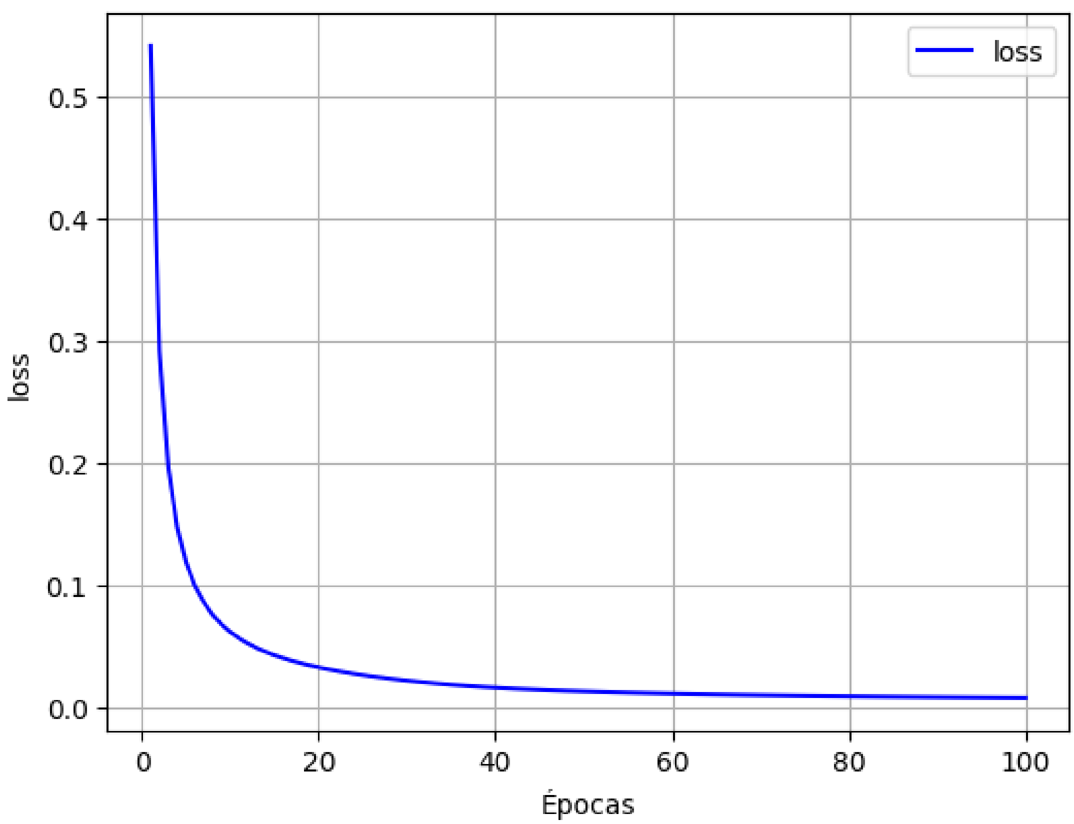
```
array([[0.00325246],
       [0.00080298],
       [0.01624356],
       [0.00324703],
       [0.00825115],
       [0.0105885 ],
       [0.00109648],
       [0.00427437],
       [0.0089893 ],
       [0.0088868 ]], dtype=float32)
```

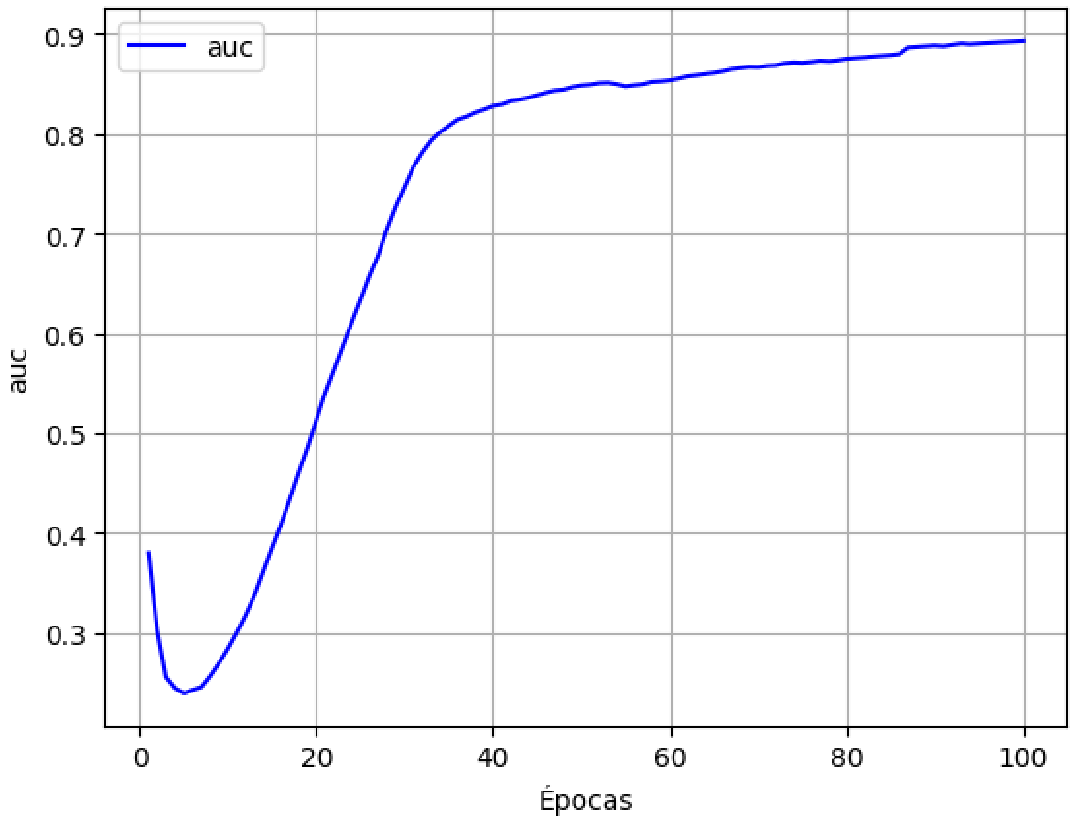
Análise dos resultados

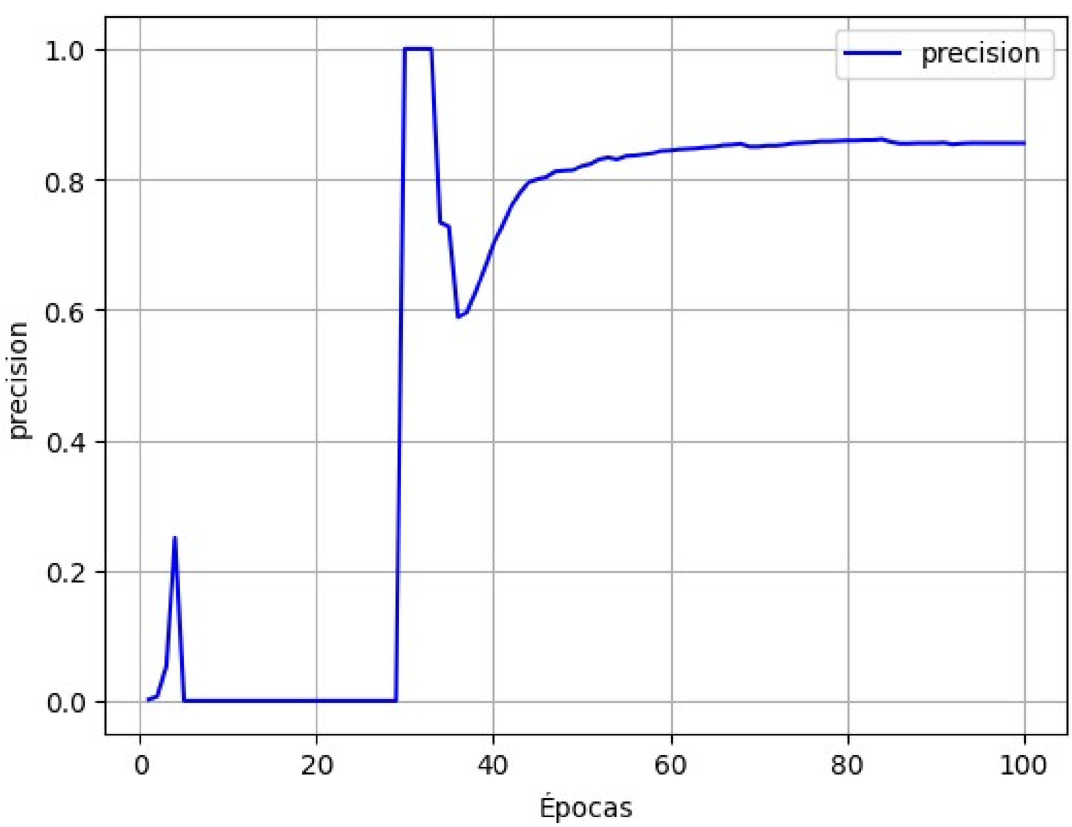
Vamos fazer os gráficos da função de custo e de algumas métricas dos resultados dos conjuntos de treinamento e validação. Eles são úteis para verificar se há "overfitting". Além disso, vamos gráficos de algumas métricas criadas.

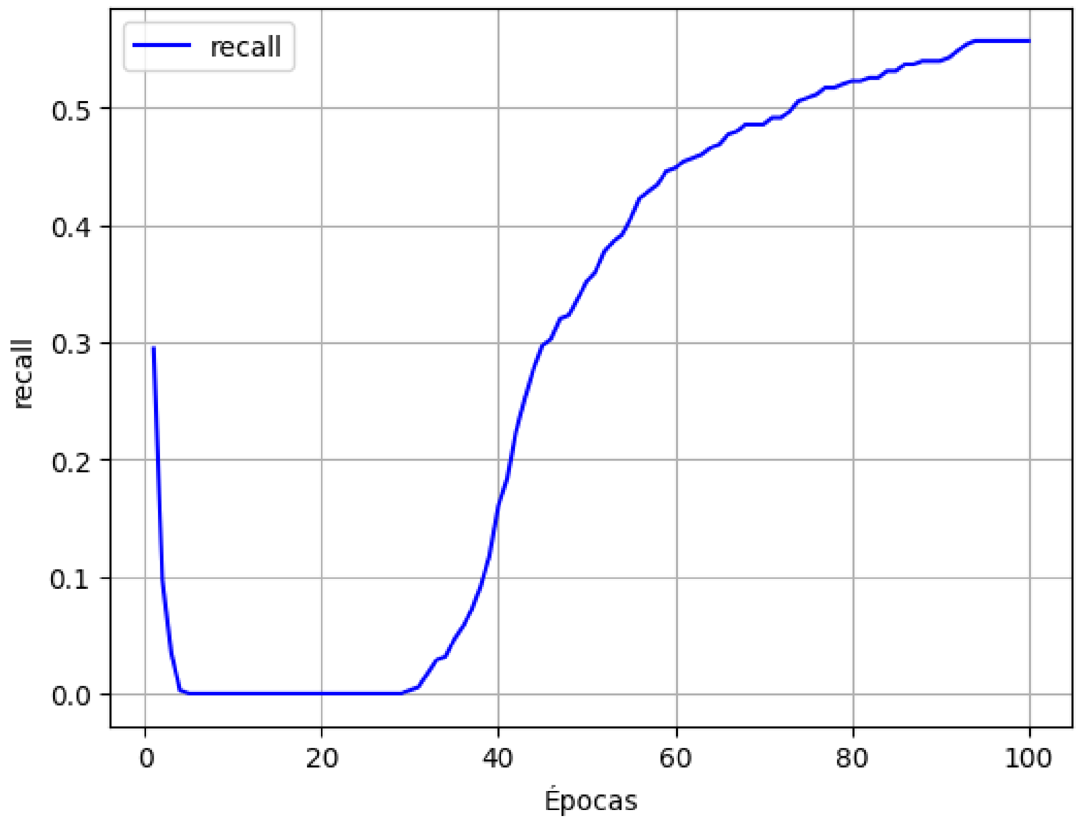
```
def plot_metrics(history, metrics):
    # Recupera hostorico do treinamento
    history_dict = history.history
```

```
    # Cria vetor de épocas
    epocas = range(1, len(history_dict['loss']) + 1)

    for name in metrics:
        plt.plot(epocas, history_dict[name], 'b', label=name)
        plt.xlabel('Épocas')
        plt.ylabel(name)
        plt.legend()
        plt.grid()
        plt.show()

# Escolhe metricas a serem mostyradas
metrics =  ['loss', 'auc', 'precision', 'recall']

# Faz gráficos
plot_metrics(history, metrics)
```




auc
0.9
0.8
0.7
0.6
0.5
0.4
0.3
auc
0
20
40
60
80
100
Épocas


precision
1.0
0.8
0.6
0.4
0.2
0.0
precision
0
20
40
60
80
100
Épocas



Avaliação dos resultados

Vamos avaliar a RN com o conjunto de dados de teste e obter os resultados para as métricas que foram utilizadas.

```
print('Número de exemplos positivos do conjunto de teste =',
len(test_labels[test_labels>0.9]))

Número de exemplos positivos do conjunto de teste = 72

base_results = rna.evaluate(test_features, test_labels,
                            batch_size=BATCH_SIZE, verbose=0)
for name, value in zip(rna.metrics_names, base_results):
    print(name, ': ', value)
print()

loss :  0.008221064694225788
compile_metrics :  38.0
```

