MLP_Stacker

October 7, 2025

1 Ensemble: Simple Averaging & MLP Stacker

1.1 Library Importing

[1]:

```
# Python Standard Libraries
import os
import csv
import math
import random
import unicodedata

# Data Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# NLP - NLTK
import nltk
nltk.download('vader_lexicon')
from nltk.sentiment.vader import SentimentIntensityAnalyzer

# Scikit-learn
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler, StandardScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

# PyTorch
import torch
import torch.nn as nn
from torch.utils.data import Dataset, DataLoader

# TensorFlow / Keras
import tensorflow as tf
from tensorflow.keras.models import Model, Sequential
from tensorflow.keras import layers
from tensorflow.keras.layers import (
    Input, Dense, Dropout, LSTM, Bidirectional,
    Conv1D, Conv2D, MaxPooling1D, MaxPooling2D,
    Flatten, GlobalAveragePooling1D, LayerNormalization,
    MultiHeadAttention, Add, Attention, Permute, Concatenate, Lambda
)
from tensorflow.keras.optimizers import Adam
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau,␣
 ↪ModelCheckpoint
import tensorflow.keras.backend as K
from tensorflow.keras.losses import Huber

# XGBoost
import xgboost as xgb
from xgboost import XGBRegressor

# Shap
import shap
```

```
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /Users/yourth/nltk_data…
[nltk_data]   Package vader_lexicon is already up-to-date!
```

1.2 Data Importing

[2]:

```
company_list = ['TSLA', 'AAPL', 'AMZN', 'GOOGL', 'MSFT']
stock_data_dict = {}

for symbol in company_list:
    path = f"./data/filtered/{symbol}_filtered.csv"
    stock_data_dict[symbol] = pd.read_csv(path)
```

[3]:

```
stock = 'TSLA'
```

1.3 Train Test Split

[97]:

```
def data_integration(stock, n_past=8, n_future=1):
    df = stock_data_dict[stock]

    # 1. Data Lagging
    # Lag technical indicators to avoid leakage
    lag_cols = [
        'Adj Close', 'High', 'Low', 'Volume', 'SMA_5', 'SMA_20',
        'BB_Mid', 'BB_Std', 'BB_Upper', 'BB_Lower',
        'RSI_14', 'Log_Return', 'OBV', 'Vader_Polarity'
    ]

    for col in lag_cols:
        if col == "Adj Close":
```

```
        df[f"{col} (lag1)"] = df[col].shift(1)
    else:
        df[col] = df[col].shift(1)

feature_cols = [
    'Adj Close',  # should be the first one for Y
    'Adj Close (lag1)',
    'SMA_5',              # short-term trend
    'Volume',
    'BB_Mid',           # risk signal
    'Log_Return',
    'DayOfWeek',
    'Month',
    'Vader_Polarity'  # should be the last one for SENTIMENT
]

df = df[feature_cols]

train_size = 0.7
train_split_idx = int(train_size * len(df))

df_filtered = df[feature_cols]
df_filtered = df_filtered.iloc[1:] # delete nan       lag1

# Step 0: Define split boundaries BEFORE scaling
train_df = df_filtered.iloc[:train_split_idx]
test_df  = df_filtered.iloc[train_split_idx:]

# Step 1: Fit scaler only on training data (Avoid Data Leakage)
scaler = MinMaxScaler()
scaler.fit(train_df)

scaler_target = MinMaxScaler()
scaler_target.fit(train_df[['Adj Close']])

# Step 2: Scale training and test data separately
train_scaled = scaler.transform(train_df)
test_scaled  = scaler.transform(test_df)

# Step 3: For inference later, only scale ['Adj Close']
scaler_for_inference = MinMaxScaler()
actual_scaled_close = scaler_for_inference.fit_transform(
    df_filtered[['Adj Close']]
)

# Step 3: Reconstruct sliding windows for train and test
def create_sequences(data, n_past, n_future):
        X, y = [], []
        for i in range(n_past, len(data) - n_future + 1):
            X.append(data[i - n_past:i, 1:])
            y.append(data[i + n_future - 1:i + n_future, [0]])  # Predict Adj␣
 ↪Close
        return np.array(X), np.array(y)

    trainX, trainY = create_sequences(train_scaled, n_past, n_future)
    testX, testY   = create_sequences(test_scaled, n_past, n_future)

    # trainY = trainY.reshape(-1, 1)
    # testY = testY.reshape(-1, 1)

    # Without Sentiment (Baseline Model)
    trainX_wo_tweet = trainX[:, :, :-1]  # Exclude last feature
    testX_wo_tweet  = testX[:, :, :-1]
    trainY_wo_tweet = trainY
    testY_wo_tweet  = testY

    # With Sentiment (Tweet-based Model)
    # trainX_with_tweet = trainX
    # testX_with_tweet  = testX
    # trainY_with_tweet = trainY
    # testY_with_tweet  = testY

    return df, scaler, trainX, trainY, testX, testY, trainX_wo_tweet,␣
 ↪testX_wo_tweet, trainY_wo_tweet, testY_wo_tweet, scaler_for_inference,␣
 ↪scaler_target
```

