mlp_models2

May 6, 2025

[1]:
```
from pandas import read_csv, DataFrame, Series
from pandas.plotting import scatter_matrix
from numpy import set_printoptions, argmax, isnan, nan, mean, random
import seaborn as sns
import statsmodels.api as sm
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler, Normalizer
from sklearn.compose import ColumnTransformer
from sklearn.pipeline import Pipeline
from sklearn.model_selection import train_test_split, KFold
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.feature_selection import RFE
import tensorflow as tf
import numpy as np
import pandas as pd
print(tf.__version__)
print("Num GPUs Available: ", len(tf.config.list_physical_devices('GPU')))
```

```
2.19.0
Num GPUs Available:  0
```

[2]:
```
# Set random seeds for reproducibility
tf.keras.backend.clear_session()
random.seed(42)
tf.random.set_seed(42)

from perform_kickstarer_eda import X_train, X_test, y_train, y_test

# Define the model architecture
input_layer = tf.keras.layers.Input(shape=(X_train.shape[1],))
hidden = tf.keras.layers.Dense(32, activation="relu",␣
  ↪kernel_initializer='he_normal')(input_layer)
# Change output layer to sigmoid for binary classification
output = tf.keras.layers.Dense(1, activation="sigmoid")(hidden)

model = tf.keras.models.Model(inputs=[input_layer], outputs=[output])
# Change to binary_crossentropy loss and classification metrics
```

```
model.compile(
    loss="binary_crossentropy",
    optimizer="adam",
    metrics=["accuracy", tf.keras.metrics.AUC(), tf.keras.metrics.Precision(),␣
 ↪tf.keras.metrics.Recall()]
)
model.summary()
```

```
ks_data shape: (20632, 68)
Categorical columns: ['country', 'currency', 'deadline', 'state_changed_at',
'created_at', 'launched_at', 'category', 'deadline_weekday',
'state_changed_at_weekday', 'created_at_weekday', 'launched_at_weekday',
'launch_to_deadline', 'launch_to_state_change']
Converted boolean column: staff_pick
Dropped 'state_changed_at_weekday' column
Dropping original column: category
Encoded column: category → 25 features
Dropping original column: deadline_weekday
Encoded column: deadline_weekday → 7 features
Dropping original column: created_at_weekday
Encoded column: created_at_weekday → 7 features
Dropping original column: launched_at_weekday
Encoded column: launched_at_weekday → 7 features
Total categorical columns after encoding: 54
```

```
Model: "functional"


 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 32)                      2,144

 dense_1 (Dense)                   (None, 1)                          33


 Total params: 2,177 (8.50 KB)

 Trainable params: 2,177 (8.50 KB)

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

[3]:

```
# Train the classifier
history = model.fit(
    X_train, y_train,
    epochs=50,
    validation_data=(X_test, y_test),
    verbose=1
)
```

```
Epoch 1/50
```

```
/opt/anaconda3/envs/tf/lib/python3.12/site-
packages/keras/src/models/functional.py:238: UserWarning: The structure of
`inputs` doesn't match the expected structure.
Expected: ['keras_tensor']
Received: inputs=Tensor(shape=(None, 66))
  warnings.warn(msg)
```

```
487/516            0s 934us/step -
accuracy: 0.8073 - auc: 0.8370 - loss: 0.4257 - precision: 0.7495 - recall:
0.4447
```

```
/opt/anaconda3/envs/tf/lib/python3.12/site-
packages/keras/src/models/functional.py:238: UserWarning: The structure of
`inputs` doesn't match the expected structure.
Expected: ['keras_tensor']
Received: inputs=Tensor(shape=(None, 66))
  warnings.warn(msg)
```

```
516/516            2s 2ms/step -
accuracy: 0.8109 - auc: 0.8430 - loss: 0.4189 - precision: 0.7550 - recall:
0.4579 - val_accuracy: 0.9144 - val_auc: 0.9692 - val_loss: 0.2139 -
val_precision: 0.8602 - val_recall: 0.8497
Epoch 2/50
516/516            4s 8ms/step -
accuracy: 0.9176 - auc: 0.9725 - loss: 0.1999 - precision: 0.8636 - recall:
0.8508 - val_accuracy: 0.9169 - val_auc: 0.9725 - val_loss: 0.1960 -
val_precision: 0.8608 - val_recall: 0.8587
Epoch 3/50
516/516            1s 2ms/step -
accuracy: 0.9219 - auc: 0.9750 - loss: 0.1860 - precision: 0.8691 - recall:
0.8608 - val_accuracy: 0.9200 - val_auc: 0.9736 - val_loss: 0.1911 -
val_precision: 0.8640 - val_recall: 0.8668
Epoch 4/50
516/516            1s 2ms/step -
accuracy: 0.9234 - auc: 0.9760 - loss: 0.1812 - precision: 0.8704 - recall:
0.8653 - val_accuracy: 0.9212 - val_auc: 0.9741 - val_loss: 0.1886 -
val_precision: 0.8652 - val_recall: 0.8701
Epoch 5/50
516/516            1s 2ms/step -
accuracy: 0.9244 - auc: 0.9768 - loss: 0.1782 - precision: 0.8708 - recall:
0.8685 - val_accuracy: 0.9220 - val_auc: 0.9747 - val_loss: 0.1863 -
val_precision: 0.8661 - val_recall: 0.8717
```

```
Epoch 6/50
516/516            1s 2ms/step -
accuracy: 0.9256 - auc: 0.9774 - loss: 0.1757 - precision: 0.8720 - recall:
0.8719 - val_accuracy: 0.9224 - val_auc: 0.9753 - val_loss: 0.1842 -
val_precision: 0.8663 - val_recall: 0.8734
Epoch 7/50
516/516            1s 1ms/step -
accuracy: 0.9267 - auc: 0.9779 - loss: 0.1733 - precision: 0.8723 - recall:
0.8758 - val_accuracy: 0.9227 - val_auc: 0.9757 - val_loss: 0.1824 -
val_precision: 0.8670 - val_recall: 0.8734
Epoch 8/50
516/516            1s 2ms/step -
accuracy: 0.9274 - auc: 0.9785 - loss: 0.1711 - precision: 0.8736 - recall:
0.8772 - val_accuracy: 0.9220 - val_auc: 0.9760 - val_loss: 0.1810 -
val_precision: 0.8661 - val_recall: 0.8717
Epoch 9/50
516/516            1s 2ms/step -
accuracy: 0.9284 - auc: 0.9789 - loss: 0.1688 - precision: 0.8741 - recall:
0.8805 - val_accuracy: 0.9222 - val_auc: 0.9765 - val_loss: 0.1794 -
val_precision: 0.8662 - val_recall: 0.8725
Epoch 10/50
516/516            1s 2ms/step -
accuracy: 0.9295 - auc: 0.9794 - loss: 0.1665 - precision: 0.8753 - recall:
0.8832 - val_accuracy: 0.9229 - val_auc: 0.9770 - val_loss: 0.1776 -
val_precision: 0.8689 - val_recall: 0.8717
Epoch 11/50
516/516            1s 1ms/step -
accuracy: 0.9300 - auc: 0.9800 - loss: 0.1640 - precision: 0.8753 - recall:
0.8851 - val_accuracy: 0.9241 - val_auc: 0.9774 - val_loss: 0.1759 -
val_precision: 0.8737 - val_recall: 0.8701
Epoch 12/50
516/516            1s 2ms/step -
accuracy: 0.9312 - auc: 0.9805 - loss: 0.1619 - precision: 0.8774 - recall:
0.8872 - val_accuracy: 0.9246 - val_auc: 0.9775 - val_loss: 0.1750 -
val_precision: 0.8751 - val_recall: 0.8701
Epoch 13/50
516/516            1s 1ms/step -
accuracy: 0.9316 - auc: 0.9810 - loss: 0.1598 - precision: 0.8787 - recall:
0.8870 - val_accuracy: 0.9244 - val_auc: 0.9777 - val_loss: 0.1744 -
val_precision: 0.8738 - val_recall: 0.8709
Epoch 14/50
516/516            1s 2ms/step -
accuracy: 0.9325 - auc: 0.9814 - loss: 0.1580 - precision: 0.8799 - recall:
0.8890 - val_accuracy: 0.9246 - val_auc: 0.9779 - val_loss: 0.1737 -
val_precision: 0.8739 - val_recall: 0.8717
Epoch 15/50
516/516            2s 4ms/step -
accuracy: 0.9341 - auc: 0.9818 - loss: 0.1562 - precision: 0.8818 - recall:
```

```
0.8928 - val_accuracy: 0.9244 - val_auc: 0.9781 - val_loss: 0.1727 -
val_precision: 0.8732 - val_recall: 0.8717
Epoch 16/50
516/516            1s 3ms/step -
accuracy: 0.9358 - auc: 0.9822 - loss: 0.1546 - precision: 0.8846 - recall:
0.8961 - val_accuracy: 0.9249 - val_auc: 0.9783 - val_loss: 0.1720 -
val_precision: 0.8734 - val_recall: 0.8734
Epoch 17/50
516/516            1s 1ms/step -
accuracy: 0.9362 - auc: 0.9825 - loss: 0.1531 - precision: 0.8850 - recall:
0.8967 - val_accuracy: 0.9266 - val_auc: 0.9785 - val_loss: 0.1717 -
val_precision: 0.8753 - val_recall: 0.8775
Epoch 18/50
516/516            1s 1ms/step -
accuracy: 0.9376 - auc: 0.9826 - loss: 0.1516 - precision: 0.8855 - recall:
0.9017 - val_accuracy: 0.9263 - val_auc: 0.9783 - val_loss: 0.1711 -
val_precision: 0.8758 - val_recall: 0.8758
Epoch 19/50
516/516            1s 1ms/step -
accuracy: 0.9383 - auc: 0.9829 - loss: 0.1503 - precision: 0.8881 - recall:
0.9012 - val_accuracy: 0.9275 - val_auc: 0.9787 - val_loss: 0.1708 -
val_precision: 0.8776 - val_recall: 0.8783
Epoch 20/50
516/516            1s 1ms/step -
accuracy: 0.9392 - auc: 0.9832 - loss: 0.1491 - precision: 0.8901 - recall:
0.9021 - val_accuracy: 0.9295 - val_auc: 0.9788 - val_loss: 0.1701 -
val_precision: 0.8802 - val_recall: 0.8824
Epoch 21/50
516/516            1s 989us/step -
accuracy: 0.9397 - auc: 0.9835 - loss: 0.1479 - precision: 0.8905 - recall:
0.9034 - val_accuracy: 0.9304 - val_auc: 0.9786 - val_loss: 0.1701 -
val_precision: 0.8806 - val_recall: 0.8856
Epoch 22/50
516/516            1s 1ms/step -
accuracy: 0.9399 - auc: 0.9837 - loss: 0.1467 - precision: 0.8911 - recall:
0.9034 - val_accuracy: 0.9317 - val_auc: 0.9784 - val_loss: 0.1698 -
val_precision: 0.8811 - val_recall: 0.8897
Epoch 23/50
516/516            1s 1ms/step -
accuracy: 0.9399 - auc: 0.9839 - loss: 0.1455 - precision: 0.8908 - recall:
0.9038 - val_accuracy: 0.9314 - val_auc: 0.9785 - val_loss: 0.1694 -
val_precision: 0.8797 - val_recall: 0.8905
Epoch 24/50
516/516            1s 1ms/step -
accuracy: 0.9400 - auc: 0.9842 - loss: 0.1444 - precision: 0.8907 - recall:
0.9044 - val_accuracy: 0.9321 - val_auc: 0.9786 - val_loss: 0.1694 -
val_precision: 0.8819 - val_recall: 0.8905
Epoch 25/50
```

```
516/516            1s 1ms/step -
accuracy: 0.9403 - auc: 0.9844 - loss: 0.1435 - precision: 0.8909 - recall:
0.9054 - val_accuracy: 0.9326 - val_auc: 0.9787 - val_loss: 0.1693 -
val_precision: 0.8821 - val_recall: 0.8922
Epoch 26/50
516/516            1s 983us/step -
accuracy: 0.9410 - auc: 0.9846 - loss: 0.1426 - precision: 0.8928 - recall:
0.9056 - val_accuracy: 0.9317 - val_auc: 0.9787 - val_loss: 0.1693 -
val_precision: 0.8798 - val_recall: 0.8913
Epoch 27/50
516/516            1s 992us/step -
accuracy: 0.9414 - auc: 0.9847 - loss: 0.1417 - precision: 0.8921 - recall:
0.9081 - val_accuracy: 0.9317 - val_auc: 0.9787 - val_loss: 0.1693 -
val_precision: 0.8805 - val_recall: 0.8905
Epoch 28/50
516/516            1s 1ms/step -
accuracy: 0.9416 - auc: 0.9849 - loss: 0.1408 - precision: 0.8928 - recall:
0.9080 - val_accuracy: 0.9317 - val_auc: 0.9787 - val_loss: 0.1694 -
val_precision: 0.8805 - val_recall: 0.8905
Epoch 29/50
516/516            1s 1ms/step -
accuracy: 0.9424 - auc: 0.9851 - loss: 0.1401 - precision: 0.8932 - recall:
0.9106 - val_accuracy: 0.9326 - val_auc: 0.9787 - val_loss: 0.1695 -
val_precision: 0.8827 - val_recall: 0.8913
Epoch 30/50
516/516            1s 1ms/step -
accuracy: 0.9429 - auc: 0.9852 - loss: 0.1391 - precision: 0.8939 - recall:
0.9115 - val_accuracy: 0.9317 - val_auc: 0.9787 - val_loss: 0.1694 -
val_precision: 0.8805 - val_recall: 0.8905
Epoch 31/50
516/516            1s 1ms/step -
accuracy: 0.9438 - auc: 0.9854 - loss: 0.1384 - precision: 0.8954 - recall:
0.9131 - val_accuracy: 0.9317 - val_auc: 0.9787 - val_loss: 0.1697 -
val_precision: 0.8798 - val_recall: 0.8913
Epoch 32/50
516/516            1s 1ms/step -
accuracy: 0.9439 - auc: 0.9855 - loss: 0.1376 - precision: 0.8948 - recall:
0.9143 - val_accuracy: 0.9314 - val_auc: 0.9786 - val_loss: 0.1698 -
val_precision: 0.8785 - val_recall: 0.8922
Epoch 33/50
516/516            1s 2ms/step -
accuracy: 0.9441 - auc: 0.9857 - loss: 0.1370 - precision: 0.8949 - recall:
0.9151 - val_accuracy: 0.9307 - val_auc: 0.9786 - val_loss: 0.1699 -
val_precision: 0.8776 - val_recall: 0.8905
Epoch 34/50
516/516            1s 2ms/step -
accuracy: 0.9448 - auc: 0.9858 - loss: 0.1364 - precision: 0.8964 - recall:
0.9157 - val_accuracy: 0.9307 - val_auc: 0.9786 - val_loss: 0.1702 -
```

```
val_precision: 0.8776 - val_recall: 0.8905
Epoch 35/50
516/516            1s 3ms/step -
accuracy: 0.9452 - auc: 0.9860 - loss: 0.1356 - precision: 0.8976 - recall:
0.9157 - val_accuracy: 0.9307 - val_auc: 0.9786 - val_loss: 0.1703 -
val_precision: 0.8770 - val_recall: 0.8913
Epoch 36/50
516/516            1s 2ms/step -
accuracy: 0.9454 - auc: 0.9861 - loss: 0.1349 - precision: 0.8990 - recall:
0.9148 - val_accuracy: 0.9297 - val_auc: 0.9786 - val_loss: 0.1704 -
val_precision: 0.8742 - val_recall: 0.8913
Epoch 37/50
516/516            1s 1ms/step -
accuracy: 0.9456 - auc: 0.9862 - loss: 0.1343 - precision: 0.8997 - recall:
0.9146 - val_accuracy: 0.9297 - val_auc: 0.9786 - val_loss: 0.1706 -
val_precision: 0.8724 - val_recall: 0.8938
Epoch 38/50
516/516            1s 1ms/step -
accuracy: 0.9459 - auc: 0.9864 - loss: 0.1335 - precision: 0.9009 - recall:
0.9145 - val_accuracy: 0.9295 - val_auc: 0.9786 - val_loss: 0.1708 -
val_precision: 0.8735 - val_recall: 0.8913
Epoch 39/50
516/516            1s 1ms/step -
accuracy: 0.9462 - auc: 0.9866 - loss: 0.1329 - precision: 0.9014 - recall:
0.9148 - val_accuracy: 0.9307 - val_auc: 0.9786 - val_loss: 0.1709 -
val_precision: 0.8752 - val_recall: 0.8938
Epoch 40/50
516/516            1s 1ms/step -
accuracy: 0.9463 - auc: 0.9866 - loss: 0.1324 - precision: 0.9012 - recall:
0.9157 - val_accuracy: 0.9307 - val_auc: 0.9786 - val_loss: 0.1709 -
val_precision: 0.8728 - val_recall: 0.8971
Epoch 41/50
516/516            1s 1ms/step -
accuracy: 0.9472 - auc: 0.9867 - loss: 0.1316 - precision: 0.9034 - recall:
0.9164 - val_accuracy: 0.9307 - val_auc: 0.9787 - val_loss: 0.1711 -
val_precision: 0.8728 - val_recall: 0.8971
Epoch 42/50
516/516            1s 1ms/step -
accuracy: 0.9471 - auc: 0.9868 - loss: 0.1314 - precision: 0.9031 - recall:
0.9164 - val_accuracy: 0.9309 - val_auc: 0.9788 - val_loss: 0.1719 -
val_precision: 0.8753 - val_recall: 0.8946
Epoch 43/50
516/516            1s 1ms/step -
accuracy: 0.9475 - auc: 0.9868 - loss: 0.1309 - precision: 0.9036 - recall:
0.9171 - val_accuracy: 0.9314 - val_auc: 0.9788 - val_loss: 0.1721 -
val_precision: 0.8761 - val_recall: 0.8954
Epoch 44/50
516/516            1s 1ms/step -
```

```
accuracy: 0.9475 - auc: 0.9869 - loss: 0.1303 - precision: 0.9041 - recall:
0.9167 - val_accuracy: 0.9319 - val_auc: 0.9786 - val_loss: 0.1720 -
val_precision: 0.8769 - val_recall: 0.8962
Epoch 45/50
516/516            1s 962us/step -
accuracy: 0.9478 - auc: 0.9870 - loss: 0.1297 - precision: 0.9036 - recall:
0.9182 - val_accuracy: 0.9317 - val_auc: 0.9785 - val_loss: 0.1722 -
val_precision: 0.8762 - val_recall: 0.8962
Epoch 46/50
516/516            1s 986us/step -
accuracy: 0.9479 - auc: 0.9870 - loss: 0.1294 - precision: 0.9048 - recall:
0.9171 - val_accuracy: 0.9324 - val_auc: 0.9786 - val_loss: 0.1724 -
val_precision: 0.8783 - val_recall: 0.8962
Epoch 47/50
516/516            1s 1ms/step -
accuracy: 0.9479 - auc: 0.9871 - loss: 0.1289 - precision: 0.9041 - recall:
0.9181 - val_accuracy: 0.9314 - val_auc: 0.9785 - val_loss: 0.1727 -
val_precision: 0.8773 - val_recall: 0.8938
Epoch 48/50
516/516            1s 1ms/step -
accuracy: 0.9479 - auc: 0.9872 - loss: 0.1285 - precision: 0.9048 - recall:
0.9171 - val_accuracy: 0.9321 - val_auc: 0.9782 - val_loss: 0.1731 -
val_precision: 0.8782 - val_recall: 0.8954
Epoch 49/50
516/516            1s 1ms/step -
accuracy: 0.9480 - auc: 0.9873 - loss: 0.1280 - precision: 0.9045 - recall:
0.9180 - val_accuracy: 0.9314 - val_auc: 0.9782 - val_loss: 0.1734 -
val_precision: 0.8773 - val_recall: 0.8938
Epoch 50/50
516/516            1s 1ms/step -
accuracy: 0.9480 - auc: 0.9873 - loss: 0.1276 - precision: 0.9053 - recall:
0.9170 - val_accuracy: 0.9314 - val_auc: 0.9782 - val_loss: 0.1735 -
val_precision: 0.8773 - val_recall: 0.8938
```

[4]:

```python
# Hyperparameter tuning code for binary classifier
# configs = [
#     {"layers": [32], "activation": "relu"},
#     {"layers": [64], "activation": "relu"},
#     {"layers": [32], "activation": "tanh"},
#     {"layers": [64], "activation": "tanh"},
#     {"layers": [32], "activation": "selu"},
#     {"layers": [32], "activation": "sigmoid"},
# ]

# configs = [
#     {"layers": [16], "activation": "relu"},
#     {"layers": [32], "activation": "relu"},
#     {"layers": [64], "activation": "relu"},
#     {"layers": [128], "activation": "relu"},
#     {"layers": [256], "activation": "relu"},
#     {"layers": [32, 16], "activation": "relu"},
#     {"layers": [64, 32], "activation": "relu"},
#     {"layers": [128, 64], "activation": "relu"},
#     {"layers": [64, 32, 16], "activation": "relu"},
# ]

# Added tuning for optimizers and learning rates NOTE: Not sure if we should tune # of epochs as well (50,100,200)
configs = [
    {"layers": [16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [32], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [32], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [64], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [64], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [128], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [128], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [256], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [256], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [32, 16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [32, 16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [64, 32], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [64, 32], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [128, 64], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [128, 64], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},
    {"layers": [64, 32, 16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.001},
    {"layers": [64, 32, 16], "activation": "relu", "optimizer": "adam", "learning_rate": 0.01},

    {"layers": [16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [32], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [32], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [64], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [64], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [128], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [128], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [256], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [256], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [32, 16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [32, 16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [64, 32], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [64, 32], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [128, 64], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [128, 64], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01},
    {"layers": [64, 32, 16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.001},
    {"layers": [64, 32, 16], "activation": "relu", "optimizer": "sgd", "learning_rate": 0.01}
]

results = []
for config in configs:
    print(f"Trying config: {config}")
    tf.keras.backend.clear_session()
    if config["optimizer"] == "adam":
        optimizer = tf.keras.optimizers.Adam(config["learning_rate"])
    elif config["optimizer"] == "sgd":
        optimizer = tf.keras.optimizers.SGD(learning_rate=config["learning_rate"])

    # Build model with current configuration
    input_layer = tf.keras.layers.Input(shape=(X_train.shape[1],))
    x = input_layer
    for units in config["layers"]:
        x = tf.keras.layers.Dense(units, activation=config["activation"])(x)

    # Binary classification output
    output = tf.keras.layers.Dense(1, activation="sigmoid")(x)

    model = tf.keras.models.Model(inputs=input_layer, outputs=output)
    model.compile(
        loss="binary_crossentropy",
        optimizer=optimizer,
        metrics=["accuracy", tf.keras.metrics.AUC()]
    )
    model.summary()

    # Train model
    history = model.fit(
        X_train, y_train,
        epochs=50,
        validation_data=(X_test, y_test),
        verbose=0
    )

    # Get best results based on validation accuracy
    val_acc_hist = history.history["val_accuracy"]
    best_val_acc = np.max(val_acc_hist)
    best_epoch = np.argmax(val_acc_hist)

    best_loss = history.history["loss"][best_epoch]
    best_acc = history.history["accuracy"][best_epoch]
    best_val_loss = history.history["val_loss"][best_epoch]
    best_auc = history.history["auc"][best_epoch]
    best_val_auc = history.history["val_auc"][best_epoch]

    print(f"Best epoch: {best_epoch+1} | loss: {best_loss:.4f} - acc: {best_acc:.4f} - auc: {best_auc:.4f} - val_loss: {best_val_loss:.4f} - val_acc: {best_val_acc:.4f} - val_auc: {best_val_auc:.4f}")

    # Store results
    results.append({
        "config": config,
        "epoch": best_epoch+1,
        "loss": best_loss,
        "accuracy": best_acc,
        "auc": best_auc,
        "val_loss": best_val_loss,
        "val_accuracy": best_val_acc,
        "val_auc": best_val_auc,
        "history": history.history
    })

# Display results table
df_results = pd.DataFrame(results)
display(df_results)
```

```
Trying config: {'layers': [16], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 16)                      1,072

 dense_1 (Dense)                   (None, 1)                          17
```

```
 Total params: 1,089 (4.25 KB)

 Trainable params: 1,089 (4.25 KB)

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

```
Best epoch: 48 | loss: 0.1457 - acc: 0.9390 - auc: 0.9841 - val_loss: 0.1755 -
val_acc: 0.9290 - val_auc: 0.9780
Trying config: {'layers': [16], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #
 input_layer (InputLayer)          (None, 66)                         0

 dense (Dense)                     (None, 16)                     1,072

 dense_1 (Dense)                   (None, 1)                         17


 Total params: 1,089 (4.25 KB)

 Trainable params: 1,089 (4.25 KB)

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

```
Best epoch: 26 | loss: 0.1446 - acc: 0.9403 - auc: 0.9842 - val_loss: 0.1823 -
val_acc: 0.9290 - val_auc: 0.9757
Trying config: {'layers': [32], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                      Output Shape                 Param #

 input_layer (InputLayer)          (None, 66)                         0

 dense (Dense)                     (None, 32)                     2,144

 dense_1 (Dense)                   (None, 1)                         33


 Total params: 2,177 (8.50 KB)

 Trainable params: 2,177 (8.50 KB)

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

```
Best epoch: 29 | loss: 0.1366 - acc: 0.9436 - auc: 0.9861 - val_loss: 0.1698 -
val_acc: 0.9358 - val_auc: 0.9786
Trying config: {'layers': [32], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 32)                      2,144

 dense_1 (Dense)                   (None, 1)                          33


 Total params: 2,177 (8.50 KB)

 Trainable params: 2,177 (8.50 KB)

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

```
Best epoch: 7 | loss: 0.1625 - acc: 0.9321 - auc: 0.9803 - val_loss: 0.1864 -
val_acc: 0.9241 - val_auc: 0.9757
Trying config: {'layers': [64], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 64)                      4,288

 dense_1 (Dense)                   (None, 1)                          65


 Total params: 4,353 (17.00 KB)

 Trainable params: 4,353 (17.00 KB)

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

```
Best epoch: 17 | loss: 0.1419 - acc: 0.9416 - auc: 0.9850 - val_loss: 0.1729 -
val_acc: 0.9270 - val_auc: 0.9783
Trying config: {'layers': [64], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 64)                      4,288

 dense_1 (Dense)                  (None, 1)                          65


 Total params: 4,353 (17.00 KB)

 Trainable params: 4,353 (17.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 13 | loss: 0.1382 - acc: 0.9429 - auc: 0.9855 - val_loss: 0.1894 -
val_acc: 0.9266 - val_auc: 0.9750
Trying config: {'layers': [128], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 128)                     8,576

 dense_1 (Dense)                  (None, 1)                         129


 Total params: 8,705 (34.00 KB)

 Trainable params: 8,705 (34.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 14 | loss: 0.1336 - acc: 0.9459 - auc: 0.9867 - val_loss: 0.1798 -
val_acc: 0.9280 - val_auc: 0.9766
```

```
Trying config: {'layers': [128], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                      Output Shape                Param #

 input_layer (InputLayer)          (None, 66)                        0

 dense (Dense)                     (None, 128)                   8,576

 dense_1 (Dense)                   (None, 1)                       129


 Total params: 8,705 (34.00 KB)

 Trainable params: 8,705 (34.00 KB)

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

```
Best epoch: 3 | loss: 0.1785 - acc: 0.9246 - auc: 0.9759 - val_loss: 0.1859 -
val_acc: 0.9237 - val_auc: 0.9738
Trying config: {'layers': [256], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                      Output Shape                Param #

 input_layer (InputLayer)          (None, 66)                        0

 dense (Dense)                     (None, 256)                  17,152

 dense_1 (Dense)                   (None, 1)                       257


 Total params: 17,409 (68.00 KB)

 Trainable params: 17,409 (68.00 KB)

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

```
Best epoch: 10 | loss: 0.1372 - acc: 0.9443 - auc: 0.9860 - val_loss: 0.1798 -
val_acc: 0.9278 - val_auc: 0.9770
Trying config: {'layers': [256], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                      Output Shape                     Param #

 input_layer (InputLayer)          (None, 66)                             0

 dense (Dense)                     (None, 256)                       17,152

 dense_1 (Dense)                   (None, 1)                            257


 Total params: 17,409 (68.00 KB)

 Trainable params: 17,409 (68.00 KB)

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

```
Best epoch: 2 | loss: 0.1904 - acc: 0.9196 - auc: 0.9727 - val_loss: 0.1927 -
val_acc: 0.9220 - val_auc: 0.9727
Trying config: {'layers': [32, 16], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                      Output Shape                     Param #

 input_layer (InputLayer)          (None, 66)                             0

 dense (Dense)                     (None, 32)                         2,144

 dense_1 (Dense)                   (None, 16)                           528

 dense_2 (Dense)                   (None, 1)                             17


 Total params: 2,689 (10.50 KB)
 Trainable params: 2,689 (10.50 KB)

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

```
Best epoch: 21 | loss: 0.1366 - acc: 0.9442 - auc: 0.9860 - val_loss: 0.1899 -
val_acc: 0.9261 - val_auc: 0.9751
Trying config: {'layers': [32, 16], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 32)                      2,144

 dense_1 (Dense)                   (None, 16)                        528

 dense_2 (Dense)                   (None, 1)                          17
```

```
 Total params: 2,689 (10.50 KB)

 Trainable params: 2,689 (10.50 KB)

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

```
Best epoch: 19 | loss: 0.1424 - acc: 0.9418 - auc: 0.9846 - val_loss: 0.2017 -
val_acc: 0.9268 - val_auc: 0.9733
Trying config: {'layers': [64, 32], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 64)                      4,288

 dense_1 (Dense)                   (None, 32)                      2,080
 dense_2 (Dense)                     (None, 1)                              33


 Total params: 6,401 (25.00 KB)

 Trainable params: 6,401 (25.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 8 | loss: 0.1559 - acc: 0.9361 - auc: 0.9817 - val_loss: 0.1915 -
val_acc: 0.9234 - val_auc: 0.9736
Trying config: {'layers': [64, 32], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
Model: "functional"


 Layer (type)                        Output Shape                     Param #

 input_layer (InputLayer)            (None, 66)                             0

 dense (Dense)                       (None, 64)                         4,288

 dense_1 (Dense)                     (None, 32)                         2,080

 dense_2 (Dense)                     (None, 1)                             33


 Total params: 6,401 (25.00 KB)

 Trainable params: 6,401 (25.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 12 | loss: 0.1428 - acc: 0.9393 - auc: 0.9845 - val_loss: 0.2173 -
val_acc: 0.9270 - val_auc: 0.9709
Trying config: {'layers': [128, 64], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
Model: "functional"


 Layer (type)                        Output Shape                     Param #
 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 128)                     8,576

 dense_1 (Dense)                   (None, 64)                      8,256

 dense_2 (Dense)                   (None, 1)                          65


Total params: 16,897 (66.00 KB)

Trainable params: 16,897 (66.00 KB)

Non-trainable params: 0 (0.00 B)

Best epoch: 8 | loss: 0.1359 - acc: 0.9435 - auc: 0.9858 - val_loss: 0.1991 -
val_acc: 0.9198 - val_auc: 0.9739
Trying config: {'layers': [128, 64], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.01}
Model: "functional"


 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 128)                     8,576

 dense_1 (Dense)                   (None, 64)                      8,256

 dense_2 (Dense)                   (None, 1)                          65


Total params: 16,897 (66.00 KB)

Trainable params: 16,897 (66.00 KB)

Non-trainable params: 0 (0.00 B)

Best epoch: 5 | loss: 0.1676 - acc: 0.9304 - auc: 0.9789 - val_loss: 0.2095 -
val_acc: 0.9232 - val_auc: 0.9701
Trying config: {'layers': [64, 32, 16], 'activation': 'relu', 'optimizer':
'adam', 'learning_rate': 0.001}
```

```
Model: "functional"

 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 64)                      4,288

 dense_1 (Dense)                  (None, 32)                      2,080

 dense_2 (Dense)                  (None, 16)                        528

 dense_3 (Dense)                  (None, 1)                          17


 Total params: 6,913 (27.00 KB)

 Trainable params: 6,913 (27.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 11 | loss: 0.1359 - acc: 0.9445 - auc: 0.9862 - val_loss: 0.1972 -
val_acc: 0.9200 - val_auc: 0.9749
Trying config: {'layers': [64, 32, 16], 'activation': 'relu', 'optimizer':
'adam', 'learning_rate': 0.01}
Model: "functional"

 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 64)                      4,288

 dense_1 (Dense)                  (None, 32)                      2,080

 dense_2 (Dense)                  (None, 16)                        528

 dense_3 (Dense)                  (None, 1)                          17


 Total params: 6,913 (27.00 KB)
 Trainable params: 6,913 (27.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 13 | loss: 0.1410 - acc: 0.9402 - auc: 0.9850 - val_loss: 0.2076 -
val_acc: 0.9275 - val_auc: 0.9727
Trying config: {'layers': [16], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}

Model: "functional"


  Layer (type)                     Output Shape                  Param #

  input_layer (InputLayer)         (None, 66)                          0

  dense (Dense)                    (None, 16)                      1,072

  dense_1 (Dense)                  (None, 1)                          17


 Total params: 1,089 (4.25 KB)

 Trainable params: 1,089 (4.25 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 49 | loss: 0.1993 - acc: 0.9187 - auc: 0.9713 - val_loss: 0.2016 -
val_acc: 0.9195 - val_auc: 0.9707
Trying config: {'layers': [16], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}

Model: "functional"


  Layer (type)                     Output Shape                  Param #

  input_layer (InputLayer)         (None, 66)                          0

  dense (Dense)                    (None, 16)                      1,072

  dense_1 (Dense)                  (None, 1)                          17
 Total params: 1,089 (4.25 KB)

 Trainable params: 1,089 (4.25 KB)

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

```
Best epoch: 47 | loss: 0.1729 - acc: 0.9282 - auc: 0.9779 - val_loss: 0.1809 -
val_acc: 0.9237 - val_auc: 0.9757
Trying config: {'layers': [32], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 32)                      2,144

 dense_1 (Dense)                   (None, 1)                          33
```

```
 Total params: 2,177 (8.50 KB)

 Trainable params: 2,177 (8.50 KB)

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

```
Best epoch: 50 | loss: 0.1986 - acc: 0.9201 - auc: 0.9715 - val_loss: 0.2029 -
val_acc: 0.9169 - val_auc: 0.9706
Trying config: {'layers': [32], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"
```

```
 Layer (type)                      Output Shape                  Param #

 input_layer (InputLayer)          (None, 66)                          0

 dense (Dense)                     (None, 32)                      2,144

 dense_1 (Dense)                   (None, 1)                          33
 Total params: 2,177 (8.50 KB)

 Trainable params: 2,177 (8.50 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 47 | loss: 0.1699 - acc: 0.9290 - auc: 0.9784 - val_loss: 0.1794 -
val_acc: 0.9239 - val_auc: 0.9767
Trying config: {'layers': [64], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
Model: "functional"


  Layer (type)                      Output Shape                   Param #

  input_layer (InputLayer)          (None, 66)                           0

  dense (Dense)                     (None, 64)                       4,288

  dense_1 (Dense)                   (None, 1)                           65


 Total params: 4,353 (17.00 KB)

 Trainable params: 4,353 (17.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 50 | loss: 0.1983 - acc: 0.9188 - auc: 0.9718 - val_loss: 0.2037 -
val_acc: 0.9157 - val_auc: 0.9704
Trying config: {'layers': [64], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
Model: "functional"


  Layer (type)                      Output Shape                   Param #

  input_layer (InputLayer)          (None, 66)                           0

  dense (Dense)                     (None, 64)                       4,288
 dense_1 (Dense)                  (None, 1)                         65


 Total params: 4,353 (17.00 KB)

 Trainable params: 4,353 (17.00 KB)

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

```
Best epoch: 16 | loss: 0.1829 - acc: 0.9245 - auc: 0.9755 - val_loss: 0.1870 -
val_acc: 0.9217 - val_auc: 0.9747
Trying config: {'layers': [128], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 128)                     8,576

 dense_1 (Dense)                  (None, 1)                         129


 Total params: 8,705 (34.00 KB)

 Trainable params: 8,705 (34.00 KB)

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

```
Best epoch: 40 | loss: 0.2021 - acc: 0.9190 - auc: 0.9710 - val_loss: 0.2058 -
val_acc: 0.9178 - val_auc: 0.9702
Trying config: {'layers': [128], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0
 dense (Dense)                      (None, 128)                        8,576

 dense_1 (Dense)                    (None, 1)                            129


 Total params: 8,705 (34.00 KB)

 Trainable params: 8,705 (34.00 KB)

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

```
Best epoch: 50 | loss: 0.1677 - acc: 0.9306 - auc: 0.9791 - val_loss: 0.1777 -
val_acc: 0.9244 - val_auc: 0.9767
Trying config: {'layers': [256], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                       Output Shape                     Param #

 input_layer (InputLayer)           (None, 66)                             0

 dense (Dense)                      (None, 256)                       17,152

 dense_1 (Dense)                    (None, 1)                            257


 Total params: 17,409 (68.00 KB)

 Trainable params: 17,409 (68.00 KB)

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

```
Best epoch: 48 | loss: 0.1983 - acc: 0.9205 - auc: 0.9720 - val_loss: 0.2021 -
val_acc: 0.9183 - val_auc: 0.9711
Trying config: {'layers': [256], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                       Output Shape                     Param #
 input_layer (InputLayer)         (None, 66)                         0

 dense (Dense)                    (None, 256)                   17,152

 dense_1 (Dense)                  (None, 1)                        257


 Total params: 17,409 (68.00 KB)

 Trainable params: 17,409 (68.00 KB)

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

```
Best epoch: 48 | loss: 0.1671 - acc: 0.9315 - auc: 0.9793 - val_loss: 0.1763 -
val_acc: 0.9256 - val_auc: 0.9770
Trying config: {'layers': [32, 16], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"
```

```
 Layer (type)                     Output Shape                 Param #

 input_layer (InputLayer)         (None, 66)                         0

 dense (Dense)                    (None, 32)                     2,144

 dense_1 (Dense)                  (None, 16)                       528

 dense_2 (Dense)                  (None, 1)                         17


 Total params: 2,689 (10.50 KB)

 Trainable params: 2,689 (10.50 KB)

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

```
Best epoch: 50 | loss: 0.1918 - acc: 0.9219 - auc: 0.9729 - val_loss: 0.1960 -
val_acc: 0.9186 - val_auc: 0.9719
Trying config: {'layers': [32, 16], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 32)                      2,144

 dense_1 (Dense)                  (None, 16)                        528

 dense_2 (Dense)                  (None, 1)                          17


 Total params: 2,689 (10.50 KB)

 Trainable params: 2,689 (10.50 KB)

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

```
Best epoch: 45 | loss: 0.1698 - acc: 0.9295 - auc: 0.9786 - val_loss: 0.1787 -
val_acc: 0.9258 - val_auc: 0.9763
Trying config: {'layers': [64, 32], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"


 Layer (type)                     Output Shape                  Param #

 input_layer (InputLayer)         (None, 66)                          0

 dense (Dense)                    (None, 64)                      4,288

 dense_1 (Dense)                  (None, 32)                      2,080

 dense_2 (Dense)                  (None, 1)                          33


 Total params: 6,401 (25.00 KB)

 Trainable params: 6,401 (25.00 KB)
 Non-trainable params: 0 (0.00 B)

Best epoch: 50 | loss: 0.1889 - acc: 0.9243 - auc: 0.9740 - val_loss: 0.1934 -
val_acc: 0.9205 - val_auc: 0.9729
Trying config: {'layers': [64, 32], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"


  Layer (type)                      Output Shape                  Param #

  input_layer (InputLayer)          (None, 66)                          0

  dense (Dense)                     (None, 64)                      4,288

  dense_1 (Dense)                   (None, 32)                      2,080

  dense_2 (Dense)                   (None, 1)                          33


 Total params: 6,401 (25.00 KB)

 Trainable params: 6,401 (25.00 KB)

 Non-trainable params: 0 (0.00 B)

Best epoch: 50 | loss: 0.1676 - acc: 0.9309 - auc: 0.9789 - val_loss: 0.1765 -
val_acc: 0.9261 - val_auc: 0.9770
Trying config: {'layers': [128, 64], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.001}
```

```
Model: "functional"


  Layer (type)                      Output Shape                  Param #

  input_layer (InputLayer)          (None, 66)                          0

  dense (Dense)                     (None, 128)                     8,576

  dense_1 (Dense)                   (None, 64)                      8,256

  dense_2 (Dense)                   (None, 1)                          65
```

```
 Total params: 16,897 (66.00 KB)

 Trainable params: 16,897 (66.00 KB)

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

```
Best epoch: 41 | loss: 0.1905 - acc: 0.9227 - auc: 0.9736 - val_loss: 0.1938 -
val_acc: 0.9210 - val_auc: 0.9728
Trying config: {'layers': [128, 64], 'activation': 'relu', 'optimizer': 'sgd',
'learning_rate': 0.01}
```

```
Model: "functional"
```

```
  Layer (type)                      Output Shape                  Param #

  input_layer (InputLayer)          (None, 66)                          0

  dense (Dense)                     (None, 128)                     8,576

  dense_1 (Dense)                   (None, 64)                      8,256

  dense_2 (Dense)                   (None, 1)                          65
```

```
 Total params: 16,897 (66.00 KB)

 Trainable params: 16,897 (66.00 KB)

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

```
Best epoch: 44 | loss: 0.1650 - acc: 0.9318 - auc: 0.9798 - val_loss: 0.1778 -
val_acc: 0.9263 - val_auc: 0.9768
Trying config: {'layers': [64, 32, 16], 'activation': 'relu', 'optimizer':
'sgd', 'learning_rate': 0.001}
```

```
Model: "functional"
```

```
  Layer (type)                      Output Shape                  Param #

  input_layer (InputLayer)          (None, 66)                          0
```

```
 dense (Dense)                    (None, 64)                     4,288

 dense_1 (Dense)                  (None, 32)                     2,080

 dense_2 (Dense)                  (None, 16)                       528

 dense_3 (Dense)                  (None, 1)                         17


 Total params: 6,913 (27.00 KB)

 Trainable params: 6,913 (27.00 KB)

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

```
Best epoch: 49 | loss: 0.1819 - acc: 0.9259 - auc: 0.9753 - val_loss: 0.1916 -
val_acc: 0.9227 - val_auc: 0.9728
Trying config: {'layers': [64, 32, 16], 'activation': 'relu', 'optimizer':
'sgd', 'learning_rate': 0.01}
```

```
Model: "functional"


 Layer (type)                     Output Shape                   Param #

 input_layer (InputLayer)         (None, 66)                         0

 dense (Dense)                    (None, 64)                     4,288

 dense_1 (Dense)                  (None, 32)                     2,080

 dense_2 (Dense)                  (None, 16)                       528

 dense_3 (Dense)                  (None, 1)                         17


 Total params: 6,913 (27.00 KB)

 Trainable params: 6,913 (27.00 KB)

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

```
Best epoch: 26 | loss: 0.1735 - acc: 0.9281 - auc: 0.9773 - val_loss: 0.1828 -
val_acc: 0.9268 - val_auc: 0.9751

                                               config  epoch      loss  \
0   {'layers': [16], 'activation': 'relu', 'optimi…     48  0.145703
1   {'layers': [16], 'activation': 'relu', 'optimi…     26  0.144606
2   {'layers': [32], 'activation': 'relu', 'optimi…     29  0.136632
3   {'layers': [32], 'activation': 'relu', 'optimi…      7  0.162516
4   {'layers': [64], 'activation': 'relu', 'optimi…     17  0.141885
5   {'layers': [64], 'activation': 'relu', 'optimi…     13  0.138187
6   {'layers': [128], 'activation': 'relu', 'optim…     14  0.133628
7   {'layers': [128], 'activation': 'relu', 'optim…      3  0.178487
8   {'layers': [256], 'activation': 'relu', 'optim…     10  0.137175
9   {'layers': [256], 'activation': 'relu', 'optim…      2  0.190442
10  {'layers': [32, 16], 'activation': 'relu', 'op…     21  0.136650
11  {'layers': [32, 16], 'activation': 'relu', 'op…     19  0.142415
12  {'layers': [64, 32], 'activation': 'relu', 'op…      8  0.155859
13  {'layers': [64, 32], 'activation': 'relu', 'op…     12  0.142817
14  {'layers': [128, 64], 'activation': 'relu', 'o…      8  0.135938
15  {'layers': [128, 64], 'activation': 'relu', 'o…      5  0.167635
16  {'layers': [64, 32, 16], 'activation': 'relu',…     11  0.135948
17  {'layers': [64, 32, 16], 'activation': 'relu',…     13  0.140965
18  {'layers': [16], 'activation': 'relu', 'optimi…     49  0.199262
19  {'layers': [16], 'activation': 'relu', 'optimi…     47  0.172927
20  {'layers': [32], 'activation': 'relu', 'optimi…     50  0.198580
21  {'layers': [32], 'activation': 'relu', 'optimi…     47  0.169906
22  {'layers': [64], 'activation': 'relu', 'optimi…     50  0.198311
23  {'layers': [64], 'activation': 'relu', 'optimi…     16  0.182866
24  {'layers': [128], 'activation': 'relu', 'optim…     40  0.202101
25  {'layers': [128], 'activation': 'relu', 'optim…     50  0.167731
26  {'layers': [256], 'activation': 'relu', 'optim…     48  0.198265
27  {'layers': [256], 'activation': 'relu', 'optim…     48  0.167084
28  {'layers': [32, 16], 'activation': 'relu', 'op…     50  0.191812
29  {'layers': [32, 16], 'activation': 'relu', 'op…     45  0.169831
30  {'layers': [64, 32], 'activation': 'relu', 'op…     50  0.188867
31  {'layers': [64, 32], 'activation': 'relu', 'op…     50  0.167625
32  {'layers': [128, 64], 'activation': 'relu', 'o…     41  0.190455
33  {'layers': [128, 64], 'activation': 'relu', 'o…     44  0.164975
34  {'layers': [64, 32, 16], 'activation': 'relu',…     49  0.181944
35  {'layers': [64, 32, 16], 'activation': 'relu',…     26  0.173536

    accuracy       auc  val_loss  val_accuracy   val_auc  \
0   0.938973  0.984088  0.175476      0.928987  0.977989
1   0.940307  0.984190  0.182258      0.928987  0.975658
2   0.943579  0.986101  0.169837      0.935773  0.978587
3   0.932125  0.980286  0.186433      0.924140  0.975744
4   0.941579  0.985026  0.172933      0.927048  0.978265
5   0.942852  0.985531  0.189438      0.926563  0.974996
```

```
6   0.945882  0.986685  0.179798      0.928017  0.976630
7   0.924550  0.975948  0.185900      0.923655  0.973831
8   0.944306  0.985970  0.179788      0.927775  0.977044
9   0.919641  0.972728  0.192678      0.921958  0.972683
10  0.944185  0.986018  0.189865      0.926078  0.975147
11  0.941761  0.984597  0.201684      0.926806  0.973281
12  0.936125  0.981721  0.191548      0.923413  0.973571
13  0.939276  0.984505  0.217273      0.927048  0.970852
14  0.943458  0.985766  0.199126      0.919777  0.973882
15  0.930428  0.978888  0.209476      0.923170  0.970119
16  0.944488  0.986185  0.197207      0.920019  0.974929
17  0.940185  0.985007  0.207591      0.927533  0.972676
18  0.918732  0.971263  0.201628      0.919535  0.970741
19  0.928186  0.977872  0.180870      0.923655  0.975706
20  0.920126  0.971518  0.202938      0.916869  0.970563
21  0.929035  0.978423  0.179437      0.923897  0.976679
22  0.918793  0.971820  0.203675      0.915657  0.970432
23  0.924489  0.975464  0.187018      0.921716  0.974744
24  0.918975  0.970987  0.205779      0.917838  0.970169
25  0.930610  0.979097  0.177662      0.924382  0.976730
26  0.920490  0.972012  0.202094      0.918323  0.971065
27  0.931459  0.979304  0.176306      0.925594  0.976984
28  0.921884  0.972946  0.196013      0.918565  0.971909
29  0.929459  0.978557  0.178744      0.925836  0.976333
30  0.924308  0.973959  0.193353      0.920504  0.972890
31  0.930853  0.978914  0.176473      0.926078  0.976985
32  0.922732  0.973636  0.193794      0.920989  0.972752
33  0.931822  0.979785  0.177769      0.926321  0.976790
34  0.925883  0.975257  0.191618      0.922685  0.972846
35  0.928126  0.977341  0.182849      0.926806  0.975107

                                              history
0   {'accuracy': [0.8291012644767761, 0.9138839840…
1   {'accuracy': [0.9055814743041992, 0.9213380813…
2   {'accuracy': [0.8794618248939514, 0.9200048446…
3   {'accuracy': [0.9038240313529968, 0.9213986992…
4   {'accuracy': [0.8819465637207031, 0.9212775230…
5   {'accuracy': [0.9075813293457031, 0.9207926988…
6   {'accuracy': [0.8807345032691956, 0.9217623472…
7   {'accuracy': [0.9073995351791382, 0.9198836684…
8   {'accuracy': [0.9006726741790771, 0.9236409664…
9   {'accuracy': [0.9069147109985352, 0.9196412563…
10  {'accuracy': [0.8709169030189514, 0.9231562018…
11  {'accuracy': [0.9056420922279358, 0.9198836684…
12  {'accuracy': [0.8952184915542603, 0.9225501418…
13  {'accuracy': [0.9057632684707642, 0.9192776083…
14  {'accuracy': [0.9040058255195618, 0.9252772331…
15  {'accuracy': [0.9060056805610657, 0.9195200204…
```

```
16  {'accuracy': [0.9021271467208862, 0.9251560568…
17  {'accuracy': [0.9076419472694397, 0.9190351963…
18  {'accuracy': [0.7759529948234558, 0.8489182591…
19  {'accuracy': [0.8676443696022034, 0.9041270017…
20  {'accuracy': [0.70438152551651, 0.817708015441…
21  {'accuracy': [0.8588570356369019, 0.9015211462…
22  {'accuracy': [0.7271074652671814, 0.8460699319…
23  {'accuracy': [0.8681291937828064, 0.9045512676…
24  {'accuracy': [0.7362583875656128, 0.8397672772…
25  {'accuracy': [0.8671595454216003, 0.9058238863…
26  {'accuracy': [0.7791042923927307, 0.8663111329…
27  {'accuracy': [0.8778255581855774, 0.9060662984…
28  {'accuracy': [0.6964426636695862, 0.7764983773…
29  {'accuracy': [0.8646142482757568, 0.9073389768…
30  {'accuracy': [0.7365614175796509, 0.8351615071…
31  {'accuracy': [0.8837646245956421, 0.9089146256…
32  {'accuracy': [0.7219562530517578, 0.8513423204…
33  {'accuracy': [0.8769771456718445, 0.9100660681…
34  {'accuracy': [0.815647542476654, 0.86419004201…
35  {'accuracy': [0.8501303195953369, 0.9055814743…
```

[5]:

```python
# Select the best model configuration based on validation accuracy
best_idx = df_results["val_accuracy"].idxmax()
best_config = df_results.loc[best_idx, "config"]
print(f"Best configuration: {best_config}")

# Build final model with best configuration
tf.keras.backend.clear_session()
input_layer = tf.keras.layers.Input(shape=(X_train.shape[1],))
x = input_layer
for units in best_config["layers"]:
    x = tf.keras.layers.Dense(units, activation=best_config["activation"])(x)
output = tf.keras.layers.Dense(1, activation="sigmoid")(x)

final_model = tf.keras.models.Model(inputs=input_layer, outputs=output)
final_model.compile(
    loss="binary_crossentropy",
    optimizer="adam",
    metrics=["accuracy", tf.keras.metrics.AUC(), tf.keras.metrics.Precision(),␣
 ↪tf.keras.metrics.Recall()]
)

# Train the final model
final_history = final_model.fit(
    X_train, y_train,
    epochs=50,
    validation_data=(X_test, y_test),
    verbose=1
)
```

```
Best configuration: {'layers': [32], 'activation': 'relu', 'optimizer': 'adam',
'learning_rate': 0.001}
Epoch 1/50
516/516            2s 2ms/step -
accuracy: 0.7497 - auc: 0.7922 - loss: 0.4712 - precision: 0.6238 - recall:
0.4282 - val_accuracy: 0.9123 - val_auc: 0.9675 - val_loss: 0.2193 -
val_precision: 0.8622 - val_recall: 0.8382
Epoch 2/50
516/516            1s 1ms/step -
accuracy: 0.9181 - auc: 0.9716 - loss: 0.2028 - precision: 0.8626 - recall:
0.8542 - val_accuracy: 0.9178 - val_auc: 0.9718 - val_loss: 0.1978 -
val_precision: 0.8672 - val_recall: 0.8538
Epoch 3/50
516/516            1s 955us/step -
accuracy: 0.9225 - auc: 0.9748 - loss: 0.1865 - precision: 0.8670 - recall:
0.8663 - val_accuracy: 0.9205 - val_auc: 0.9733 - val_loss: 0.1916 -
val_precision: 0.8678 - val_recall: 0.8636
Epoch 4/50
516/516            0s 871us/step -
accuracy: 0.9241 - auc: 0.9759 - loss: 0.1811 - precision: 0.8685 - recall:
0.8705 - val_accuracy: 0.9227 - val_auc: 0.9740 - val_loss: 0.1884 -
val_precision: 0.8700 - val_recall: 0.8693
Epoch 5/50
516/516            0s 841us/step -
accuracy: 0.9270 - auc: 0.9766 - loss: 0.1779 - precision: 0.8740 - recall:
0.8748 - val_accuracy: 0.9241 - val_auc: 0.9747 - val_loss: 0.1858 -
val_precision: 0.8712 - val_recall: 0.8734
Epoch 6/50
516/516            1s 1ms/step -
accuracy: 0.9276 - auc: 0.9773 - loss: 0.1750 - precision: 0.8732 - recall:
0.8782 - val_accuracy: 0.9258 - val_auc: 0.9753 - val_loss: 0.1834 -
val_precision: 0.8726 - val_recall: 0.8783
Epoch 7/50
516/516            1s 1ms/step -
accuracy: 0.9290 - auc: 0.9781 - loss: 0.1719 - precision: 0.8743 - recall:
0.8827 - val_accuracy: 0.9258 - val_auc: 0.9758 - val_loss: 0.1813 -
val_precision: 0.8726 - val_recall: 0.8783
Epoch 8/50
516/516            0s 874us/step -
accuracy: 0.9308 - auc: 0.9789 - loss: 0.1688 - precision: 0.8756 - recall:
0.8880 - val_accuracy: 0.9258 - val_auc: 0.9763 - val_loss: 0.1794 -
val_precision: 0.8714 - val_recall: 0.8799
Epoch 9/50
516/516            0s 846us/step -
accuracy: 0.9320 - auc: 0.9797 - loss: 0.1657 - precision: 0.8780 - recall:
```

```
0.8894 - val_accuracy: 0.9266 - val_auc: 0.9767 - val_loss: 0.1780 -
val_precision: 0.8711 - val_recall: 0.8832
Epoch 10/50
516/516            0s 844us/step -
accuracy: 0.9336 - auc: 0.9804 - loss: 0.1630 - precision: 0.8790 - recall:
0.8946 - val_accuracy: 0.9258 - val_auc: 0.9768 - val_loss: 0.1773 -
val_precision: 0.8702 - val_recall: 0.8815
Epoch 11/50
516/516            0s 867us/step -
accuracy: 0.9345 - auc: 0.9809 - loss: 0.1605 - precision: 0.8797 - recall:
0.8971 - val_accuracy: 0.9251 - val_auc: 0.9770 - val_loss: 0.1766 -
val_precision: 0.8698 - val_recall: 0.8791
Epoch 12/50
516/516            1s 1ms/step -
accuracy: 0.9353 - auc: 0.9813 - loss: 0.1583 - precision: 0.8812 - recall:
0.8985 - val_accuracy: 0.9258 - val_auc: 0.9771 - val_loss: 0.1761 -
val_precision: 0.8732 - val_recall: 0.8775
Epoch 13/50
516/516            1s 2ms/step -
accuracy: 0.9364 - auc: 0.9818 - loss: 0.1562 - precision: 0.8839 - recall:
0.8992 - val_accuracy: 0.9258 - val_auc: 0.9773 - val_loss: 0.1756 -
val_precision: 0.8738 - val_recall: 0.8766
Epoch 14/50
516/516            1s 2ms/step -
accuracy: 0.9374 - auc: 0.9822 - loss: 0.1543 - precision: 0.8848 - recall:
0.9018 - val_accuracy: 0.9254 - val_auc: 0.9774 - val_loss: 0.1753 -
val_precision: 0.8742 - val_recall: 0.8742
Epoch 15/50
516/516            1s 2ms/step -
accuracy: 0.9383 - auc: 0.9826 - loss: 0.1526 - precision: 0.8867 - recall:
0.9031 - val_accuracy: 0.9261 - val_auc: 0.9776 - val_loss: 0.1747 -
val_precision: 0.8745 - val_recall: 0.8766
Epoch 16/50
516/516            1s 1ms/step -
accuracy: 0.9390 - auc: 0.9829 - loss: 0.1511 - precision: 0.8875 - recall:
0.9046 - val_accuracy: 0.9270 - val_auc: 0.9777 - val_loss: 0.1747 -
val_precision: 0.8780 - val_recall: 0.8758
Epoch 17/50
516/516            1s 989us/step -
accuracy: 0.9393 - auc: 0.9833 - loss: 0.1496 - precision: 0.8876 - recall:
0.9059 - val_accuracy: 0.9258 - val_auc: 0.9778 - val_loss: 0.1746 -
val_precision: 0.8768 - val_recall: 0.8725
Epoch 18/50
516/516            1s 1ms/step -
accuracy: 0.9404 - auc: 0.9835 - loss: 0.1483 - precision: 0.8901 - recall:
0.9070 - val_accuracy: 0.9261 - val_auc: 0.9777 - val_loss: 0.1743 -
val_precision: 0.8769 - val_recall: 0.8734
Epoch 19/50
```

```
516/516            1s 1ms/step -
accuracy: 0.9410 - auc: 0.9837 - loss: 0.1473 - precision: 0.8911 - recall:
0.9079 - val_accuracy: 0.9268 - val_auc: 0.9778 - val_loss: 0.1740 -
val_precision: 0.8785 - val_recall: 0.8742
Epoch 20/50
516/516            1s 1ms/step -
accuracy: 0.9419 - auc: 0.9840 - loss: 0.1461 - precision: 0.8928 - recall:
0.9093 - val_accuracy: 0.9263 - val_auc: 0.9777 - val_loss: 0.1739 -
val_precision: 0.8770 - val_recall: 0.8742
Epoch 21/50
516/516            1s 1ms/step -
accuracy: 0.9418 - auc: 0.9842 - loss: 0.1450 - precision: 0.8925 - recall:
0.9090 - val_accuracy: 0.9266 - val_auc: 0.9777 - val_loss: 0.1739 -
val_precision: 0.8796 - val_recall: 0.8717
Epoch 22/50
516/516            1s 1ms/step -
accuracy: 0.9430 - auc: 0.9845 - loss: 0.1439 - precision: 0.8941 - recall:
0.9119 - val_accuracy: 0.9261 - val_auc: 0.9775 - val_loss: 0.1737 -
val_precision: 0.8794 - val_recall: 0.8701
Epoch 23/50
516/516            1s 1ms/step -
accuracy: 0.9437 - auc: 0.9847 - loss: 0.1428 - precision: 0.8944 - recall:
0.9142 - val_accuracy: 0.9256 - val_auc: 0.9775 - val_loss: 0.1740 -
val_precision: 0.8786 - val_recall: 0.8693
Epoch 24/50
516/516            1s 1ms/step -
accuracy: 0.9436 - auc: 0.9849 - loss: 0.1418 - precision: 0.8941 - recall:
0.9142 - val_accuracy: 0.9268 - val_auc: 0.9775 - val_loss: 0.1743 -
val_precision: 0.8810 - val_recall: 0.8709
Epoch 25/50
516/516            2s 3ms/step -
accuracy: 0.9434 - auc: 0.9852 - loss: 0.1409 - precision: 0.8940 - recall:
0.9134 - val_accuracy: 0.9261 - val_auc: 0.9774 - val_loss: 0.1747 -
val_precision: 0.8801 - val_recall: 0.8693
Epoch 26/50
516/516            1s 1ms/step -
accuracy: 0.9445 - auc: 0.9853 - loss: 0.1400 - precision: 0.8966 - recall:
0.9145 - val_accuracy: 0.9254 - val_auc: 0.9775 - val_loss: 0.1748 -
val_precision: 0.8798 - val_recall: 0.8668
Epoch 27/50
516/516            1s 990us/step -
accuracy: 0.9442 - auc: 0.9854 - loss: 0.1392 - precision: 0.8967 - recall:
0.9130 - val_accuracy: 0.9254 - val_auc: 0.9774 - val_loss: 0.1751 -
val_precision: 0.8798 - val_recall: 0.8668
Epoch 28/50
516/516            1s 1ms/step -
accuracy: 0.9446 - auc: 0.9856 - loss: 0.1384 - precision: 0.8975 - recall:
0.9139 - val_accuracy: 0.9249 - val_auc: 0.9776 - val_loss: 0.1754 -
```

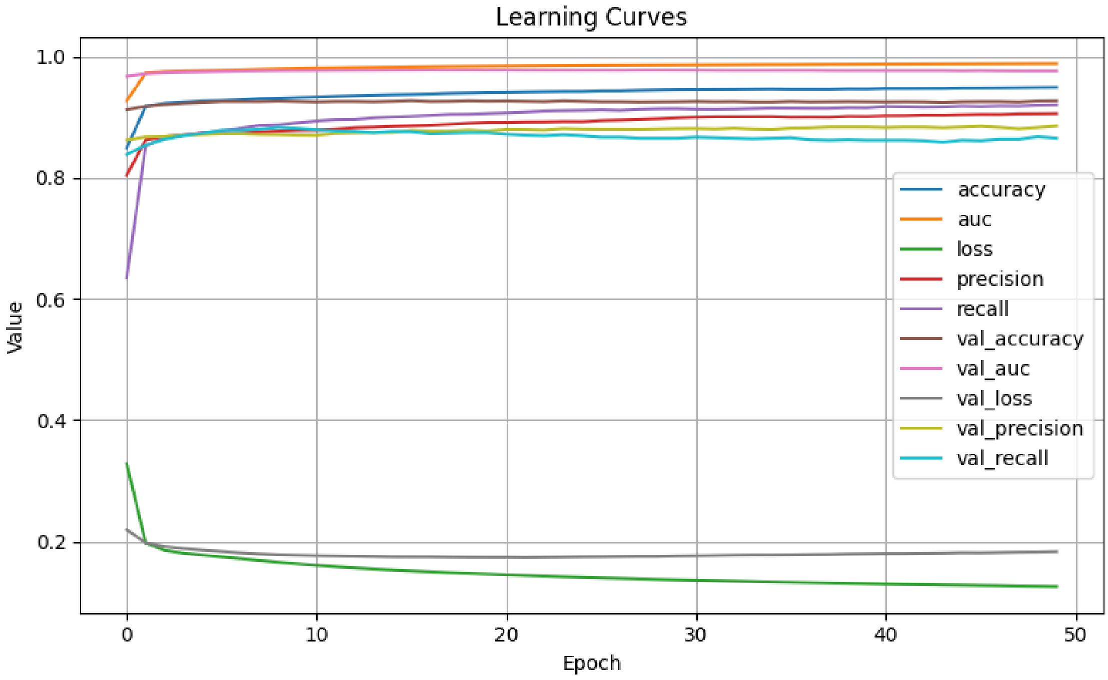
```
val_precision: 0.8796 - val_recall: 0.8652
Epoch 29/50
516/516            1s 1ms/step -
accuracy: 0.9456 - auc: 0.9858 - loss: 0.1375 - precision: 0.8998 - recall:
0.9144 - val_accuracy: 0.9251 - val_auc: 0.9777 - val_loss: 0.1754 -
val_precision: 0.8803 - val_recall: 0.8652
Epoch 30/50
516/516            1s 966us/step -
accuracy: 0.9460 - auc: 0.9859 - loss: 0.1369 - precision: 0.9004 - recall:
0.9155 - val_accuracy: 0.9254 - val_auc: 0.9776 - val_loss: 0.1761 -
val_precision: 0.8810 - val_recall: 0.8652
Epoch 31/50
516/516            1s 991us/step -
accuracy: 0.9460 - auc: 0.9861 - loss: 0.1361 - precision: 0.9012 - recall:
0.9143 - val_accuracy: 0.9258 - val_auc: 0.9775 - val_loss: 0.1764 -
val_precision: 0.8812 - val_recall: 0.8668
Epoch 32/50
516/516            1s 974us/step -
accuracy: 0.9460 - auc: 0.9862 - loss: 0.1354 - precision: 0.9016 - recall:
0.9139 - val_accuracy: 0.9254 - val_auc: 0.9772 - val_loss: 0.1768 -
val_precision: 0.8804 - val_recall: 0.8660
Epoch 33/50
516/516            0s 859us/step -
accuracy: 0.9464 - auc: 0.9863 - loss: 0.1348 - precision: 0.9018 - recall:
0.9150 - val_accuracy: 0.9256 - val_auc: 0.9772 - val_loss: 0.1774 -
val_precision: 0.8818 - val_recall: 0.8652
Epoch 34/50
516/516            0s 844us/step -
accuracy: 0.9466 - auc: 0.9866 - loss: 0.1340 - precision: 0.9015 - recall:
0.9164 - val_accuracy: 0.9249 - val_auc: 0.9772 - val_loss: 0.1777 -
val_precision: 0.8802 - val_recall: 0.8644
Epoch 35/50
516/516            0s 835us/step -
accuracy: 0.9472 - auc: 0.9867 - loss: 0.1333 - precision: 0.9020 - recall:
0.9178 - val_accuracy: 0.9249 - val_auc: 0.9772 - val_loss: 0.1777 -
val_precision: 0.8796 - val_recall: 0.8652
Epoch 36/50
516/516            1s 951us/step -
accuracy: 0.9471 - auc: 0.9868 - loss: 0.1328 - precision: 0.9016 - recall:
0.9180 - val_accuracy: 0.9258 - val_auc: 0.9772 - val_loss: 0.1779 -
val_precision: 0.8819 - val_recall: 0.8660
Epoch 37/50
516/516            1s 948us/step -
accuracy: 0.9466 - auc: 0.9870 - loss: 0.1322 - precision: 0.9012 - recall:
0.9168 - val_accuracy: 0.9251 - val_auc: 0.9772 - val_loss: 0.1783 -
val_precision: 0.8822 - val_recall: 0.8627
Epoch 38/50
516/516            0s 833us/step -
```

```
accuracy: 0.9467 - auc: 0.9871 - loss: 0.1317 - precision: 0.9018 - recall:
0.9165 - val_accuracy: 0.9254 - val_auc: 0.9769 - val_loss: 0.1786 -
val_precision: 0.8836 - val_recall: 0.8619
Epoch 39/50
516/516             1s 1ms/step -
accuracy: 0.9476 - auc: 0.9872 - loss: 0.1312 - precision: 0.9032 - recall:
0.9179 - val_accuracy: 0.9256 - val_auc: 0.9768 - val_loss: 0.1792 -
val_precision: 0.8837 - val_recall: 0.8627
Epoch 40/50
516/516             1s 2ms/step -
accuracy: 0.9472 - auc: 0.9873 - loss: 0.1306 - precision: 0.9021 - recall:
0.9180 - val_accuracy: 0.9254 - val_auc: 0.9768 - val_loss: 0.1795 -
val_precision: 0.8836 - val_recall: 0.8619
Epoch 41/50
516/516             1s 1ms/step -
accuracy: 0.9481 - auc: 0.9874 - loss: 0.1300 - precision: 0.9036 - recall:
0.9196 - val_accuracy: 0.9251 - val_auc: 0.9767 - val_loss: 0.1799 -
val_precision: 0.8828 - val_recall: 0.8619
Epoch 42/50
516/516             1s 1ms/step -
accuracy: 0.9482 - auc: 0.9874 - loss: 0.1295 - precision: 0.9040 - recall:
0.9194 - val_accuracy: 0.9254 - val_auc: 0.9768 - val_loss: 0.1800 -
val_precision: 0.8836 - val_recall: 0.8619
Epoch 43/50
516/516             1s 1ms/step -
accuracy: 0.9482 - auc: 0.9875 - loss: 0.1293 - precision: 0.9048 - recall:
0.9182 - val_accuracy: 0.9251 - val_auc: 0.9768 - val_loss: 0.1804 -
val_precision: 0.8835 - val_recall: 0.8611
Epoch 44/50
516/516             1s 963us/step -
accuracy: 0.9480 - auc: 0.9876 - loss: 0.1286 - precision: 0.9040 - recall:
0.9185 - val_accuracy: 0.9241 - val_auc: 0.9767 - val_loss: 0.1808 -
val_precision: 0.8825 - val_recall: 0.8587
Epoch 45/50
516/516             1s 1ms/step -
accuracy: 0.9483 - auc: 0.9877 - loss: 0.1282 - precision: 0.9042 - recall:
0.9196 - val_accuracy: 0.9254 - val_auc: 0.9763 - val_loss: 0.1815 -
val_precision: 0.8836 - val_recall: 0.8619
Epoch 46/50
516/516             1s 962us/step -
accuracy: 0.9485 - auc: 0.9878 - loss: 0.1276 - precision: 0.9047 - recall:
0.9194 - val_accuracy: 0.9256 - val_auc: 0.9766 - val_loss: 0.1813 -
val_precision: 0.8850 - val_recall: 0.8611
Epoch 47/50
516/516             0s 834us/step -
accuracy: 0.9484 - auc: 0.9879 - loss: 0.1272 - precision: 0.9045 - recall:
0.9194 - val_accuracy: 0.9256 - val_auc: 0.9762 - val_loss: 0.1819 -
val_precision: 0.8830 - val_recall: 0.8636
```

```
Epoch 48/50
516/516             0s 841us/step -
accuracy: 0.9490 - auc: 0.9880 - loss: 0.1266 - precision: 0.9059 - recall:
0.9202 - val_accuracy: 0.9249 - val_auc: 0.9761 - val_loss: 0.1824 -
val_precision: 0.8808 - val_recall: 0.8636
Epoch 49/50
516/516             0s 894us/step -
accuracy: 0.9492 - auc: 0.9880 - loss: 0.1263 - precision: 0.9057 - recall:
0.9211 - val_accuracy: 0.9266 - val_auc: 0.9761 - val_loss: 0.1826 -
val_precision: 0.8828 - val_recall: 0.8676
Epoch 50/50
516/516             1s 1ms/step -
accuracy: 0.9498 - auc: 0.9881 - loss: 0.1260 - precision: 0.9058 - recall:
0.9230 - val_accuracy: 0.9268 - val_auc: 0.9761 - val_loss: 0.1832 -
val_precision: 0.8855 - val_recall: 0.8652
```

[6]:

```python
# Plot learning curves for final model
DataFrame(final_history.history).plot(figsize=(8, 5))
plt.grid(True)
plt.title("Learning Curves")
plt.xlabel("Epoch")
plt.ylabel("Value")
plt.legend()
plt.tight_layout()
plt.show()
```



```
[7]: # Evaluate the final model
     test_loss, test_acc, test_auc, test_precision, test_recall = final_model.
      ↪evaluate(X_test, y_test)

     # Get predictions
     y_pred_prob = final_model.predict(X_test).flatten()
     y_pred = (y_pred_prob > 0.5).astype(int)

     from sklearn.metrics import classification_report, confusion_matrix, roc_curve,␣
      ↪auc

     # Print metrics
     print(f"Test Accuracy: {test_acc:.4f}")
     print(f"Test AUC: {test_auc:.4f}")
     print(f"Test Precision: {test_precision:.4f}")
     print(f"Test Recall: {test_recall:.4f}")

     # Print classification report
     print("\nClassification Report:")
     print(classification_report(y_test, y_pred))

     # Print confusion matrix
     print("\nConfusion Matrix:")
     print(confusion_matrix(y_test, y_pred))

     # Plot ROC curve
     fpr, tpr, _ = roc_curve(y_test, y_pred_prob)
     roc_auc = auc(fpr, tpr)

     plt.figure(figsize=(8, 6))
     plt.plot(fpr, tpr, label=f'ROC Curve (AUC = {roc_auc:.4f})')
     plt.plot([0, 1], [0, 1], 'k--')
     plt.xlim([0.0, 1.0])
     plt.ylim([0.0, 1.05])
     plt.xlabel('False Positive Rate')
     plt.ylabel('True Positive Rate')
     plt.title('Receiver Operating Characteristic')
     plt.legend(loc="lower right")
     plt.grid(True)
     plt.show()
```

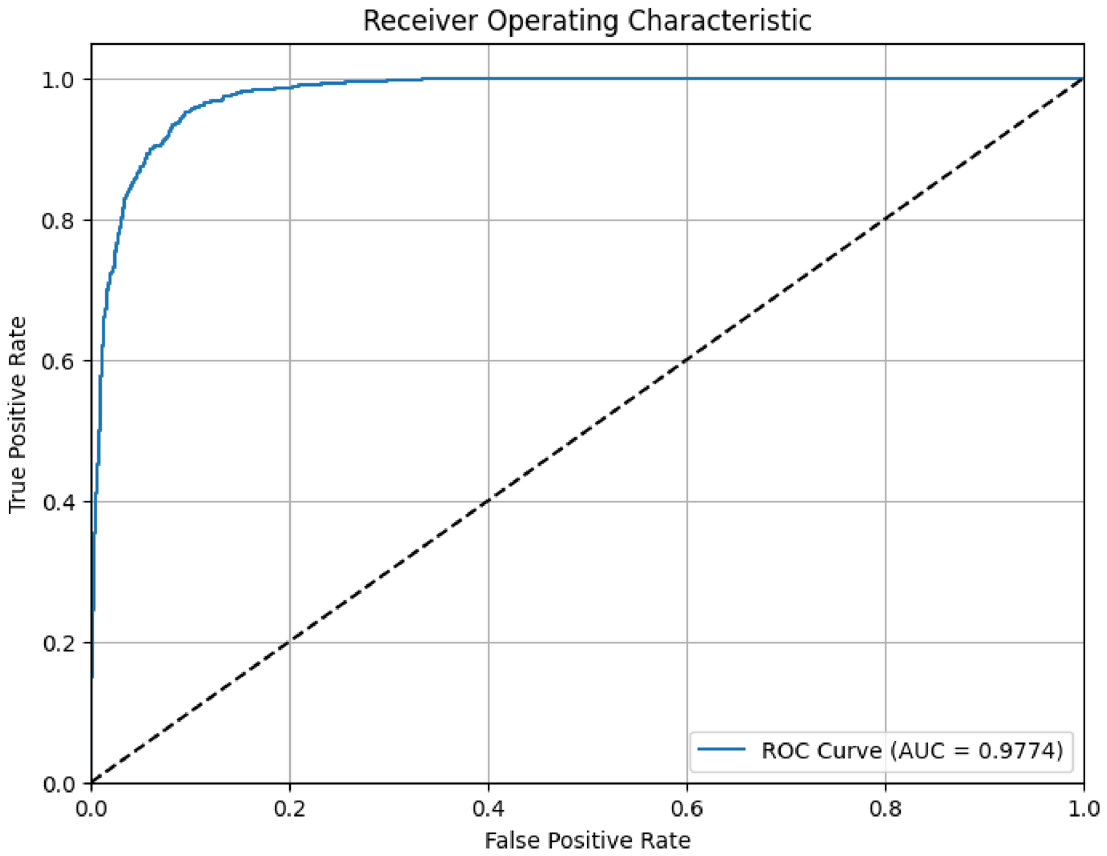
```
129/129              0s 779us/step -
accuracy: 0.9284 - auc: 0.9771 - loss: 0.1778 - precision: 0.8866 - recall:
0.8675
129/129              0s 495us/step
Test Accuracy: 0.9268
Test AUC: 0.9761
Test Precision: 0.8855
Test Recall: 0.8652

Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.95      0.95      2902
           1       0.89      0.87      0.88      1224

    accuracy                           0.93      4126
   macro avg       0.91      0.91      0.91      4126
weighted avg       0.93      0.93      0.93      4126


Confusion Matrix:
[[2765  137]
 [ 165 1059]]
```


Receiver Operating Characteristic
True Positive Rate
False Positive Rate
ROC Curve (AUC = 0.9774)