Cálculo da Pontuação F1

```
precision = base_results[5]
recall = base_results[6]
```

```
F1 = 2*precision*recall/(precision + recall)
print('Pontuação F1 = ', F1)
```

```
Pontuação F1 =  0.9378695729538441
```

Matriz de confusão

Podemos usar uma matriz de confusão para resumir as classes reais e previstas, onde o eixo horozontal é a classe prevista e o eixo vertical é a classe real.

```
#train_pred_base = rna.predict(train_features, batch_size=BATCH_SIZE)
test_pred_base = rna.predict(test_features, batch_size=BATCH_SIZE)
print('Saídas de alguns exemplos de teste:')
print(test_pred_base[:10])
```

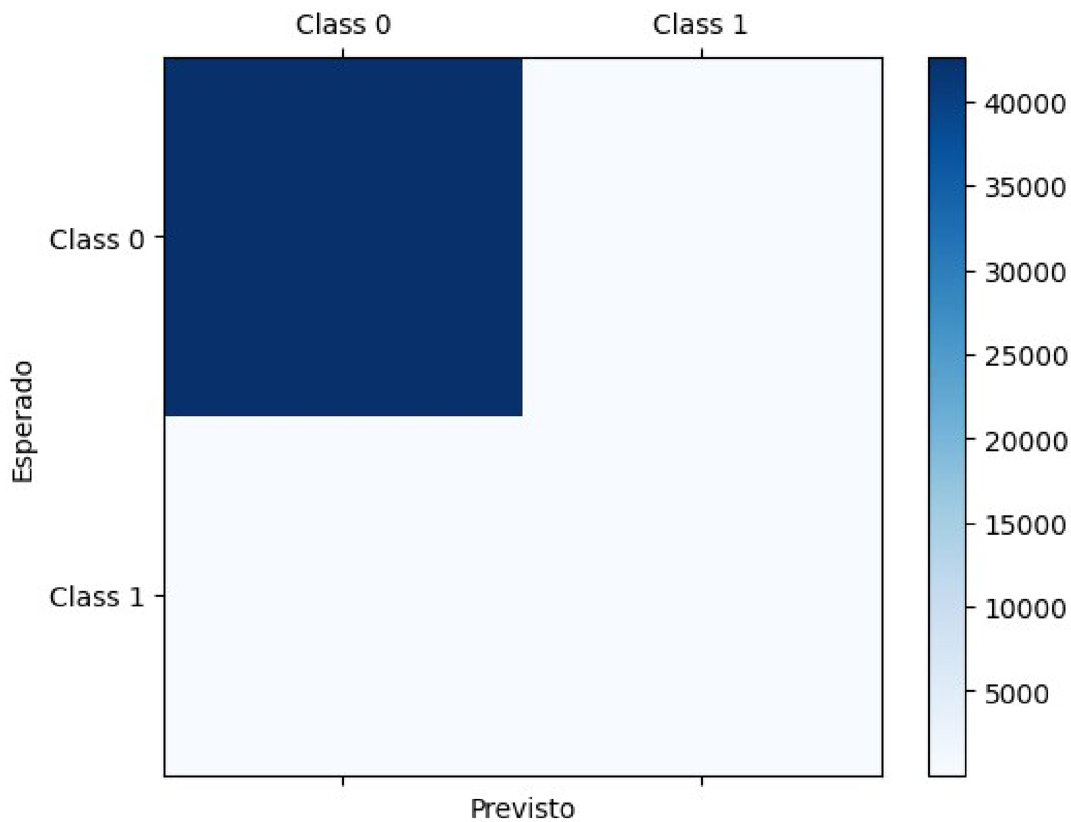
```
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 16ms/step
Saídas de alguns exemplos de teste:
[[0.00325246]
 [0.00080298]
 [0.01624356]
 [0.00324703]
 [0.00825115]
 [0.0105885 ]
 [0.00109648]
 [0.00427437]
 [0.0089893 ]
 [0.00888681]]
```

```
conf_mat = confusion_matrix(y_true=test_labels,
y_pred=np.round(test_pred_base))
print('Matriz de confusão:\n', conf_mat)

labels = ['Class 0', 'Class 1']
plt.figure(figsize=(6,6))
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(conf_mat, cmap=plt.cm.Blues)
fig.colorbar(cax)
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Previsto')
plt.ylabel('Esperado')
plt.show()
```

```
Matriz de confusão:
 [[42644     5]
 [   34    38]]
```

```
<ipython-input-22-3184717bfa2a>:10: UserWarning: FixedFormatter should
only be used together with FixedLocator
```

```
  ax.set_xticklabels([''] + labels)
<ipython-input-22-3184717bfa2a>:11: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_yticklabels([''] + labels)

<Figure size 600x600 with 0 Axes>
```



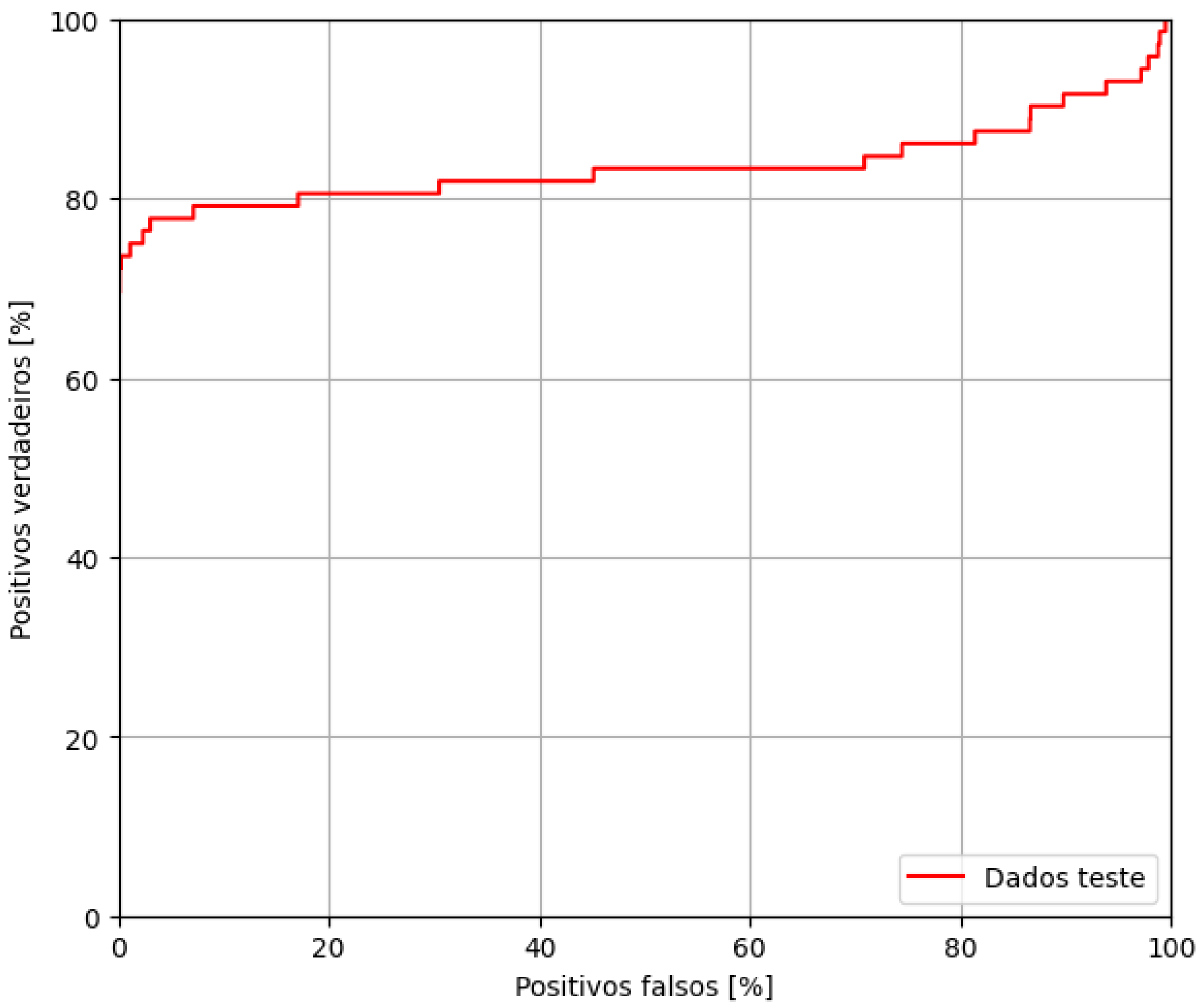
Se o modelo tivesse previsto tudo perfeitamente, a matriz de confusão seria uma matriz diagonal com os valores fora da diagonal principal, indicando as previsões incorretas, iguais a zero. Nesse caso, a matriz mostra que se tem relativamente poucos falsos positivos, o que significa que havia relativamente poucas transações legítimas que foram sinalizadas incorretamente. No entanto, seria desejado ter menos falsos negativos, apesar do custo de aumentar o número de falsos positivos. Essa troca pode ser preferível porque os falsos negativos permitiriam a realização de transações fraudulentas, ao passo que os falsos positivos podem fazer com que um e-mail seja enviado a um cliente solicitando a verificação da atividade do cartão.

Gráfico do ROC

Agora vamos fazer o gráfico do ROC. Este gráfico é útil porque mostra como varia o desempenho da RN em função do valor do limiar para considerar as classes dos exemplos.

```
#fp_train, tp_train, limiar = sklearn.metrics.roc_curve(train_labels, train_pred_base)
```

```
fp_test, tp_test, limiar = sklearn.metrics.roc_curve(test_labels,
test_pred_base)
plt.figure(figsize=(7, 6))
#plt.plot(100*fp_train, 100*tp_train, 'b', label='Dados treinamento')
plt.plot(100*fp_test, 100*tp_test, 'r', label='Dados teste')
plt.xlabel('Positivos falsos [%]')
plt.ylabel('Positivos verdadeiros [%]')
plt.xlim([0,100])
plt.ylim([0,100])
plt.grid(True)
ax = plt.gca()
plt.legend(loc='lower right')
plt.show()
```




```
limiar[:10]
```

```
array([       inf, 0.996437  , 0.97462815, 0.963465  , 0.9631998 ,
       0.96264446, 0.8854667 , 0.8821007 , 0.49927828, 0.47054875],
      dtype=float32)
```

Observa-se que a precisão é relativamente alta, mas a revocação e a área sob a curva ROC (AUC) não são tão altas quanto se desejaria. Os classificadores geralmente enfrentam desafios ao tentar maximizar a precisão e a revocação, o que é especialmente verdadeiro quando se trabalha com conjuntos de dados desbalanceados.

É importante considerar os custos dos diferentes tipos de erros no contexto do problem. Neste exemplo, um falso negativo (uma transação fraudulenta é perdida) pode ter um custo financeiro, enquanto um falso positivo (uma transação é sinalizada incorretamente como fraudulenta) pode prejudicar o relacionamento com o cliente.

6. Treinamento com pesos para cada classe

Calculo dos pesos das classes

O objetivo é identificar transações fraudulentas, mas não se tem muitas desses exemplos positivos para trabalhar, então, uma solução é o classificador dar um peso maior para os poucos exemplos dessa classe que estão disponíveis. Pode-se fazer isso passando pesos para cada classe por meio de um parâmetro. Isso faz com que o modelo "preste mais atenção" aos exemplos de uma classe sub-representada.

```
# Cálculo dos pesos das duas classe
weight_for_0 = (1 / neg)*(total)/2.0
weight_for_1 = (1 / pos)*(total)/2.0

# Dicionário de pesos das classes para treinamento
class_weights = {0: weight_for_0, 1: weight_for_1}

print('Peso da classe 0: {:.2f}'.format(weight_for_0))
print('Peso da classe 1: {:.2f}'.format(weight_for_1))

Peso da classe 0: 0.50
Peso da classe 1: 289.44
```

Treinamento da RNA com pesos de classe

Vamos treinar novamente a RN com pesos diferentes para cada classe e avaliar como isso afeta as previsões.