[98]:

```
df, scaler, trainX, trainY, testX, testY, trainX_wo_tweet, testX_wo_tweet,␣
 ↪trainY_wo_tweet, testY_wo_tweet, scaler_for_inference, scaler_target =␣
 ↪data_integration(stock=stock)
```

[99]:

```
# after you call data_integration(...)
print("Any NaNs in trainX?", np.isnan(trainX).any())
print("Any NaNs in trainY?", np.isnan(trainY).any())
print("Any NaNs in testX? ", np.isnan(testX).any())
```

```
Any NaNs in trainX? True
Any NaNs in trainY? False
Any NaNs in testX?  False
```

[100]:

```
print(f"trainX: {trainX.shape}")
print(f"trainY: {trainY.shape}")
print(f"testX: {testX.shape}")
print(f"testY: {testY.shape}")
print()
print(f"trainX_wo_tweet: {trainX_wo_tweet.shape}")
print(f"testX_wo_tweet: {testX_wo_tweet.shape}")
print(f"trainY_wo_tweet: {trainY_wo_tweet.shape}")
print(f"testY_wo_tweet: {testY_wo_tweet.shape}")
```

```
trainX: (872, 8, 8)
trainY: (872, 1, 1)
testX: (369, 8, 8)
testY: (369, 1, 1)

trainX_wo_tweet: (872, 8, 7)
testX_wo_tweet: (369, 8, 7)
trainY_wo_tweet: (872, 1, 1)
testY_wo_tweet: (369, 1, 1)
```

1.4 CNN-BiLSTM

1.4.1 1. Configuration

[101]:

```
def cnn_biLSTM(input_shape, output_dim):
    inputs = Input(shape=input_shape)

    x = Conv1D(128, kernel_size=2, strides=1, padding='valid')(inputs)
    x = MaxPooling1D(pool_size=2, strides=2)(x)

    x = Conv1D(64, kernel_size=2, strides=1, padding='valid')(x)
    x = MaxPooling1D(pool_size=1, strides=2)(x)

    x = Bidirectional(LSTM(256, return_sequences=True))(x)
    x = Dropout(0.2)(x)
    x = Bidirectional(LSTM(256, return_sequences=True))(x)
    x = Dropout(0.2)(x)

    # === Add attention here ===
    # attn_out = Attention(use_scale=True)([x, x])  # Self-attention: query = value = key = x
    # x = GlobalAveragePooling1D()(attn_out)

    x = Dense(32, activation='relu')(x)
    outputs = Dense(output_dim, activation='relu')(x)

    return Model(inputs, outputs)
```

[102]:

```
# # Build models
cnnBiLSTM_woSent = cnn_biLSTM(
    (trainX_wo_tweet.shape[1], trainX_wo_tweet.shape[2]), trainY.shape[2]
)
cnnBiLSTM_woSent.compile(
    optimizer=Adam(learning_rate=0.001),
    # loss=integrated_loss(delta=0.1, lambda_dir=0.16),  #  adjust as needed
    # loss='mse',  #  adjust as needed
    loss=Huber(0.1),
    metrics=['mae']
)
```

```
[103]: early_stop = EarlyStopping(
           monitor='val_loss',
           patience=20,
           restore_best_weights=True
       )

       reduce_lr = ReduceLROnPlateau(
           monitor='val_loss',
           factor=0.5,         # ← good default
           patience=3,
           min_lr=1e-6,
           verbose=1
       )
```

```
[104]: # after you call data_integration(...)
       print("Any NaNs in trainX?", np.isnan(trainX).any())
       print("Any NaNs in trainY?", np.isnan(trainY).any())
       print("Any NaNs in testX? ", np.isnan(testX).any())
```

```
Any NaNs in trainX? True
Any NaNs in trainY? False
Any NaNs in testX?  False
```

```
[105]: # Fit models
       history_cnnBiLSTM_woSent = cnnBiLSTM_woSent.fit(
           trainX_wo_tweet,
           trainY_wo_tweet,
           epochs=50,
           batch_size=64,
           validation_data=(testX_wo_tweet, testY_wo_tweet),  # ← use your test split␣
        ↪here
           verbose=1,
           callbacks=[early_stop, reduce_lr]
       )
```

```
Epoch 1/50
14/14              2s 28ms/step -
loss: nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 0.0010
Epoch 2/50
14/14              0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 0.0010
```

```
Epoch 3/50
 9/14            0s 6ms/step - loss:
nan - mae: nan
Epoch 3: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.
14/14            0s 10ms/step -
loss: nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 0.0010
Epoch 4/50
14/14            0s 10ms/step -
loss: nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 5.0000e-04
Epoch 5/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 5.0000e-04
Epoch 6/50
 8/14            0s 8ms/step - loss:
nan - mae: nan
Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.
14/14            0s 10ms/step -
loss: nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 5.0000e-04
Epoch 7/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 2.5000e-04
Epoch 8/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 2.5000e-04
Epoch 9/50
 9/14            0s 7ms/step - loss:
nan - mae: nan
Epoch 9: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.
14/14            0s 10ms/step -
loss: nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 2.5000e-04
Epoch 10/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.2500e-04
Epoch 11/50
14/14            0s 8ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.2500e-04
Epoch 12/50
10/14            0s 6ms/step - loss:
nan - mae: nan
Epoch 12: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.2500e-04
Epoch 13/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 6.2500e-05
Epoch 14/50
14/14            0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 6.2500e-05
```

```
Epoch 15/50
10/14           0s 6ms/step - loss:
nan - mae: nan
Epoch 15: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 6.2500e-05
Epoch 16/50
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 3.1250e-05
Epoch 17/50
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 3.1250e-05
Epoch 18/50
10/14           0s 6ms/step - loss:
nan - mae: nan
Epoch 18: ReduceLROnPlateau reducing learning rate to 1.5625000742147677e-05.
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 3.1250e-05
Epoch 19/50
14/14           0s 8ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.5625e-05
Epoch 20/50
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.5625e-05
Epoch 21/50
10/14           0s 6ms/step - loss:
nan - mae: nan
Epoch 21: ReduceLROnPlateau reducing learning rate to 7.812500371073838e-06.
14/14           0s 9ms/step - loss:
nan - mae: nan - val_loss: nan - val_mae: nan - learning_rate: 1.5625e-05
```

1.4.2 2. Tuning

[75]:

```
# # Fit models
# history_cnnBiLSTM_woSent = cnnBiLSTM_woSent.fit(
#     trainX_wo_tweet,
#     trainY_wo_tweet,
#     epochs=50,
#     batch_size=64,
#     validation_data=(testX_wo_tweet, testY_wo_tweet),  # ← use your test split here
#     verbose=1,
#     callbacks=[early_stop, reduce_lr]
# )
```

1.4.3 3. Best Parameter for CNN-BiLSTM

1. Learning Rate: 0.001

2. Huber loss: delta = 0.1
3. Early Stop: patience = 20
4. Reduce LR: factor=0.5, patience=3, min_lr=1e-6,

1.5 Transformer

1.5.1 1. Configuration

[76]:

```
class PositionalEncoding(tf.keras.layers.Layer):
    def __init__(self, sequence_len, d_model):
        super().__init__()
        self.pos_encoding = self.positional_encoding(sequence_len, d_model)

    def get_config(self):
        return {"sequence_len": self.pos_encoding.shape[0], "d_model": self.pos_encoding.shape[1]}

    def positional_encoding(self, position, d_model):
        angle_rads = self.get_angles(np.arange(position)[:, np.newaxis],
                                     np.arange(d_model)[np.newaxis, :],
                                     d_model)
        angle_rads[:, 0::2] = np.sin(angle_rads[:, 0::2])
        angle_rads[:, 1::2] = np.cos(angle_rads[:, 1::2])
        return tf.cast(angle_rads[np.newaxis, ...], dtype=tf.float32)

    def get_angles(self, pos, i, d_model):
        angle_rates = 1 / np.power(10000, (2 * (i // 2)) / np.float32(d_model))
        return pos * angle_rates

    def call(self, x):
        return x + self.pos_encoding[:, :tf.shape(x)[1], :]
```

[77]:

```
# # ========== 1. Transformer Encoder ==========
# def transformer_encoder(inputs, head_size, num_heads, ff_dim, dropout):
#     """
#     Builds a single Transformer encoder block.
#     """
#     # Multi-head self-attention
#     attention_output = MultiHeadAttention(num_heads=num_heads, key_dim=head_size, dropout=dropout)(inputs, inputs)
#     attention_output = Add()([inputs, attention_output])
#     attention_output = LayerNormalization()(attention_output)

#     # Feed-forward network
#     ffn_output = Dense(ff_dim, activation='relu')(attention_output)
#     # ffn_output = Dropout(dropout)(ffn_output)  # <--- Dropout after first FFN layer
#     ffn_output = Dense(inputs.shape[-1])(ffn_output)
#     ffn_output = Add()([attention_output, ffn_output])
#     output = LayerNormalization()(ffn_output)

#     return output

# def build_transformer_model(input_shape, head_size=64, num_heads=4,␣
 ↪ff_dim=128, num_layers=2, dropout=0.1):
#     inputs = Input(shape=input_shape)
#     x = PositionalEncoding(input_shape[0], input_shape[1])(inputs)  # add␣
 ↪positional encoding

#     for _ in range(num_layers):
#         x = transformer_encoder(x, head_size, num_heads, ff_dim, dropout)

#     x = GlobalAveragePooling1D()(x)
#     outputs = Dense(1)(x)
#     return Model(inputs, outputs)
```

[78]:

```
def transformer_encoder(inputs, head_size, num_heads, ff_dim, dropout=0.15,␣
 ↪epsilon=1e-6, kernel_size=1):
    # Pre-LN Self Attention
    x = layers.LayerNormalization(epsilon=epsilon)(inputs)
    x = layers.MultiHeadAttention(
        key_dim=head_size, num_heads=num_heads, dropout=dropout
    )(x, x)
    x = layers.Dropout(dropout)(x)
    x = layers.Add()([x, inputs])

    # Feed-forward block
    y = layers.LayerNormalization(epsilon=epsilon)(x)
    y = layers.Conv1D(filters=ff_dim, kernel_size=kernel_size,␣
 ↪activation='relu')(y)
    y = layers.Dropout(dropout)(y)
    y = layers.Conv1D(filters=inputs.shape[-1], kernel_size=kernel_size)(y)
    return layers.Add()([x, y])

def build_transformer_model(input_shape, head_size=64, num_heads=4, ff_dim=128,
                            num_layers=2, dropout=0.15, mlp_units=[64],␣
 ↪mlp_dropout=0.1):
    inputs = Input(shape=input_shape)
    x = PositionalEncoding(input_shape[0], input_shape[1])(inputs)

    for _ in range(num_layers):
        x = transformer_encoder(x, head_size, num_heads, ff_dim, dropout)

    x = layers.GlobalAveragePooling1D()(x)
    for units in mlp_units:
        x = layers.Dense(units, activation='relu')(x)
        x = layers.Dropout(mlp_dropout)(x)

    outputs = layers.Dense(1)(x)
    return Model(inputs, outputs)
```

1.5.2 2. Tuning

[79]:

```
# # ========== Train model without sentiment ==========
transformer_woSent = build_transformer_model((trainX_wo_tweet.shape[1],␣
 ↪trainX_wo_tweet.shape[2]))
transformer_woSent.compile(
    optimizer=Adam(0.001),
    # loss=integrated_loss(),
    loss=Huber(0.05),
    # loss='mse',
    metrics=['mae']
)
# # history_wo_sent = model_wo_sent.fit(  #   save history here
# #     trainX_wo_tweet, trainY_wo_tweet,
# #     validation_data=(testX_wo_tweet, testY_wo_tweet),
# #     epochs=50, batch_size=64, verbose=1,
# #     callbacks=[early_stop_wo, reduce_lr]
# # )
```

1.5.3 3. Best Parameter for Transformer

1. Learning Rate: 0.001
2. Huber loss: delta = 0.05
3. Early Stop: patience = 15
4. Reduce LR: factor=0.5, patience=3, min_lr=1e-5,

1.6 Ensemble

[80]:

```
early_stop_cnn = EarlyStopping(
    monitor='val_loss',
    patience=20,
    restore_best_weights=True
)

reduce_lr_cnn = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.5,         # ← good default
    patience=3,
    min_lr=1e-6,
    verbose=1
)
```

```
[81]: # For Transformer *without* sentiment (cleaner input, stop sooner)
early_stop_tran = EarlyStopping(
    monitor='val_loss',
    patience=15,
    restore_best_weights=True
)

reduce_lr_tran = ReduceLROnPlateau(
    monitor='val_loss',
    factor=0.5,          # ← good default
    patience=3,
    min_lr=1e-5,
    verbose=1
)
```

```
[82]: # ==================== Train Base Models ====================
# 1. