```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from keras import *
from keras.callbacks import *
import os
from sklearn.preprocessing import MinMaxScaler, RobustScaler
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, f1_score, roc_auc_score,
classification_report
from commons import mean_absolute_percentage_error
from keras.layers import *
from sklearn.pipeline import Pipeline
from keras.utils import to_categorical
from tensorflow.keras.models import load_model
from keras.optimizers import *
from scikeras.wrappers import KerasClassifier

!pip install scikeras
```

```
Collecting scikeras
  Downloading scikeras-0.13.0-py3-none-any.whl.metadata (3.1 kB)
Requirement already satisfied: keras>=3.2.0 in c:\users\vanda\
anaconda3\lib\site-packages (from scikeras) (3.6.0)
Requirement already satisfied: scikit-learn>=1.4.2 in c:\users\vanda\
anaconda3\lib\site-packages (from scikeras) (1.4.2)
Requirement already satisfied: absl-py in c:\users\vanda\anaconda3\
lib\site-packages (from keras>=3.2.0->scikeras) (2.1.0)
Requirement already satisfied: numpy in c:\users\vanda\anaconda3\lib\
site-packages (from keras>=3.2.0->scikeras) (1.26.4)
Requirement already satisfied: rich in c:\users\vanda\anaconda3\lib\
site-packages (from keras>=3.2.0->scikeras) (13.3.5)
Requirement already satisfied: namex in c:\users\vanda\anaconda3\lib\
site-packages (from keras>=3.2.0->scikeras) (0.0.8)
Requirement already satisfied: h5py in c:\users\vanda\anaconda3\lib\
site-packages (from keras>=3.2.0->scikeras) (3.11.0)
Requirement already satisfied: optree in c:\users\vanda\anaconda3\lib\
site-packages (from keras>=3.2.0->scikeras) (0.13.1)
Requirement already satisfied: ml-dtypes in c:\users\vanda\anaconda3\
lib\site-packages (from keras>=3.2.0->scikeras) (0.4.1)
Requirement already satisfied: packaging in c:\users\vanda\anaconda3\
lib\site-packages (from keras>=3.2.0->scikeras) (23.2)
Requirement already satisfied: scipy>=1.6.0 in c:\users\vanda\
anaconda3\lib\site-packages (from scikit-learn>=1.4.2->scikeras)
(1.13.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\vanda\
anaconda3\lib\site-packages (from scikit-learn>=1.4.2->scikeras)
(1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\vanda\
anaconda3\lib\site-packages (from scikit-learn>=1.4.2->scikeras)
```

```
(2.2.0)
Requirement already satisfied: typing-extensions>=4.5.0 in c:\users\
vanda\anaconda3\lib\site-packages (from optree->keras>=3.2.0-
>scikeras) (4.11.0)
Requirement already satisfied: markdown-it-py<3.0.0,>=2.2.0 in c:\
users\vanda\anaconda3\lib\site-packages (from rich->keras>=3.2.0-
>scikeras) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\
vanda\anaconda3\lib\site-packages (from rich->keras>=3.2.0->scikeras)
(2.15.1)
Requirement already satisfied: mdurl~=0.1 in c:\users\vanda\anaconda3\
lib\site-packages (from markdown-it-py<3.0.0,>=2.2.0->rich-
>keras>=3.2.0->scikeras) (0.1.0)
Downloading scikeras-0.13.0-py3-none-any.whl (26 kB)
Installing collected packages: scikeras
Successfully installed scikeras-0.13.0
```

```
np.random.seed(7)

# load dataset
dataframe = pd.read_csv("./datasets/pca_95_cls.csv", sep=',')

dataframe.head(3)
```

```
          0         1         2         3         4         5
6  \
0  0.074162  0.015329 -0.048046  0.042709  0.007321 -0.014251
0.001355
1  0.094841  0.072671 -0.077840 -0.014523  0.027039 -0.053013
0.056817
2  0.064880  0.028643 -0.038454  0.019065  0.028725 -0.014173 -
0.002313

          7         8         9  ...        41        42        43
44  \
0 -0.044263 -0.014403 -0.036199  ...  0.017701 -0.020600 -0.021125 -
0.001148
1 -0.009060  0.047423 -0.009912  ... -0.047544  0.013065  0.065670
0.006482
2 -0.031474 -0.009467 -0.034115  ...  0.020285  0.006481 -0.012896
0.008115

         45        46        47        48        49  priceUSD
0 -0.004502 -0.012360 -0.032049  0.007081  0.006557         1
1  0.020321  0.007130  0.016320  0.013705 -0.042491         1
2 -0.022120 -0.021993  0.012241  0.021045 -0.033730         1

[3 rows x 51 columns]
```

```
dataframe.shape
```

```
(735, 51)

length=dataframe.shape[1]-1

length

50

# split into input (X) and output (Y) variables
X = dataframe.iloc[:,0:length]
y = dataframe['priceUSD']

X.head(3)

          0         1         2         3         4         5
6  \
0  0.074162  0.015329 -0.048046  0.042709  0.007321 -0.014251
0.001355
1  0.094841  0.072671 -0.077840 -0.014523  0.027039 -0.053013
0.056817
2  0.064880  0.028643 -0.038454  0.019065  0.028725 -0.014173 -
0.002313

          7         8         9  ...        40        41        42
43  \
0 -0.044263 -0.014403 -0.036199  ... -0.004087  0.017701 -0.020600 -
0.021125
1 -0.009060  0.047423 -0.009912  ...  0.003421 -0.047544  0.013065
0.065670
2 -0.031474 -0.009467 -0.034115  ...  0.014521  0.020285  0.006481 -
0.012896

         44        45        46        47        48        49
0 -0.001148 -0.004502 -0.012360 -0.032049  0.007081  0.006557
1  0.006482  0.020321  0.007130  0.016320  0.013705 -0.042491
2  0.008115 -0.022120 -0.021993  0.012241  0.021045 -0.033730

[3 rows x 50 columns]

y=np.ravel(y)

y

array([1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 0, 0,
1,
       1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0,
0,
       1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1,
0,
       0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 1, 0,
0,
       0, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0,
1,
        1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0,
0,
        1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1,
1,
        0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0,
1,
        1, 1, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
        1, 1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1,
        1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1,
1,
        1, 0, 1, 1, 1, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1,
0,
        1, 1, 1, 0, 1, 1, 0, 0, 1, 0, 1, 1, 1, 1, 1, 1, 0, 0, 1, 0, 1,
0,
        0, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 1, 1, 0,
0,
        1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0,
0,
        1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0,
1,
        0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 1, 0, 0, 0,
0,
        0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1,
0,
        0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1, 1, 1, 1,
0,
        1, 1, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 0,
0,
        1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 0, 0, 1,
0,
        0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0,
0,
        0, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 1,
1,
        1, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 0, 0, 1,
0,
        0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 1, 1, 0,
0,
        0, 0, 1, 1, 1, 1, 0, 0, 1, 1, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0,
0,
        1, 1, 1, 0, 0, 1, 0, 0, 1, 0, 1, 1, 1, 0, 1, 1, 1, 1, 1, 0, 0,
0,
        1, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 1, 1,
0,
        0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, 0, 0,
0,
       1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 0, 1, 1, 0,
1,
       1, 0, 1, 1, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, 0, 1,
0,
       1, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 1, 1, 0,
1,
       1, 1, 1, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0,
1,
       0, 1, 1, 0, 1, 1, 0, 1, 1], dtype=int64)
```

```
shape=X.shape[1]



X_train,X_test, y_train, y_test = train_test_split(X,y,test_size=0.2,
train_size=0.8, shuffle=False, random_state=7)

estimators=[]

estimators.append(['robust',RobustScaler()])

estimators.append(['mixmax',MinMaxScaler()])

scale=Pipeline(estimators,verbose=True)

scale.fit(X_train)
```

```
[Pipeline] ............ (step 1 of 2) Processing robust, total=   0.0s
[Pipeline] ............ (step 2 of 2) Processing mixmax, total=   0.0s
```

```
Pipeline(steps=[('robust', RobustScaler()), ['mixmax',
MinMaxScaler()]],
         verbose=True)
```

```
X_train=scale.transform(X_train)



X_test=scale.transform(X_test)

def lr_schedule(epoch):
    """Learning Rate Schedule

    Learning rate is scheduled to be reduced after 80, 120, 160, 180
epochs.
    Called automatically every epoch as part of callbacks during
training.

    # Arguments
        epoch (int): The number of epochs

    # Returns
        lr (float32): learning rate
    """
    lr = 1e-2
    if epoch > 180:
        lr *= 0.5e-3
    elif epoch > 160:
        lr *= 1e-3
    elif epoch > 120:
        lr *= 1e-2
    elif epoch > 80:
        lr *= 1e-1
    print('Learning rate: ', lr)
    return lr

def sequential_model(initializer='normal', activation='relu', 
neurons=300, NUM_FEATURES=shape, **kwargs):
    # Create model
    model = Sequential()
    model.add(Input(shape=(NUM_FEATURES,)))  # Set the correct input 
shape
    model.add(Dense(400, kernel_initializer=initializer, 
activation=activation))
    model.add(Dense(500, activation=activation))
    model.add(Dense(100, activation=activation))
    model.add(Dense(1, activation='sigmoid', 
kernel_initializer=initializer))  # Single output neuron with sigmoid

    # Define and compile optimizer
    adam = Adam(learning_rate=lr_schedule(0), amsgrad=True)
    model.compile(loss='binary_crossentropy', optimizer=adam, 
metrics=['accuracy'])

    return model



mcp_save = 
ModelCheckpoint('trained_models/ANN_cls_interval3_pca.keras', 
save_best_only=True, monitor='val_loss', mode='max')
earlyStopping = EarlyStopping(monitor='val_loss', 
patience=100,verbose=1, mode='max')

classifier=KerasClassifier(
    build_fn=sequential_model,batch_size=32, 
epochs=1000,validation_split=0.1,validation_freq=1, 
shuffle=True,use_multiprocessing=True, 
callbacks=[mcp_save,earlyStopping])



classifier.fit(X_train,y_train)
```

```
Learning rate:  0.01
Epoch 1/1000

C:\Users\vanda\anaconda3\Lib\site-packages\scikeras\wrappers.py:925:
UserWarning: ``build_fn`` will be renamed to ``model`` in a future
release, at which point use of ``build_fn`` will raise an Error
instead.
  X, y = self._initialize(X, y)

17/17 ━━━━━━━━━━━━━━━━━━━━ 3s 29ms/step - accuracy: 0.4637 - loss:
0.7409 - val_accuracy: 0.4576 - val_loss: 0.7005
Epoch 2/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 9ms/step - accuracy: 0.5318 - loss:
0.6969 - val_accuracy: 0.4576 - val_loss: 0.7027
Epoch 3/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5393 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.6955
Epoch 4/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5158 - loss:
0.6931 - val_accuracy: 0.4576 - val_loss: 0.6950
Epoch 5/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5263 - loss:
0.6921 - val_accuracy: 0.4576 - val_loss: 0.6982
Epoch 6/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5160 - loss:
0.6928 - val_accuracy: 0.4576 - val_loss: 0.6976
Epoch 7/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5636 - loss:
0.6888 - val_accuracy: 0.4576 - val_loss: 0.6997
Epoch 8/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5441 - loss:
0.6897 - val_accuracy: 0.4576 - val_loss: 0.6987
Epoch 9/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5013 - loss:
0.6943 - val_accuracy: 0.4576 - val_loss: 0.7003
Epoch 10/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5206 - loss:
0.6926 - val_accuracy: 0.4576 - val_loss: 0.6997
Epoch 11/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5559 - loss:
0.6885 - val_accuracy: 0.4576 - val_loss: 0.7017
Epoch 12/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5208 - loss:
0.6926 - val_accuracy: 0.4576 - val_loss: 0.7001
Epoch 13/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5503 - loss:
0.6892 - val_accuracy: 0.4576 - val_loss: 0.7003
Epoch 14/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5036 - loss:
0.6946 - val_accuracy: 0.4576 - val_loss: 0.7008
```

```
Epoch 15/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5378 - loss:
0.6903 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 16/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5428 - loss:
0.6893 - val_accuracy: 0.4576 - val_loss: 0.7019
Epoch 17/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5533 - loss:
0.6884 - val_accuracy: 0.4576 - val_loss: 0.7013
Epoch 18/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5450 - loss:
0.6893 - val_accuracy: 0.4576 - val_loss: 0.6994
Epoch 19/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5203 - loss:
0.6925 - val_accuracy: 0.4576 - val_loss: 0.6987
Epoch 20/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5240 - loss:
0.6921 - val_accuracy: 0.4576 - val_loss: 0.6997
Epoch 21/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - accuracy: 0.5213 - loss:
0.6924 - val_accuracy: 0.4576 - val_loss: 0.7007
Epoch 22/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5387 - loss:
0.6903 - val_accuracy: 0.4576 - val_loss: 0.7006
Epoch 23/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5140 - loss:
0.6935 - val_accuracy: 0.4576 - val_loss: 0.7000
Epoch 24/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5151 - loss:
0.6931 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 25/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5119 - loss:
0.6939 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 26/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5214 - loss:
0.6927 - val_accuracy: 0.4576 - val_loss: 0.7007
Epoch 27/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5346 - loss:
0.6909 - val_accuracy: 0.4576 - val_loss: 0.7019
Epoch 28/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5251 - loss:
0.6922 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 29/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5343 - loss:
0.6909 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 30/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5328 - loss:
0.6911 - val_accuracy: 0.4576 - val_loss: 0.7014
Epoch 31/1000
```

```
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5121 - loss:
0.6940 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 32/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5394 - loss:
0.6902 - val_accuracy: 0.4576 - val_loss: 0.7010
Epoch 33/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5432 - loss:
0.6897 - val_accuracy: 0.4576 - val_loss: 0.7014
Epoch 34/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5371 - loss:
0.6905 - val_accuracy: 0.4576 - val_loss: 0.7015
Epoch 35/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5261 - loss:
0.6920 - val_accuracy: 0.4576 - val_loss: 0.7005
Epoch 36/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5021 - loss:
0.6951 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 37/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5312 - loss:
0.6912 - val_accuracy: 0.4576 - val_loss: 0.7009
Epoch 38/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 11ms/step - accuracy: 0.5364 - loss:
0.6906 - val_accuracy: 0.4576 - val_loss: 0.7014
Epoch 39/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5703 - loss:
0.6860 - val_accuracy: 0.4576 - val_loss: 0.7019
Epoch 40/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5223 - loss:
0.6925 - val_accuracy: 0.4576 - val_loss: 0.7003
Epoch 41/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5369 - loss:
0.6906 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 42/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5431 - loss:
0.6897 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 43/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5336 - loss:
0.6910 - val_accuracy: 0.4576 - val_loss: 0.7016
Epoch 44/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5299 - loss:
0.6915 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 45/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5378 - loss:
0.6905 - val_accuracy: 0.4576 - val_loss: 0.7010
Epoch 46/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5576 - loss:
0.6877 - val_accuracy: 0.4576 - val_loss: 0.7017
Epoch 47/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5473 - loss:
```

```
0.6890 - val_accuracy: 0.4576 - val_loss: 0.7004
Epoch 48/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5187 - loss:
0.6928 - val_accuracy: 0.4576 - val_loss: 0.6990
Epoch 49/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5617 - loss:
0.6881 - val_accuracy: 0.4576 - val_loss: 0.7001
Epoch 50/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5521 - loss:
0.6888 - val_accuracy: 0.4576 - val_loss: 0.7005
Epoch 51/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5176 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.6994
Epoch 52/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 8ms/step - accuracy: 0.5225 - loss:
0.6922 - val_accuracy: 0.4576 - val_loss: 0.7005
Epoch 53/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5611 - loss:
0.6876 - val_accuracy: 0.4576 - val_loss: 0.7004
Epoch 54/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5172 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.6996
Epoch 55/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5607 - loss:
0.6879 - val_accuracy: 0.4576 - val_loss: 0.7016
Epoch 56/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5480 - loss:
0.6889 - val_accuracy: 0.4576 - val_loss: 0.7009
Epoch 57/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.4905 - loss:
0.6968 - val_accuracy: 0.4576 - val_loss: 0.6994
Epoch 58/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5376 - loss:
0.6905 - val_accuracy: 0.4576 - val_loss: 0.7014
Epoch 59/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5181 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.7000
Epoch 60/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5169 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.7008
Epoch 61/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5290 - loss:
0.6916 - val_accuracy: 0.4576 - val_loss: 0.7022
Epoch 62/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5249 - loss:
0.6923 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 63/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5461 - loss:
0.6893 - val_accuracy: 0.4576 - val_loss: 0.7009
```

```
Epoch 64/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5389 - loss:
0.6904 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 65/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5543 - loss:
0.6881 - val_accuracy: 0.4576 - val_loss: 0.7013
Epoch 66/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5176 - loss:
0.6931 - val_accuracy: 0.4576 - val_loss: 0.7003
Epoch 67/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5431 - loss:
0.6899 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 68/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5275 - loss:
0.6919 - val_accuracy: 0.4576 - val_loss: 0.7010
Epoch 69/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5506 - loss:
0.6886 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 70/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5229 - loss:
0.6925 - val_accuracy: 0.4576 - val_loss: 0.7004
Epoch 71/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5487 - loss:
0.6893 - val_accuracy: 0.4576 - val_loss: 0.7010
Epoch 72/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5740 - loss:
0.6856 - val_accuracy: 0.4576 - val_loss: 0.7026
Epoch 73/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.5579 - loss:
0.6873 - val_accuracy: 0.4576 - val_loss: 0.7013
Epoch 74/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5572 - loss:
0.6877 - val_accuracy: 0.4576 - val_loss: 0.7001
Epoch 75/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5641 - loss:
0.6872 - val_accuracy: 0.4576 - val_loss: 0.7007
Epoch 76/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5460 - loss:
0.6894 - val_accuracy: 0.4576 - val_loss: 0.6994
Epoch 77/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5388 - loss:
0.6904 - val_accuracy: 0.4576 - val_loss: 0.6987
Epoch 78/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5143 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.6997
Epoch 79/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5279 - loss:
0.6916 - val_accuracy: 0.4576 - val_loss: 0.6994
Epoch 80/1000
```

```
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5278 - loss:
0.6916 - val_accuracy: 0.4576 - val_loss: 0.7003
Epoch 81/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5140 - loss:
0.6934 - val_accuracy: 0.4576 - val_loss: 0.7002
Epoch 82/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.5334 - loss:
0.6909 - val_accuracy: 0.4576 - val_loss: 0.7014
Epoch 83/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5639 - loss:
0.6868 - val_accuracy: 0.4576 - val_loss: 0.7010
Epoch 84/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5271 - loss:
0.6918 - val_accuracy: 0.4576 - val_loss: 0.7006
Epoch 85/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5348 - loss:
0.6909 - val_accuracy: 0.4576 - val_loss: 0.7007
Epoch 86/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5193 - loss:
0.6928 - val_accuracy: 0.4576 - val_loss: 0.7000
Epoch 87/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5504 - loss:
0.6890 - val_accuracy: 0.4576 - val_loss: 0.7018
Epoch 88/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5321 - loss:
0.6911 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 89/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.5480 - loss:
0.6890 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 90/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5291 - loss:
0.6915 - val_accuracy: 0.4576 - val_loss: 0.6996
Epoch 91/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5205 - loss:
0.6925 - val_accuracy: 0.4576 - val_loss: 0.7001
Epoch 92/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5374 - loss:
0.6906 - val_accuracy: 0.4576 - val_loss: 0.7006
Epoch 93/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5177 - loss:
0.6930 - val_accuracy: 0.4576 - val_loss: 0.7005
Epoch 94/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5218 - loss:
0.6924 - val_accuracy: 0.4576 - val_loss: 0.7018
Epoch 95/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5252 - loss:
0.6921 - val_accuracy: 0.4576 - val_loss: 0.7020
Epoch 96/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 10ms/step - accuracy: 0.5239 - loss:
```

```
0.6924 - val_accuracy: 0.4576 - val_loss: 0.7025
Epoch 97/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5491 - loss:
0.6886 - val_accuracy: 0.4576 - val_loss: 0.7016
Epoch 98/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5413 - loss:
0.6900 - val_accuracy: 0.4576 - val_loss: 0.7013
Epoch 99/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5261 - loss:
0.6919 - val_accuracy: 0.4576 - val_loss: 0.6997
Epoch 100/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5299 - loss:
0.6914 - val_accuracy: 0.4576 - val_loss: 0.7012
Epoch 101/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5115 - loss:
0.6940 - val_accuracy: 0.4576 - val_loss: 0.7007
Epoch 102/1000
17/17 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5529 - loss:
0.6885 - val_accuracy: 0.4576 - val_loss: 0.7011
Epoch 102: early stopping

KerasClassifier(
	model=None
	build_fn=<function sequential_model at 0x00000193D3EC5260>
	warm_start=False
	random_state=None
	optimizer=rmsprop
	loss=None
	metrics=None
	batch_size=32
	validation_batch_size=None
	verbose=1
	callbacks=[<keras.src.callbacks.model_checkpoint.ModelCheckpoint
object at 0x00000193D29476B0>,
<keras.src.callbacks.early_stopping.EarlyStopping object at
0x00000193D283BBC0>]
	validation_split=0.1
	shuffle=True
	run_eagerly=False
	epochs=1000
	validation_freq=1
	use_multiprocessing=True
	class_weight=None
)

prediction_model =
load_model('trained_models/ANN_cls_interval3_pca.keras',compile=False)

y_pred = (prediction_model.predict(X_test) > 0.5).astype("int32")
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 17ms/step
```

```python
acc=accuracy_score(y_test,y_pred)
acc
```

```
0.48299319727891155
```

```python
f1=f1_score(y_test,y_pred,average='weighted')
f1
```

```
0.3146102477688323
```

```python
auc=roc_auc_score(y_test,y_pred)
auc
```

```
0.5
```

```python
y_prob=[prediction_model.predict(X_test).max() for i in
range(len(y_test))]
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 696us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 873us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 392us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 886us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 186us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
```

```
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 873us/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 3ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step
5/5 ━━━━━━━━━━━━━━━━━━━━ 0s 4ms/step

print(classification_report(y_test,y_pred,labels=[0,1],
target_names=['decrease','increase']))

              precision    recall  f1-score   support

    decrease       0.00      0.00      0.00        76
    increase       0.48      1.00      0.65        71

    accuracy                           0.48       147
   macro avg       0.24      0.50      0.33       147
weighted avg       0.23      0.48      0.31       147


C:\Users\vanda\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1509: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
C:\Users\vanda\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1509: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
C:\Users\vanda\anaconda3\Lib\site-packages\sklearn\metrics\
_classification.py:1509: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
  _warn_prf(average, modifier, f"{metric.capitalize()} is",
len(result))
```

```
predictions=pd.DataFrame(zip(np.ravel(y_test),np.ravel(y_pred)),column
s=['y_test','y_pred'])

predictions
```

```
     y_test  y_pred
0         1       1
1         1       1
2         1       1
3         0       1
4         0       1
..      ...     ...
142       1       1
143       1       1
144       0       1
145       1       1
146       1       1

[147 rows x 2 columns]
```