Importante:

- Usar `class_weights` muda o valor da função de custo. Isso pode afetar a estabilidade do treinamento dependendo do otimizador. Alguns otimizadores, tal como, momento e RMSProp podem falhar.
- O otimizador usado é Adam, que é pouco afetado pela mudança de escala da função de custo.

- Observe que devido à ponderação, as perdas totais não são comparáveis entre os dois modelos.

```
rna_pond = make_model(METRICS, features_shape)

pond_history = rna_pond.fit(train_features, train_labels,
batch_size=BATCH_SIZE,
                   epochs=100,
                   #callbacks = [early_stopping],
                   validation_data=(val_features, val_labels),
                   # Pesos das classes
                   class_weight=class_weights)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/layers/core/
dense.py:87: UserWarning: Do not pass an `input_shape`/`input_dim`
argument to a layer. When using Sequential models, prefer using an
`Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer,
**kwargs)

Epoch 1/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 7s 99ms/step - accuracy: 0.7868 - auc:
0.8108 - fn: 71.9800 - fp: 34491.1406 - loss: 0.6393 - precision:
0.0064 - recall: 0.6817 - tn: 112290.7188 - tp: 179.6400 -
val_accuracy: 0.7132 - val_auc: 0.8601 - val_fn: 13.0000 - val_fp:
12241.0000 - val_loss: 0.5980 - val_precision: 0.0046 - val_recall:
0.8143 - val_tn: 30411.0000 - val_tp: 57.0000
Epoch 2/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.7388 - auc:
0.9132 - fn: 21.8600 - fp: 26339.9805 - loss: 0.3933 - precision:
0.0059 - recall: 0.8809 - tn: 77791.4766 - tp: 159.1600 -
val_accuracy: 0.8094 - val_auc: 0.8863 - val_fn: 13.0000 - val_fp:
8129.0000 - val_loss: 0.4989 - val_precision: 0.0070 - val_recall:
0.8143 - val_tn: 34523.0000 - val_tp: 57.0000
Epoch 3/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8294 - auc:
0.9475 - fn: 22.2200 - fp: 17288.9199 - loss: 0.3262 - precision:
0.0095 - recall: 0.8960 - tn: 86832.0625 - tp: 169.2800 -
val_accuracy: 0.8738 - val_auc: 0.9048 - val_fn: 11.0000 - val_fp:
5381.0000 - val_loss: 0.4263 - val_precision: 0.0108 - val_recall:
0.8429 - val_tn: 37271.0000 - val_tp: 59.0000
Epoch 4/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.8845 - auc:
0.9534 - fn: 20.9000 - fp: 11670.7803 - loss: 0.3036 - precision:
0.0156 - recall: 0.9024 - tn: 92444.6562 - tp: 176.1400 -
val_accuracy: 0.9100 - val_auc: 0.9170 - val_fn: 10.0000 - val_fp:
3833.0000 - val_loss: 0.3757 - val_precision: 0.0154 - val_recall:
0.8571 - val_tn: 38819.0000 - val_tp: 60.0000
Epoch 5/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9188 - auc:
0.9508 - fn: 21.3600 - fp: 8123.8198 - loss: 0.2802 - precision:
```

```
0.0189 - recall: 0.8800 - tn: 96002.5000 - tp: 164.8000 -
val_accuracy: 0.9344 - val_auc: 0.9235 - val_fn: 10.0000 - val_fp:
2791.0000 - val_loss: 0.3389 - val_precision: 0.0210 - val_recall:
0.8571 - val_tn: 39861.0000 - val_tp: 60.0000
Epoch 6/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9429 - auc:
0.9584 - fn: 23.6200 - fp: 5807.5400 - loss: 0.2572 - precision:
0.0264 - recall: 0.8707 - tn: 98320.2969 - tp: 161.0200 -
val_accuracy: 0.9496 - val_auc: 0.9268 - val_fn: 11.0000 - val_fp:
2143.0000 - val_loss: 0.3089 - val_precision: 0.0268 - val_recall:
0.8429 - val_tn: 40509.0000 - val_tp: 59.0000
Epoch 7/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9556 - auc:
0.9701 - fn: 22.1800 - fp: 4518.7202 - loss: 0.2358 - precision:
0.0349 - recall: 0.8862 - tn: 99605.5391 - tp: 166.0400 -
val_accuracy: 0.9589 - val_auc: 0.9290 - val_fn: 11.0000 - val_fp:
1743.0000 - val_loss: 0.2855 - val_precision: 0.0327 - val_recall:
0.8429 - val_tn: 40909.0000 - val_tp: 59.0000
Epoch 8/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9645 - auc:
0.9660 - fn: 23.3400 - fp: 3626.1799 - loss: 0.2327 - precision:
0.0439 - recall: 0.8828 - tn: 100497.0625 - tp: 165.9000 -
val_accuracy: 0.9651 - val_auc: 0.9302 - val_fn: 10.0000 - val_fp:
1480.0000 - val_loss: 0.2664 - val_precision: 0.0390 - val_recall:
0.8571 - val_tn: 41172.0000 - val_tp: 60.0000
Epoch 9/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9711 - auc:
0.9731 - fn: 19.8800 - fp: 2922.6001 - loss: 0.2069 - precision:
0.0490 - recall: 0.8926 - tn: 101212.9219 - tp: 157.0800 -
val_accuracy: 0.9691 - val_auc: 0.9309 - val_fn: 10.0000 - val_fp:
1311.0000 - val_loss: 0.2519 - val_precision: 0.0438 - val_recall:
0.8571 - val_tn: 41341.0000 - val_tp: 60.0000
Epoch 10/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9730 - auc:
0.9722 - fn: 20.5800 - fp: 2758.4399 - loss: 0.2029 - precision:
0.0591 - recall: 0.9025 - tn: 101365.2812 - tp: 168.1800 -
val_accuracy: 0.9724 - val_auc: 0.9326 - val_fn: 10.0000 - val_fp:
1168.0000 - val_loss: 0.2374 - val_precision: 0.0489 - val_recall:
0.8571 - val_tn: 41484.0000 - val_tp: 60.0000
Epoch 11/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9747 - auc:
0.9739 - fn: 18.2400 - fp: 2597.9800 - loss: 0.1980 - precision:
0.0648 - recall: 0.9108 - tn: 101523.2422 - tp: 173.0200 -
val_accuracy: 0.9748 - val_auc: 0.9329 - val_fn: 10.0000 - val_fp:
1065.0000 - val_loss: 0.2252 - val_precision: 0.0533 - val_recall:
0.8571 - val_tn: 41587.0000 - val_tp: 60.0000
Epoch 12/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9767 - auc:
0.9762 - fn: 21.5800 - fp: 2421.0601 - loss: 0.1938 - precision:
```

```
0.0608 - recall: 0.8761 - tn: 101710.8984 - tp: 158.9400 -
val_accuracy: 0.9760 - val_auc: 0.9338 - val_fn: 10.0000 - val_fp:
1015.0000 - val_loss: 0.2159 - val_precision: 0.0558 - val_recall:
0.8571 - val_tn: 41637.0000 - val_tp: 60.0000
Epoch 13/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9785 - auc:
0.9848 - fn: 17.4200 - fp: 2215.0801 - loss: 0.1673 - precision:
0.0674 - recall: 0.9109 - tn: 101917.0781 - tp: 162.9000 -
val_accuracy: 0.9770 - val_auc: 0.9342 - val_fn: 10.0000 - val_fp:
974.0000 - val_loss: 0.2065 - val_precision: 0.0580 - val_recall:
0.8571 - val_tn: 41678.0000 - val_tp: 60.0000
Epoch 14/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9791 - auc:
0.9827 - fn: 19.1000 - fp: 2157.4399 - loss: 0.1715 - precision:
0.0701 - recall: 0.9031 - tn: 101973.4375 - tp: 162.5000 -
val_accuracy: 0.9778 - val_auc: 0.9348 - val_fn: 10.0000 - val_fp:
938.0000 - val_loss: 0.1984 - val_precision: 0.0601 - val_recall:
0.8571 - val_tn: 41714.0000 - val_tp: 60.0000
Epoch 15/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9796 - auc:
0.9807 - fn: 16.2600 - fp: 2081.5801 - loss: 0.1693 - precision:
0.0732 - recall: 0.9109 - tn: 102048.2031 - tp: 166.4400 -
val_accuracy: 0.9784 - val_auc: 0.9356 - val_fn: 10.0000 - val_fp:
911.0000 - val_loss: 0.1912 - val_precision: 0.0618 - val_recall:
0.8571 - val_tn: 41741.0000 - val_tp: 60.0000
Epoch 16/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9798 - auc:
0.9738 - fn: 20.0200 - fp: 2064.6001 - loss: 0.1805 - precision:
0.0677 - recall: 0.8814 - tn: 102070.3594 - tp: 157.5000 -
val_accuracy: 0.9793 - val_auc: 0.9361 - val_fn: 10.0000 - val_fp:
873.0000 - val_loss: 0.1841 - val_precision: 0.0643 - val_recall:
0.8571 - val_tn: 41779.0000 - val_tp: 60.0000
Epoch 17/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9804 - auc:
0.9746 - fn: 19.5200 - fp: 2036.6600 - loss: 0.1961 - precision:
0.0802 - recall: 0.8887 - tn: 102084.4766 - tp: 171.8200 -
val_accuracy: 0.9800 - val_auc: 0.9364 - val_fn: 10.0000 - val_fp:
845.0000 - val_loss: 0.1770 - val_precision: 0.0663 - val_recall:
0.8571 - val_tn: 41807.0000 - val_tp: 60.0000
Epoch 18/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9815 - auc:
0.9744 - fn: 18.3800 - fp: 1910.6400 - loss: 0.1807 - precision:
0.0758 - recall: 0.8897 - tn: 102223.1016 - tp: 160.3600 -
val_accuracy: 0.9803 - val_auc: 0.9372 - val_fn: 11.0000 - val_fp:
830.0000 - val_loss: 0.1713 - val_precision: 0.0664 - val_recall:
0.8429 - val_tn: 41822.0000 - val_tp: 59.0000
Epoch 19/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9817 - auc:
0.9769 - fn: 18.4200 - fp: 1908.0200 - loss: 0.1828 - precision:
```

```
0.0832 - recall: 0.8893 - tn: 102219.4219 - tp: 166.6200 -
val_accuracy: 0.9807 - val_auc: 0.9373 - val_fn: 11.0000 - val_fp:
814.0000 - val_loss: 0.1663 - val_precision: 0.0676 - val_recall:
0.8429 - val_tn: 41838.0000 - val_tp: 59.0000
Epoch 20/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9818 - auc:
0.9818 - fn: 19.4400 - fp: 1890.2200 - loss: 0.1648 - precision:
0.0800 - recall: 0.8968 - tn: 102238.5234 - tp: 164.3000 -
val_accuracy: 0.9808 - val_auc: 0.9384 - val_fn: 11.0000 - val_fp:
810.0000 - val_loss: 0.1618 - val_precision: 0.0679 - val_recall:
0.8429 - val_tn: 41842.0000 - val_tp: 59.0000
Epoch 21/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9825 - auc:
0.9825 - fn: 17.0000 - fp: 1814.3000 - loss: 0.1532 - precision:
0.0871 - recall: 0.9211 - tn: 102309.6016 - tp: 171.5800 -
val_accuracy: 0.9810 - val_auc: 0.9384 - val_fn: 11.0000 - val_fp:
801.0000 - val_loss: 0.1580 - val_precision: 0.0686 - val_recall:
0.8429 - val_tn: 41851.0000 - val_tp: 59.0000
Epoch 22/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9826 - auc:
0.9866 - fn: 17.4200 - fp: 1798.3800 - loss: 0.1486 - precision:
0.0834 - recall: 0.9060 - tn: 102330.6562 - tp: 166.0200 -
val_accuracy: 0.9811 - val_auc: 0.9386 - val_fn: 11.0000 - val_fp:
796.0000 - val_loss: 0.1546 - val_precision: 0.0690 - val_recall:
0.8429 - val_tn: 41856.0000 - val_tp: 59.0000
Epoch 23/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9820 - auc:
0.9857 - fn: 17.0200 - fp: 1818.8800 - loss: 0.1459 - precision:
0.0854 - recall: 0.9172 - tn: 102308.0781 - tp: 168.5000 -
val_accuracy: 0.9812 - val_auc: 0.9397 - val_fn: 11.0000 - val_fp:
794.0000 - val_loss: 0.1513 - val_precision: 0.0692 - val_recall:
0.8429 - val_tn: 41858.0000 - val_tp: 59.0000
Epoch 24/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9824 - auc:
0.9845 - fn: 16.6000 - fp: 1804.6400 - loss: 0.1508 - precision:
0.0890 - recall: 0.9156 - tn: 102320.9766 - tp: 170.2600 -
val_accuracy: 0.9814 - val_auc: 0.9395 - val_fn: 11.0000 - val_fp:
783.0000 - val_loss: 0.1480 - val_precision: 0.0701 - val_recall:
0.8429 - val_tn: 41869.0000 - val_tp: 59.0000
Epoch 25/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9830 - auc:
0.9802 - fn: 17.4800 - fp: 1756.8600 - loss: 0.1537 - precision:
0.0807 - recall: 0.8982 - tn: 102378.5781 - tp: 159.5600 -
val_accuracy: 0.9813 - val_auc: 0.9403 - val_fn: 11.0000 - val_fp:
786.0000 - val_loss: 0.1455 - val_precision: 0.0698 - val_recall:
0.8429 - val_tn: 41866.0000 - val_tp: 59.0000
Epoch 26/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9829 - auc:
0.9876 - fn: 15.8000 - fp: 1770.0601 - loss: 0.1379 - precision:
```

```
0.0812 - recall: 0.9128 - tn: 102365.8594 - tp: 160.7600 -
val_accuracy: 0.9815 - val_auc: 0.9408 - val_fn: 11.0000 - val_fp:
780.0000 - val_loss: 0.1426 - val_precision: 0.0703 - val_recall:
0.8429 - val_tn: 41872.0000 - val_tp: 59.0000
Epoch 27/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9827 - auc:
0.9860 - fn: 17.9400 - fp: 1798.7000 - loss: 0.1506 - precision:
0.0848 - recall: 0.9013 - tn: 102327.1406 - tp: 168.7000 -
val_accuracy: 0.9816 - val_auc: 0.9407 - val_fn: 11.0000 - val_fp:
775.0000 - val_loss: 0.1394 - val_precision: 0.0707 - val_recall:
0.8429 - val_tn: 41877.0000 - val_tp: 59.0000
Epoch 28/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9833 - auc:
0.9817 - fn: 16.5800 - fp: 1733.2000 - loss: 0.1550 - precision:
0.0875 - recall: 0.8980 - tn: 102397.0391 - tp: 165.6600 -
val_accuracy: 0.9817 - val_auc: 0.9410 - val_fn: 11.0000 - val_fp:
769.0000 - val_loss: 0.1374 - val_precision: 0.0713 - val_recall:
0.8429 - val_tn: 41883.0000 - val_tp: 59.0000
Epoch 29/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9824 - auc:
0.9785 - fn: 18.7600 - fp: 1802.4800 - loss: 0.1715 - precision:
0.0868 - recall: 0.8912 - tn: 102323.3984 - tp: 167.8400 -
val_accuracy: 0.9819 - val_auc: 0.9418 - val_fn: 11.0000 - val_fp:
762.0000 - val_loss: 0.1348 - val_precision: 0.0719 - val_recall:
0.8429 - val_tn: 41890.0000 - val_tp: 59.0000
Epoch 30/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9834 - auc:
0.9866 - fn: 13.6800 - fp: 1698.6400 - loss: 0.1317 - precision:
0.0865 - recall: 0.9278 - tn: 102435.8203 - tp: 164.3400 -
val_accuracy: 0.9817 - val_auc: 0.9416 - val_fn: 11.0000 - val_fp:
769.0000 - val_loss: 0.1331 - val_precision: 0.0713 - val_recall:
0.8429 - val_tn: 41883.0000 - val_tp: 59.0000
Epoch 31/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9831 - auc:
0.9840 - fn: 17.3600 - fp: 1762.6200 - loss: 0.1516 - precision:
0.0910 - recall: 0.9063 - tn: 102363.7969 - tp: 168.7000 -
val_accuracy: 0.9820 - val_auc: 0.9418 - val_fn: 11.0000 - val_fp:
760.0000 - val_loss: 0.1308 - val_precision: 0.0720 - val_recall:
0.8429 - val_tn: 41892.0000 - val_tp: 59.0000
Epoch 32/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9836 - auc:
0.9869 - fn: 15.1800 - fp: 1698.4399 - loss: 0.1317 - precision:
0.0833 - recall: 0.9106 - tn: 102439.2031 - tp: 159.6600 -
val_accuracy: 0.9818 - val_auc: 0.9426 - val_fn: 11.0000 - val_fp:
765.0000 - val_loss: 0.1295 - val_precision: 0.0716 - val_recall:
0.8429 - val_tn: 41887.0000 - val_tp: 59.0000
Epoch 33/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9836 - auc:
0.9853 - fn: 14.1600 - fp: 1739.7200 - loss: 0.1413 - precision:
```

```
0.0976 - recall: 0.9301 - tn: 102382.9766 - tp: 175.6200 -
val_accuracy: 0.9821 - val_auc: 0.9428 - val_fn: 11.0000 - val_fp:
753.0000 - val_loss: 0.1275 - val_precision: 0.0727 - val_recall:
0.8429 - val_tn: 41899.0000 - val_tp: 59.0000
Epoch 34/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9834 - auc:
0.9846 - fn: 14.5600 - fp: 1714.1000 - loss: 0.1314 - precision:
0.0829 - recall: 0.9227 - tn: 102424.1406 - tp: 159.6800 -
val_accuracy: 0.9820 - val_auc: 0.9428 - val_fn: 11.0000 - val_fp:
756.0000 - val_loss: 0.1259 - val_precision: 0.0724 - val_recall:
0.8429 - val_tn: 41896.0000 - val_tp: 59.0000
Epoch 35/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9831 - auc:
0.9897 - fn: 16.1400 - fp: 1759.3600 - loss: 0.1361 - precision:
0.0863 - recall: 0.9156 - tn: 102369.0781 - tp: 167.9000 -
val_accuracy: 0.9823 - val_auc: 0.9434 - val_fn: 11.0000 - val_fp:
744.0000 - val_loss: 0.1238 - val_precision: 0.0735 - val_recall:
0.8429 - val_tn: 41908.0000 - val_tp: 59.0000
Epoch 36/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9831 - auc:
0.9879 - fn: 15.9600 - fp: 1753.5800 - loss: 0.1414 - precision:
0.0932 - recall: 0.9215 - tn: 102367.4375 - tp: 175.5000 -
val_accuracy: 0.9824 - val_auc: 0.9429 - val_fn: 11.0000 - val_fp:
739.0000 - val_loss: 0.1218 - val_precision: 0.0739 - val_recall:
0.8429 - val_tn: 41913.0000 - val_tp: 59.0000
Epoch 37/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9835 - auc:
0.9896 - fn: 13.6000 - fp: 1673.0200 - loss: 0.1329 - precision:
0.0944 - recall: 0.9216 - tn: 102452.5234 - tp: 173.3400 -
val_accuracy: 0.9824 - val_auc: 0.9433 - val_fn: 11.0000 - val_fp:
743.0000 - val_loss: 0.1208 - val_precision: 0.0736 - val_recall:
0.8429 - val_tn: 41909.0000 - val_tp: 59.0000
Epoch 38/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9834 - auc:
0.9844 - fn: 16.4400 - fp: 1732.3600 - loss: 0.1489 - precision:
0.0892 - recall: 0.9052 - tn: 102398.8984 - tp: 164.7800 -
val_accuracy: 0.9826 - val_auc: 0.9438 - val_fn: 11.0000 - val_fp:
734.0000 - val_loss: 0.1190 - val_precision: 0.0744 - val_recall:
0.8429 - val_tn: 41918.0000 - val_tp: 59.0000
Epoch 39/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 7ms/step - accuracy: 0.9831 - auc:
0.9781 - fn: 16.3600 - fp: 1738.3400 - loss: 0.1632 - precision:
0.0833 - recall: 0.8939 - tn: 102395.4219 - tp: 162.3600 -
val_accuracy: 0.9827 - val_auc: 0.9438 - val_fn: 11.0000 - val_fp:
729.0000 - val_loss: 0.1176 - val_precision: 0.0749 - val_recall:
0.8429 - val_tn: 41923.0000 - val_tp: 59.0000
Epoch 40/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9836 - auc:
0.9920 - fn: 13.1600 - fp: 1686.7600 - loss: 0.1154 - precision:
```

```
0.0959 - recall: 0.9461 - tn: 102435.7188 - tp: 176.8400 -
val_accuracy: 0.9825 - val_auc: 0.9439 - val_fn: 11.0000 - val_fp:
736.0000 - val_loss: 0.1165 - val_precision: 0.0742 - val_recall:
0.8429 - val_tn: 41916.0000 - val_tp: 59.0000
Epoch 41/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9840 - auc:
0.9865 - fn: 13.9800 - fp: 1654.4399 - loss: 0.1312 - precision:
0.0907 - recall: 0.9258 - tn: 102477.7188 - tp: 166.3400 -
val_accuracy: 0.9825 - val_auc: 0.9443 - val_fn: 11.0000 - val_fp:
735.0000 - val_loss: 0.1154 - val_precision: 0.0743 - val_recall:
0.8429 - val_tn: 41917.0000 - val_tp: 59.0000
Epoch 42/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9838 - auc:
0.9892 - fn: 13.0400 - fp: 1670.9200 - loss: 0.1222 - precision:
0.0983 - recall: 0.9386 - tn: 102451.0625 - tp: 177.4600 -
val_accuracy: 0.9824 - val_auc: 0.9447 - val_fn: 11.0000 - val_fp:
739.0000 - val_loss: 0.1143 - val_precision: 0.0739 - val_recall:
0.8429 - val_tn: 41913.0000 - val_tp: 59.0000
Epoch 43/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9840 - auc:
0.9871 - fn: 14.9400 - fp: 1668.6000 - loss: 0.1401 - precision:
0.0924 - recall: 0.9126 - tn: 102460.0234 - tp: 168.9200 -
val_accuracy: 0.9825 - val_auc: 0.9444 - val_fn: 11.0000 - val_fp:
736.0000 - val_loss: 0.1131 - val_precision: 0.0742 - val_recall:
0.8429 - val_tn: 41916.0000 - val_tp: 59.0000
Epoch 44/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9841 - auc:
0.9880 - fn: 14.1800 - fp: 1658.2800 - loss: 0.1274 - precision:
0.1041 - recall: 0.9370 - tn: 102460.1016 - tp: 179.9200 -
val_accuracy: 0.9825 - val_auc: 0.9446 - val_fn: 11.0000 - val_fp:
737.0000 - val_loss: 0.1120 - val_precision: 0.0741 - val_recall:
0.8429 - val_tn: 41915.0000 - val_tp: 59.0000
Epoch 45/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9841 - auc:
0.9869 - fn: 13.9200 - fp: 1655.6400 - loss: 0.1288 - precision:
0.0906 - recall: 0.9243 - tn: 102473.3828 - tp: 169.5400 -
val_accuracy: 0.9824 - val_auc: 0.9450 - val_fn: 11.0000 - val_fp:
741.0000 - val_loss: 0.1110 - val_precision: 0.0737 - val_recall:
0.8429 - val_tn: 41911.0000 - val_tp: 59.0000
Epoch 46/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9836 - auc:
0.9913 - fn: 14.4800 - fp: 1701.9800 - loss: 0.1270 - precision:
0.0946 - recall: 0.9264 - tn: 102422.5781 - tp: 173.4400 -
val_accuracy: 0.9825 - val_auc: 0.9452 - val_fn: 11.0000 - val_fp:
736.0000 - val_loss: 0.1101 - val_precision: 0.0742 - val_recall:
0.8429 - val_tn: 41916.0000 - val_tp: 59.0000
Epoch 47/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9836 - auc:
0.9907 - fn: 14.0800 - fp: 1684.2400 - loss: 0.1259 - precision:
```

```
0.0919 - recall: 0.9140 - tn: 102442.8438 - tp: 171.3200 -
val_accuracy: 0.9825 - val_auc: 0.9456 - val_fn: 11.0000 - val_fp:
738.0000 - val_loss: 0.1095 - val_precision: 0.0740 - val_recall:
0.8429 - val_tn: 41914.0000 - val_tp: 59.0000
Epoch 48/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9835 - auc:
0.9901 - fn: 14.4200 - fp: 1695.8199 - loss: 0.1277 - precision:
0.0866 - recall: 0.9103 - tn: 102441.7031 - tp: 160.5400 -
val_accuracy: 0.9825 - val_auc: 0.9462 - val_fn: 11.0000 - val_fp:
735.0000 - val_loss: 0.1088 - val_precision: 0.0743 - val_recall:
0.8429 - val_tn: 41917.0000 - val_tp: 59.0000
Epoch 49/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9839 - auc:
0.9878 - fn: 14.0000 - fp: 1649.6200 - loss: 0.1289 - precision:
0.0909 - recall: 0.9192 - tn: 102485.1797 - tp: 163.6800 -
val_accuracy: 0.9825 - val_auc: 0.9459 - val_fn: 11.0000 - val_fp:
736.0000 - val_loss: 0.1078 - val_precision: 0.0742 - val_recall:
0.8429 - val_tn: 41916.0000 - val_tp: 59.0000
Epoch 50/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9842 - auc:
0.9952 - fn: 10.8600 - fp: 1626.6801 - loss: 0.0986 - precision:
0.1038 - recall: 0.9523 - tn: 102496.2422 - tp: 178.7000 -
val_accuracy: 0.9824 - val_auc: 0.9454 - val_fn: 11.0000 - val_fp:
742.0000 - val_loss: 0.1074 - val_precision: 0.0737 - val_recall:
0.8429 - val_tn: 41910.0000 - val_tp: 59.0000
Epoch 51/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9840 - auc:
0.9908 - fn: 13.4200 - fp: 1657.0800 - loss: 0.1168 - precision:
0.0862 - recall: 0.9264 - tn: 102478.7969 - tp: 163.1800 -
val_accuracy: 0.9824 - val_auc: 0.9458 - val_fn: 11.0000 - val_fp:
742.0000 - val_loss: 0.1068 - val_precision: 0.0737 - val_recall:
0.8429 - val_tn: 41910.0000 - val_tp: 59.0000
Epoch 52/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9840 - auc:
0.9940 - fn: 12.0800 - fp: 1656.9000 - loss: 0.1064 - precision:
0.0922 - recall: 0.9344 - tn: 102478.0000 - tp: 165.5000 -
val_accuracy: 0.9822 - val_auc: 0.9461 - val_fn: 11.0000 - val_fp:
750.0000 - val_loss: 0.1063 - val_precision: 0.0729 - val_recall:
0.8429 - val_tn: 41902.0000 - val_tp: 59.0000
Epoch 53/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9834 - auc:
0.9926 - fn: 12.6600 - fp: 1713.6000 - loss: 0.1161 - precision:
0.0837 - recall: 0.9260 - tn: 102422.2031 - tp: 164.0200 -
val_accuracy: 0.9825 - val_auc: 0.9455 - val_fn: 11.0000 - val_fp:
736.0000 - val_loss: 0.1050 - val_precision: 0.0742 - val_recall:
0.8429 - val_tn: 41916.0000 - val_tp: 59.0000
Epoch 54/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9835 - auc:
0.9856 - fn: 15.0000 - fp: 1694.2800 - loss: 0.1482 - precision:
```

```
0.0932 - recall: 0.9050 - tn: 102427.3594 - tp: 175.8400 -
val_accuracy: 0.9827 - val_auc: 0.9460 - val_fn: 11.0000 - val_fp:
727.0000 - val_loss: 0.1039 - val_precision: 0.0751 - val_recall:
0.8429 - val_tn: 41925.0000 - val_tp: 59.0000
Epoch 55/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9842 - auc:
0.9885 - fn: 14.2800 - fp: 1638.3400 - loss: 0.1286 - precision:
0.0894 - recall: 0.9087 - tn: 102493.1016 - tp: 166.7600 -
val_accuracy: 0.9827 - val_auc: 0.9457 - val_fn: 11.0000 - val_fp:
726.0000 - val_loss: 0.1029 - val_precision: 0.0752 - val_recall:
0.8429 - val_tn: 41926.0000 - val_tp: 59.0000
Epoch 56/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9848 - auc:
0.9907 - fn: 12.5800 - fp: 1589.1801 - loss: 0.1059 - precision:
0.0907 - recall: 0.9420 - tn: 102546.1797 - tp: 164.5400 -
val_accuracy: 0.9827 - val_auc: 0.9459 - val_fn: 11.0000 - val_fp:
728.0000 - val_loss: 0.1023 - val_precision: 0.0750 - val_recall:
0.8429 - val_tn: 41924.0000 - val_tp: 59.0000
Epoch 57/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9845 - auc:
0.9945 - fn: 10.8400 - fp: 1612.1600 - loss: 0.1011 - precision:
0.0981 - recall: 0.9508 - tn: 102511.4375 - tp: 178.0400 -
val_accuracy: 0.9827 - val_auc: 0.9464 - val_fn: 11.0000 - val_fp:
730.0000 - val_loss: 0.1018 - val_precision: 0.0748 - val_recall:
0.8429 - val_tn: 41922.0000 - val_tp: 59.0000
Epoch 58/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9843 - auc:
0.9903 - fn: 12.3800 - fp: 1616.2000 - loss: 0.1128 - precision:
0.0937 - recall: 0.9343 - tn: 102519.8984 - tp: 164.0000 -
val_accuracy: 0.9827 - val_auc: 0.9466 - val_fn: 11.0000 - val_fp:
728.0000 - val_loss: 0.1012 - val_precision: 0.0750 - val_recall:
0.8429 - val_tn: 41924.0000 - val_tp: 59.0000
Epoch 59/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9841 - auc:
0.9881 - fn: 14.0600 - fp: 1652.5400 - loss: 0.1319 - precision:
0.0937 - recall: 0.9166 - tn: 102472.4609 - tp: 173.4200 -
val_accuracy: 0.9828 - val_auc: 0.9467 - val_fn: 11.0000 - val_fp:
724.0000 - val_loss: 0.1005 - val_precision: 0.0754 - val_recall:
0.8429 - val_tn: 41928.0000 - val_tp: 59.0000
Epoch 60/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9843 - auc:
0.9929 - fn: 12.5000 - fp: 1632.2400 - loss: 0.1155 - precision:
0.0954 - recall: 0.9320 - tn: 102499.8828 - tp: 167.8600 -
val_accuracy: 0.9828 - val_auc: 0.9472 - val_fn: 11.0000 - val_fp:
724.0000 - val_loss: 0.0998 - val_precision: 0.0754 - val_recall:
0.8429 - val_tn: 41928.0000 - val_tp: 59.0000
Epoch 61/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9849 - auc:
0.9926 - fn: 12.9000 - fp: 1588.4399 - loss: 0.1085 - precision:
```

```
0.1007 - recall: 0.9384 - tn: 102535.9375 - tp: 175.2000 -
val_accuracy: 0.9829 - val_auc: 0.9475 - val_fn: 11.0000 - val_fp:
721.0000 - val_loss: 0.0991 - val_precision: 0.0756 - val_recall:
0.8429 - val_tn: 41931.0000 - val_tp: 59.0000
Epoch 62/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9844 - auc:
0.9953 - fn: 9.5800 - fp: 1612.9800 - loss: 0.0963 - precision: 0.0980
- recall: 0.9542 - tn: 102515.9609 - tp: 173.9600 - val_accuracy:
0.9829 - val_auc: 0.9477 - val_fn: 11.0000 - val_fp: 720.0000 -
val_loss: 0.0985 - val_precision: 0.0757 - val_recall: 0.8429 -
val_tn: 41932.0000 - val_tp: 59.0000
Epoch 63/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9839 - auc:
0.9844 - fn: 14.3600 - fp: 1648.2600 - loss: 0.1274 - precision:
0.0867 - recall: 0.9174 - tn: 102483.0234 - tp: 166.8400 -
val_accuracy: 0.9829 - val_auc: 0.9480 - val_fn: 11.0000 - val_fp:
718.0000 - val_loss: 0.0977 - val_precision: 0.0759 - val_recall:
0.8429 - val_tn: 41934.0000 - val_tp: 59.0000
Epoch 64/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9841 - auc:
0.9938 - fn: 11.7000 - fp: 1631.7200 - loss: 0.1120 - precision:
0.0946 - recall: 0.9332 - tn: 102497.2578 - tp: 171.8000 -
val_accuracy: 0.9829 - val_auc: 0.9477 - val_fn: 11.0000 - val_fp:
718.0000 - val_loss: 0.0971 - val_precision: 0.0759 - val_recall:
0.8429 - val_tn: 41934.0000 - val_tp: 59.0000
Epoch 65/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9850 - auc:
0.9934 - fn: 11.1800 - fp: 1579.3199 - loss: 0.1078 - precision:
0.1054 - recall: 0.9431 - tn: 102546.0625 - tp: 175.9200 -
val_accuracy: 0.9830 - val_auc: 0.9478 - val_fn: 11.0000 - val_fp:
716.0000 - val_loss: 0.0967 - val_precision: 0.0761 - val_recall:
0.8429 - val_tn: 41936.0000 - val_tp: 59.0000
Epoch 66/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9843 - auc:
0.9926 - fn: 12.8200 - fp: 1633.1000 - loss: 0.1093 - precision:
0.1040 - recall: 0.9416 - tn: 102490.7188 - tp: 175.8400 -
val_accuracy: 0.9830 - val_auc: 0.9482 - val_fn: 11.0000 - val_fp:
717.0000 - val_loss: 0.0960 - val_precision: 0.0760 - val_recall:
0.8429 - val_tn: 41935.0000 - val_tp: 59.0000
Epoch 67/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9841 - auc:
0.9911 - fn: 12.9400 - fp: 1618.7400 - loss: 0.1101 - precision:
0.0889 - recall: 0.9212 - tn: 102516.3828 - tp: 164.4200 -
val_accuracy: 0.9829 - val_auc: 0.9473 - val_fn: 11.0000 - val_fp:
719.0000 - val_loss: 0.0956 - val_precision: 0.0758 - val_recall:
0.8429 - val_tn: 41933.0000 - val_tp: 59.0000
Epoch 68/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9844 - auc:
0.9959 - fn: 11.1000 - fp: 1605.9399 - loss: 0.0933 - precision:
```

```
0.0931 - recall: 0.9450 - tn: 102527.2812 - tp: 168.1600 -
val_accuracy: 0.9829 - val_auc: 0.9462 - val_fn: 11.0000 - val_fp:
720.0000 - val_loss: 0.0953 - val_precision: 0.0757 - val_recall:
0.8429 - val_tn: 41932.0000 - val_tp: 59.0000
Epoch 69/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9844 - auc:
0.9940 - fn: 10.1800 - fp: 1614.1000 - loss: 0.1012 - precision:
0.0952 - recall: 0.9466 - tn: 102517.6172 - tp: 170.5800 -
val_accuracy: 0.9829 - val_auc: 0.9465 - val_fn: 11.0000 - val_fp:
721.0000 - val_loss: 0.0949 - val_precision: 0.0756 - val_recall:
0.8429 - val_tn: 41931.0000 - val_tp: 59.0000
Epoch 70/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9843 - auc:
0.9911 - fn: 13.7000 - fp: 1638.9399 - loss: 0.1300 - precision:
0.1042 - recall: 0.9206 - tn: 102474.6797 - tp: 185.1600 -
val_accuracy: 0.9831 - val_auc: 0.9468 - val_fn: 11.0000 - val_fp:
711.0000 - val_loss: 0.0940 - val_precision: 0.0766 - val_recall:
0.8429 - val_tn: 41941.0000 - val_tp: 59.0000
Epoch 71/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9845 - auc:
0.9953 - fn: 11.4400 - fp: 1612.7600 - loss: 0.0988 - precision:
0.0978 - recall: 0.9405 - tn: 102515.6562 - tp: 172.6200 -
val_accuracy: 0.9832 - val_auc: 0.9470 - val_fn: 11.0000 - val_fp:
706.0000 - val_loss: 0.0935 - val_precision: 0.0771 - val_recall:
0.8429 - val_tn: 41946.0000 - val_tp: 59.0000
Epoch 72/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9847 - auc:
0.9940 - fn: 13.4600 - fp: 1612.1801 - loss: 0.1131 - precision:
0.1054 - recall: 0.9294 - tn: 102508.3438 - tp: 178.5000 -
val_accuracy: 0.9834 - val_auc: 0.9472 - val_fn: 11.0000 - val_fp:
698.0000 - val_loss: 0.0928 - val_precision: 0.0779 - val_recall:
0.8429 - val_tn: 41954.0000 - val_tp: 59.0000
Epoch 73/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9850 - auc:
0.9944 - fn: 11.9200 - fp: 1541.4600 - loss: 0.1038 - precision:
0.1077 - recall: 0.9387 - tn: 102581.8984 - tp: 177.2000 -
val_accuracy: 0.9834 - val_auc: 0.9473 - val_fn: 11.0000 - val_fp:
700.0000 - val_loss: 0.0924 - val_precision: 0.0777 - val_recall:
0.8429 - val_tn: 41952.0000 - val_tp: 59.0000
Epoch 74/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9845 - auc:
0.9951 - fn: 12.6800 - fp: 1605.8000 - loss: 0.1045 - precision:
0.1033 - recall: 0.9310 - tn: 102515.2031 - tp: 178.8000 -
val_accuracy: 0.9835 - val_auc: 0.9475 - val_fn: 11.0000 - val_fp:
693.0000 - val_loss: 0.0918 - val_precision: 0.0785 - val_recall:
0.8429 - val_tn: 41959.0000 - val_tp: 59.0000
Epoch 75/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9849 - auc:
0.9936 - fn: 12.5000 - fp: 1577.6600 - loss: 0.1100 - precision:
0.0980 - recall: 0.9298 - tn: 102553.9375 - tp: 168.3800 -
```

```
val_accuracy: 0.9836 - val_auc: 0.9477 - val_fn: 11.0000 - val_fp:
691.0000 - val_loss: 0.0913 - val_precision: 0.0787 - val_recall:
0.8429 - val_tn: 41961.0000 - val_tp: 59.0000
Epoch 76/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9848 - auc:
0.9934 - fn: 12.1400 - fp: 1584.3000 - loss: 0.1014 - precision:
0.0940 - recall: 0.9334 - tn: 102553.6562 - tp: 162.3800 -
val_accuracy: 0.9836 - val_auc: 0.9473 - val_fn: 11.0000 - val_fp:
690.0000 - val_loss: 0.0911 - val_precision: 0.0788 - val_recall:
0.8429 - val_tn: 41962.0000 - val_tp: 59.0000
Epoch 77/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9851 - auc:
0.9930 - fn: 11.9000 - fp: 1551.7800 - loss: 0.0992 - precision:
0.0978 - recall: 0.9431 - tn: 102575.5781 - tp: 173.2200 -
val_accuracy: 0.9836 - val_auc: 0.9474 - val_fn: 11.0000 - val_fp:
690.0000 - val_loss: 0.0908 - val_precision: 0.0788 - val_recall:
0.8429 - val_tn: 41962.0000 - val_tp: 59.0000
Epoch 78/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9847 - auc:
0.9939 - fn: 11.4200 - fp: 1577.7000 - loss: 0.0928 - precision:
0.0917 - recall: 0.9509 - tn: 102553.3594 - tp: 170.0000 -
val_accuracy: 0.9836 - val_auc: 0.9476 - val_fn: 11.0000 - val_fp:
690.0000 - val_loss: 0.0903 - val_precision: 0.0788 - val_recall:
0.8429 - val_tn: 41962.0000 - val_tp: 59.0000
Epoch 79/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.9844 - auc:
0.9932 - fn: 14.3800 - fp: 1611.9600 - loss: 0.1060 - precision:
0.1001 - recall: 0.9310 - tn: 102508.9766 - tp: 177.1600 -
val_accuracy: 0.9836 - val_auc: 0.9479 - val_fn: 11.0000 - val_fp:
688.0000 - val_loss: 0.0897 - val_precision: 0.0790 - val_recall:
0.8429 - val_tn: 41964.0000 - val_tp: 59.0000
Epoch 80/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.9847 - auc:
0.9950 - fn: 10.2000 - fp: 1584.9000 - loss: 0.0930 - precision:
0.1038 - recall: 0.9552 - tn: 102535.5234 - tp: 181.8600 -
val_accuracy: 0.9837 - val_auc: 0.9480 - val_fn: 11.0000 - val_fp:
687.0000 - val_loss: 0.0894 - val_precision: 0.0791 - val_recall:
0.8429 - val_tn: 41965.0000 - val_tp: 59.0000
Epoch 81/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9842 - auc:
0.9924 - fn: 13.7000 - fp: 1609.3800 - loss: 0.1089 - precision:
0.1025 - recall: 0.9324 - tn: 102512.4219 - tp: 176.9800 -
val_accuracy: 0.9837 - val_auc: 0.9481 - val_fn: 11.0000 - val_fp:
687.0000 - val_loss: 0.0891 - val_precision: 0.0791 - val_recall:
0.8429 - val_tn: 41965.0000 - val_tp: 59.0000
Epoch 82/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.9847 - auc:
0.9959 - fn: 11.9600 - fp: 1577.4200 - loss: 0.0993 - precision:
0.1024 - recall: 0.9352 - tn: 102547.5234 - tp: 175.5800 -
```

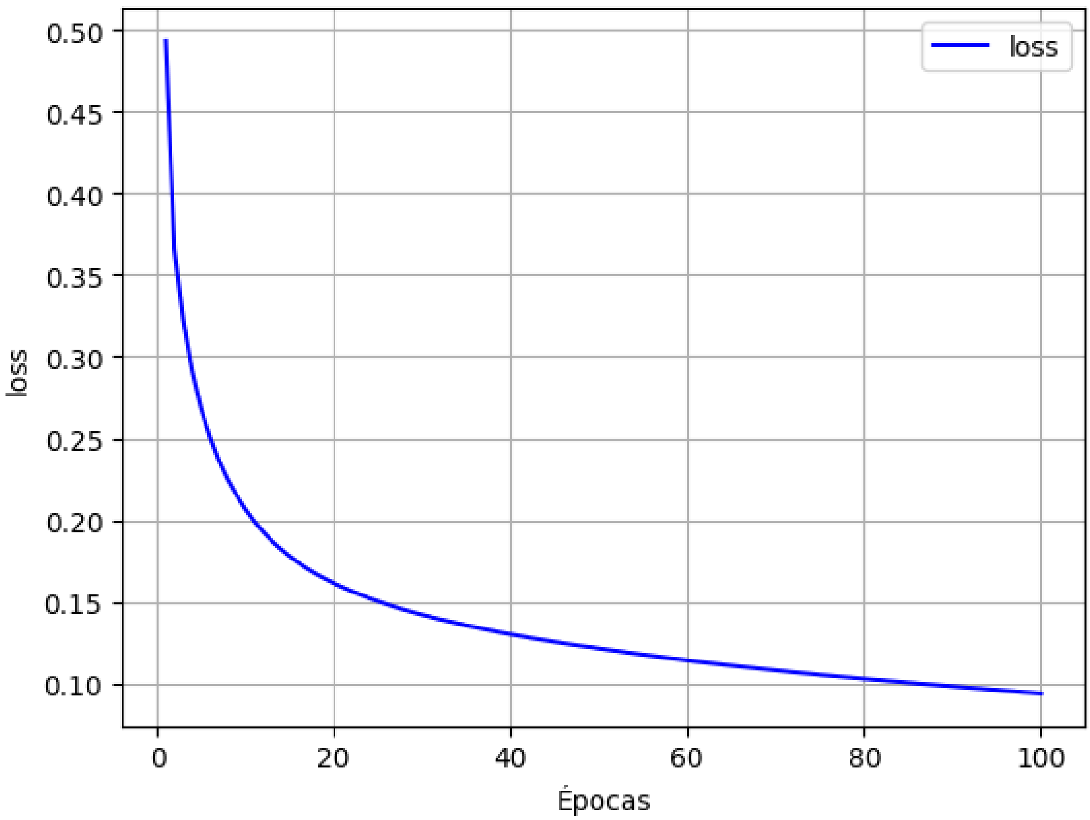
```
val_accuracy: 0.9838 - val_auc: 0.9482 - val_fn: 11.0000 - val_fp:
680.0000 - val_loss: 0.0887 - val_precision: 0.0798 - val_recall:
0.8429 - val_tn: 41972.0000 - val_tp: 59.0000
Epoch 83/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9846 - auc:
0.9929 - fn: 9.9200 - fp: 1576.2000 - loss: 0.0900 - precision: 0.0928
- recall: 0.9507 - tn: 102558.3828 - tp: 167.9800 - val_accuracy:
0.9837 - val_auc: 0.9484 - val_fn: 11.0000 - val_fp: 687.0000 -
val_loss: 0.0889 - val_precision: 0.0791 - val_recall: 0.8429 -
val_tn: 41965.0000 - val_tp: 59.0000
Epoch 84/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9842 - auc:
0.9948 - fn: 13.5000 - fp: 1617.9399 - loss: 0.1019 - precision:
0.0953 - recall: 0.9266 - tn: 102509.9375 - tp: 171.1000 -
val_accuracy: 0.9838 - val_auc: 0.9486 - val_fn: 11.0000 - val_fp:
683.0000 - val_loss: 0.0884 - val_precision: 0.0795 - val_recall:
0.8429 - val_tn: 41969.0000 - val_tp: 59.0000
Epoch 85/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9849 - auc:
0.9942 - fn: 10.8600 - fp: 1571.4600 - loss: 0.0963 - precision:
0.1000 - recall: 0.9487 - tn: 102554.7188 - tp: 175.4400 -
val_accuracy: 0.9838 - val_auc: 0.9487 - val_fn: 11.0000 - val_fp:
683.0000 - val_loss: 0.0880 - val_precision: 0.0795 - val_recall:
0.8429 - val_tn: 41969.0000 - val_tp: 59.0000
Epoch 86/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9847 - auc:
0.9942 - fn: 13.3400 - fp: 1587.9000 - loss: 0.1059 - precision:
0.0941 - recall: 0.9100 - tn: 102543.7812 - tp: 167.4600 -
val_accuracy: 0.9838 - val_auc: 0.9489 - val_fn: 11.0000 - val_fp:
679.0000 - val_loss: 0.0874 - val_precision: 0.0799 - val_recall:
0.8429 - val_tn: 41973.0000 - val_tp: 59.0000
Epoch 87/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9846 - auc:
0.9926 - fn: 14.4200 - fp: 1625.8199 - loss: 0.1249 - precision:
0.0980 - recall: 0.9121 - tn: 102498.8438 - tp: 173.4000 -
val_accuracy: 0.9840 - val_auc: 0.9491 - val_fn: 11.0000 - val_fp:
673.0000 - val_loss: 0.0864 - val_precision: 0.0806 - val_recall:
0.8429 - val_tn: 41979.0000 - val_tp: 59.0000
Epoch 88/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9844 - auc:
0.9931 - fn: 11.7800 - fp: 1599.1400 - loss: 0.1108 - precision:
0.0932 - recall: 0.9296 - tn: 102533.8828 - tp: 167.6800 -
val_accuracy: 0.9841 - val_auc: 0.9492 - val_fn: 11.0000 - val_fp:
670.0000 - val_loss: 0.0861 - val_precision: 0.0809 - val_recall:
0.8429 - val_tn: 41982.0000 - val_tp: 59.0000
Epoch 89/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9847 - auc:
0.9947 - fn: 12.9000 - fp: 1585.7400 - loss: 0.1056 - precision:
0.1070 - recall: 0.9324 - tn: 102531.6406 - tp: 182.2000 -
```

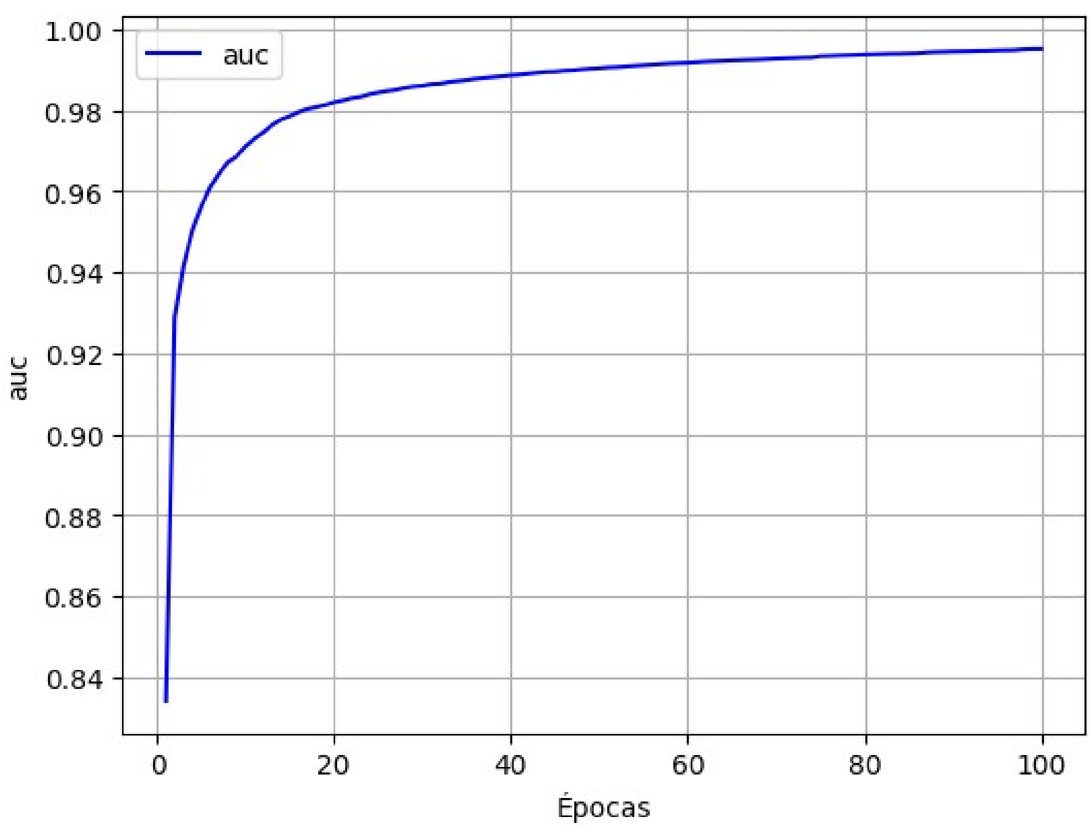
```
val_accuracy: 0.9841 - val_auc: 0.9494 - val_fn: 11.0000 - val_fp:
670.0000 - val_loss: 0.0856 - val_precision: 0.0809 - val_recall:
0.8429 - val_tn: 41982.0000 - val_tp: 59.0000
Epoch 90/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9852 - auc:
0.9960 - fn: 9.8800 - fp: 1548.3600 - loss: 0.0917 - precision: 0.1089
- recall: 0.9534 - tn: 102576.6172 - tp: 177.6200 - val_accuracy:
0.9841 - val_auc: 0.9495 - val_fn: 11.0000 - val_fp: 669.0000 -
val_loss: 0.0854 - val_precision: 0.0810 - val_recall: 0.8429 -
val_tn: 41983.0000 - val_tp: 59.0000
Epoch 91/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9845 - auc:
0.9927 - fn: 12.9600 - fp: 1593.2600 - loss: 0.1118 - precision:
0.1002 - recall: 0.9284 - tn: 102529.0391 - tp: 177.2200 -
val_accuracy: 0.9841 - val_auc: 0.9497 - val_fn: 11.0000 - val_fp:
670.0000 - val_loss: 0.0852 - val_precision: 0.0809 - val_recall:
0.8429 - val_tn: 41982.0000 - val_tp: 59.0000
Epoch 92/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9844 - auc:
0.9939 - fn: 15.3600 - fp: 1607.7000 - loss: 0.1124 - precision:
0.0923 - recall: 0.9116 - tn: 102521.8984 - tp: 167.5200 -
val_accuracy: 0.9842 - val_auc: 0.9499 - val_fn: 11.0000 - val_fp:
663.0000 - val_loss: 0.0846 - val_precision: 0.0817 - val_recall:
0.8429 - val_tn: 41989.0000 - val_tp: 59.0000
Epoch 93/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9855 - auc:
0.9960 - fn: 10.6800 - fp: 1527.9000 - loss: 0.0839 - precision:
0.1057 - recall: 0.9497 - tn: 102598.6016 - tp: 175.3000 -
val_accuracy: 0.9840 - val_auc: 0.9500 - val_fn: 11.0000 - val_fp:
671.0000 - val_loss: 0.0848 - val_precision: 0.0808 - val_recall:
0.8429 - val_tn: 41981.0000 - val_tp: 59.0000
Epoch 94/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9847 - auc:
0.9967 - fn: 10.8400 - fp: 1567.5601 - loss: 0.0852 - precision:
0.1057 - recall: 0.9484 - tn: 102557.3438 - tp: 176.7400 -
val_accuracy: 0.9841 - val_auc: 0.9501 - val_fn: 11.0000 - val_fp:
670.0000 - val_loss: 0.0845 - val_precision: 0.0809 - val_recall:
0.8429 - val_tn: 41982.0000 - val_tp: 59.0000
Epoch 95/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9844 - auc:
0.9928 - fn: 12.6800 - fp: 1603.5601 - loss: 0.1160 - precision:
0.1036 - recall: 0.9228 - tn: 102516.4219 - tp: 179.8200 -
val_accuracy: 0.9842 - val_auc: 0.9503 - val_fn: 11.0000 - val_fp:
666.0000 - val_loss: 0.0842 - val_precision: 0.0814 - val_recall:
0.8429 - val_tn: 41986.0000 - val_tp: 59.0000
Epoch 96/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9847 - auc:
0.9962 - fn: 11.7000 - fp: 1586.6600 - loss: 0.0876 - precision:
0.0984 - recall: 0.9368 - tn: 102543.8203 - tp: 170.3000 -
```

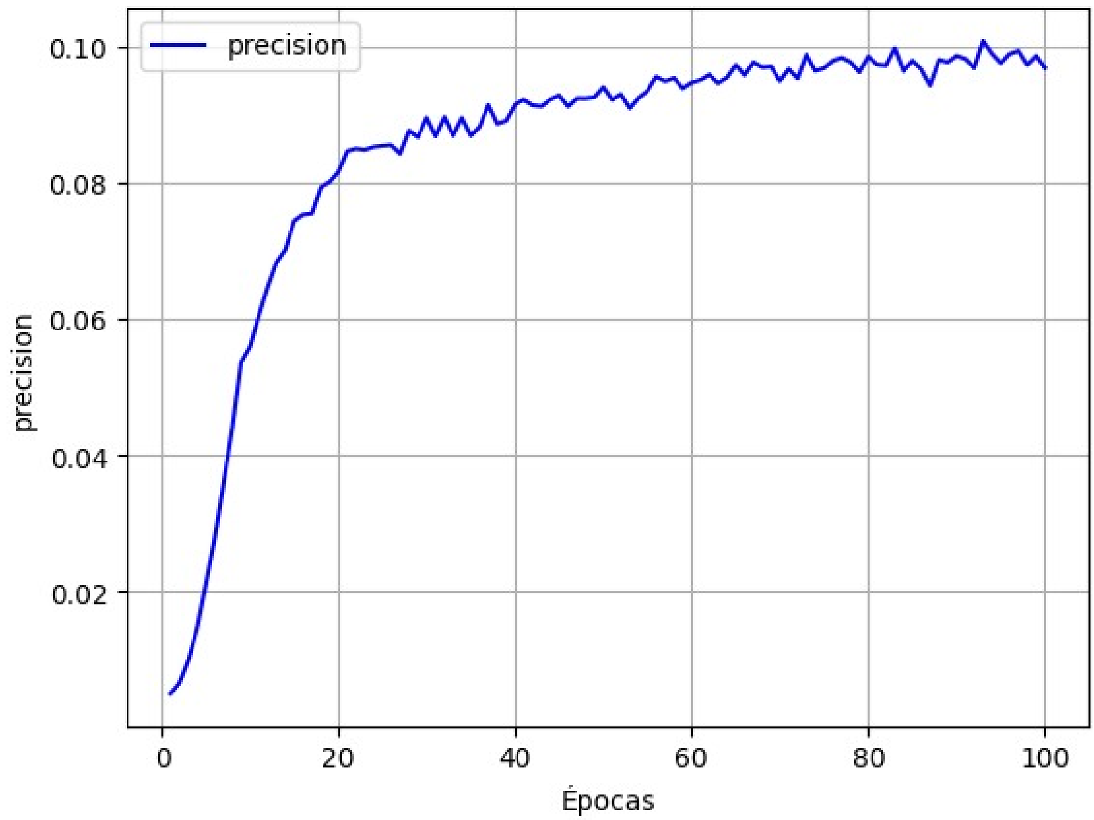
```
val_accuracy: 0.9841 - val_auc: 0.9505 - val_fn: 11.0000 - val_fp:
667.0000 - val_loss: 0.0839 - val_precision: 0.0813 - val_recall:
0.8429 - val_tn: 41985.0000 - val_tp: 59.0000
Epoch 97/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9848 - auc:
0.9966 - fn: 10.6800 - fp: 1563.6200 - loss: 0.0903 - precision:
0.1024 - recall: 0.9355 - tn: 102562.8438 - tp: 175.3400 -
val_accuracy: 0.9840 - val_auc: 0.9506 - val_fn: 11.0000 - val_fp:
672.0000 - val_loss: 0.0841 - val_precision: 0.0807 - val_recall:
0.8429 - val_tn: 41980.0000 - val_tp: 59.0000
Epoch 98/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9848 - auc:
0.9949 - fn: 12.3600 - fp: 1582.9399 - loss: 0.1022 - precision:
0.1014 - recall: 0.9313 - tn: 102546.3594 - tp: 170.8200 -
val_accuracy: 0.9842 - val_auc: 0.9485 - val_fn: 11.0000 - val_fp:
665.0000 - val_loss: 0.0834 - val_precision: 0.0815 - val_recall:
0.8429 - val_tn: 41987.0000 - val_tp: 59.0000
Epoch 99/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.9845 - auc:
0.9933 - fn: 12.2600 - fp: 1598.1801 - loss: 0.1105 - precision:
0.0896 - recall: 0.9177 - tn: 102536.7031 - tp: 165.3400 -
val_accuracy: 0.9842 - val_auc: 0.9486 - val_fn: 11.0000 - val_fp:
664.0000 - val_loss: 0.0833 - val_precision: 0.0816 - val_recall:
0.8429 - val_tn: 41988.0000 - val_tp: 59.0000
Epoch 100/100
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - accuracy: 0.9843 - auc:
0.9921 - fn: 12.5200 - fp: 1613.1000 - loss: 0.1054 - precision:
0.0866 - recall: 0.9284 - tn: 102523.6562 - tp: 163.2000 -
val_accuracy: 0.9844 - val_auc: 0.9489 - val_fn: 11.0000 - val_fp:
657.0000 - val_loss: 0.0826 - val_precision: 0.0824 - val_recall:
0.8429 - val_tn: 41995.0000 - val_tp: 59.0000
```

Visualização da função de custo e das métricas

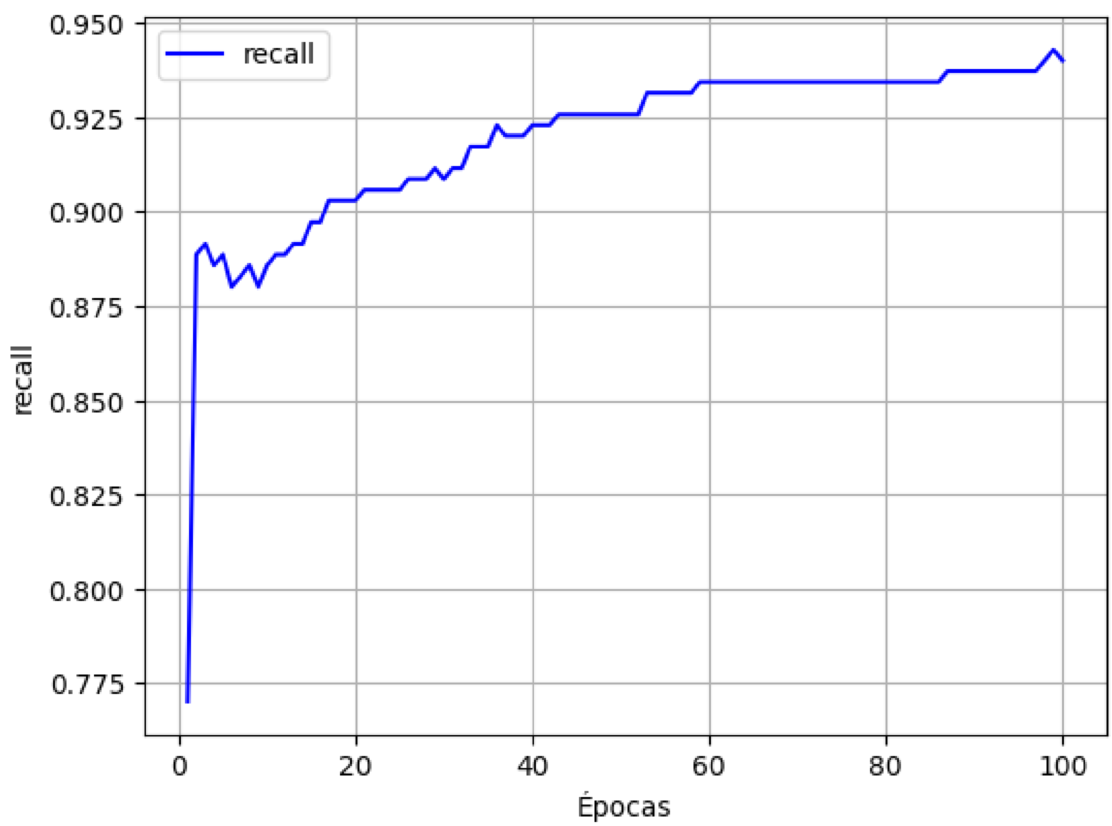
```
plot_metrics(pond_history, metrics)
```


loss
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0
20
40
60
80
100
Épocas


auc
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0
20
40
60
80
100
Épocas



precision
0.10
0.08
0.06
0.04
0.02
0
20
40
60
80
100
Épocas



recall
0.950
0.925
0.900
0.875
0.850
0.825
0.800
0.775
0
20
40
60
80
100
Épocas

Observa-se pelo gráfico da função de custo que está ocorrendo "overfitting".

Avaliação das métricas

```
train_pred_pond = rna_pond.predict(train_features,
batch_size=BATCH_SIZE)
test_pred_pond = rna_pond.predict(test_features,
batch_size=BATCH_SIZE)
```

```
49/49 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
11/11 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step
```

```
print('Número de exemplos positivos do conjunto de teste =',
len(test_labels[test_labels>0.9]))
pond_results = rna_pond.evaluate(test_features, test_labels,
                                 batch_size=BATCH_SIZE, verbose=0)
for name, value in zip(rna_pond.metrics_names, pond_results):
    print(name, ': ', value)
print()
```

```
Número de exemplos positivos do conjunto de teste = 72
loss :  0.07928558439016342
compile_metrics :  66.0
```

Cálculo da Pontuação F1

```
precision = pond_results[5]
recall = pond_results[6]
F1_pond = 2*precision*recall/(precision + recall)
print('Pontuação F1 = ', F1_pond)
```

```
Pontuação F1 =  0.17277509844253955
```

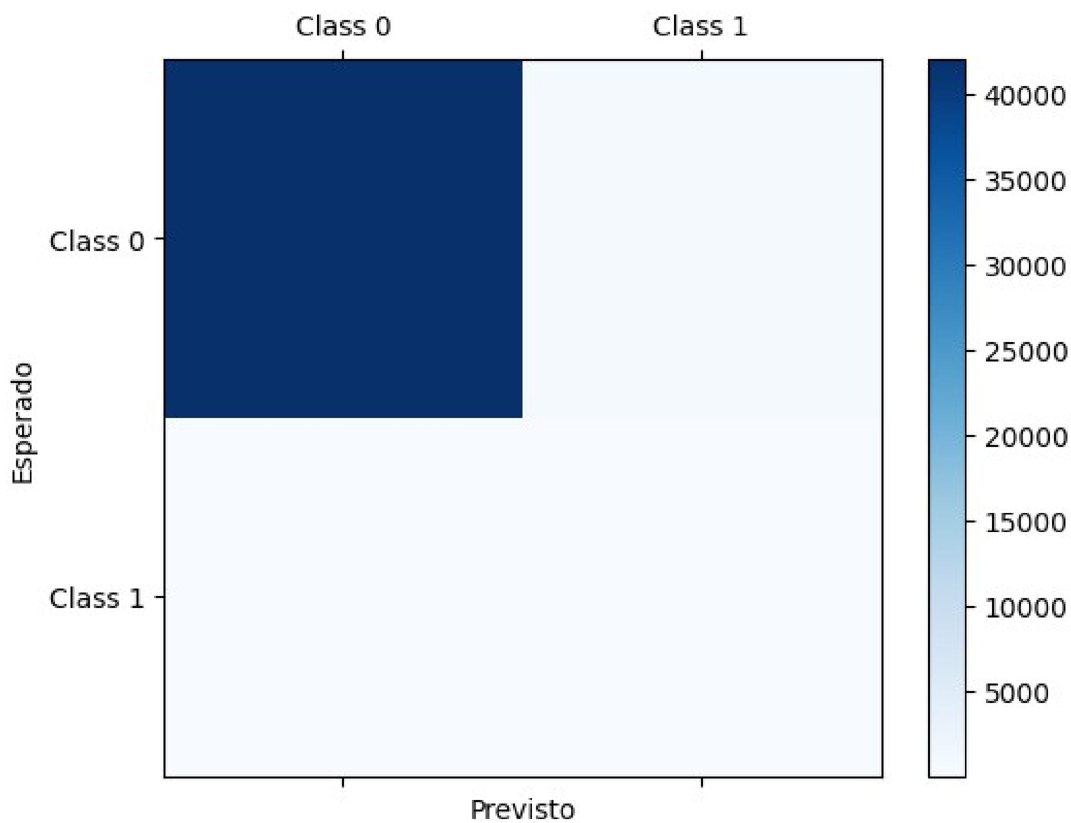
Matriz de confusão

Podemos usar a matriz de confusão para visualizar melhor as classes reais e previstas.

```
from sklearn.metrics import confusion_matrix
from matplotlib import pyplot as plt

conf_mat = confusion_matrix(y_true=test_labels,
y_pred=np.round(test_pred_pond))
print('Matriz de confusão:\n', conf_mat)

labels = ['Class 0', 'Class 1']
plt.figure(figsize=(6,6))
fig = plt.figure()
ax = fig.add_subplot(111)
cax = ax.matshow(conf_mat, cmap=plt.cm.Blues)
fig.colorbar(cax)
```

```
ax.set_xticklabels([''] + labels)
ax.set_yticklabels([''] + labels)
plt.xlabel('Previsto')
plt.ylabel('Esperado')
plt.show()
```

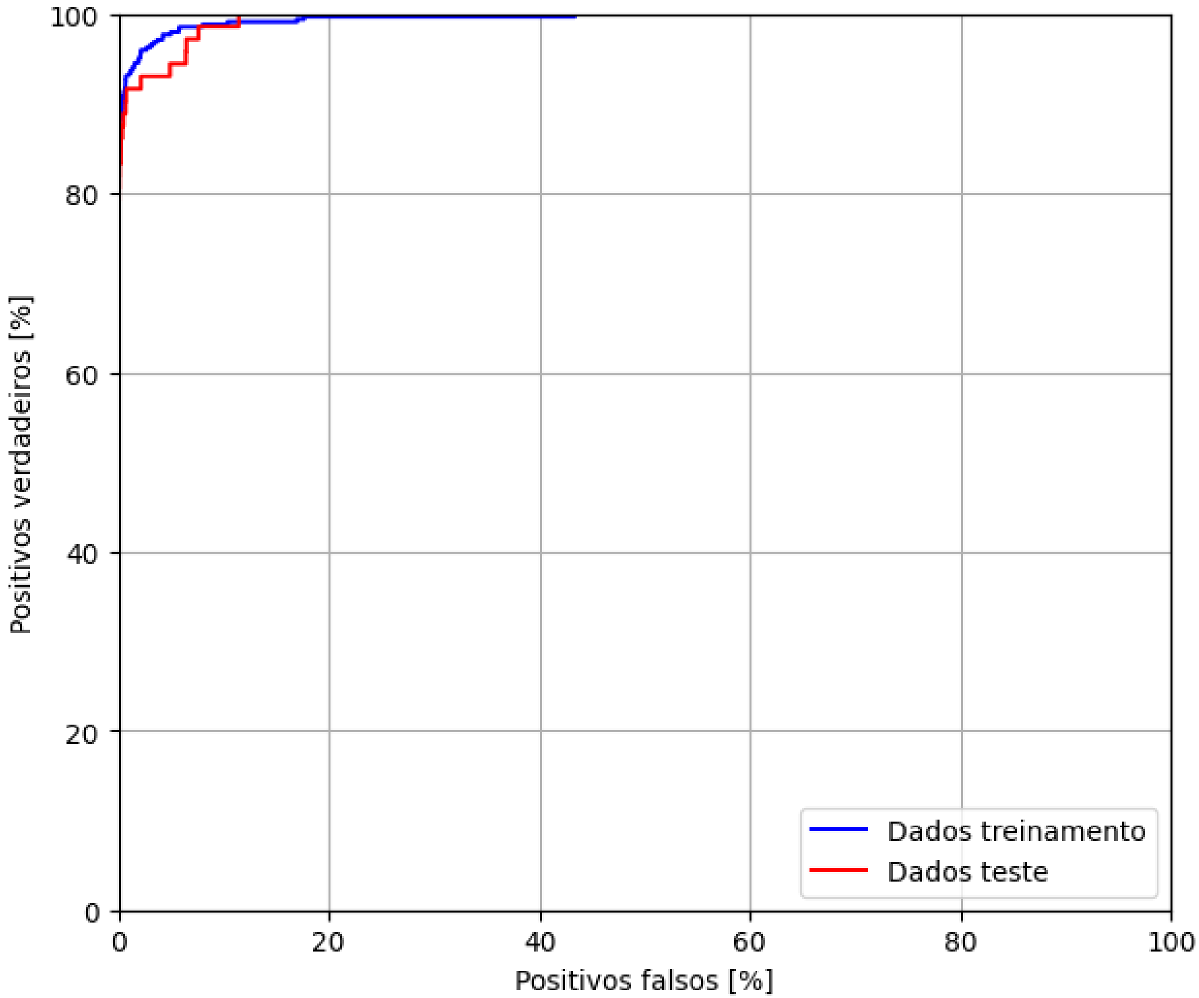
```
Matriz de confusão:
 [[42018   631]
 [    6    66]]

<ipython-input-31-2d87e45c000d>:13: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_xticklabels([''] + labels)
<ipython-input-31-2d87e45c000d>:14: UserWarning: FixedFormatter should only be used together with FixedLocator
  ax.set_yticklabels([''] + labels)

<Figure size 600x600 with 0 Axes>
```



Podemos ver que com pesos de classe, a exatidão e a precisão são menores porque há mais falsos positivos, mas inversamente a revocação e a AUC são maiores porque o modelo também encontrou mais positivos verdadeiros. Apesar de ter menor precisão, este modelo tem maior revocação (e identifica mais transações fraudulentas). Obviamente, há um custo para os dois tipos de erro, pois, não é interssante incomodar os usuários sinalizando muitas transações legítimas como fraudulentas.

Gráfico do ROC

```
fp_train, tp_train, _ = sklearn.metrics.roc_curve(train_labels,
train_pred_pond)
fp_test, tp_test, _ = sklearn.metrics.roc_curve(test_labels,
test_pred_pond)
plt.figure(figsize=(7, 6))
plt.plot(100*fp_train, 100*tp_train, 'b', label='Dados treinamento')
plt.plot(100*fp_test, 100*tp_test, 'r', label='Dados teste')
plt.xlabel('Positivos falsos [%]')
plt.ylabel('Positivos verdadeiros [%]')
plt.xlim([0,100])
plt.ylim([0,100])
plt.grid(True)
ax = plt.gca()
plt.legend(loc='lower right')
plt.show()
```



7. Conclusão

A classificação de dados com número de exemplos desequilibrado é uma tarefa inerentemente difícil, pois podem existir poucos exemplos para o treinamento. Assim, deve-se sempre começar usando os dados originais e fazer o possível para coletar o maior número possível de exemplos e analisar quais recursos podem ser relevantes para que o modelo possa obter o máximo da sua classe minoritária. Em algum ponto, o modelo pode ter dificuldades para melhorar e produzir os resultados desejados, portanto, é importante ter em mente o contexto do problema e o compromisso entre os diferentes tipos de erros.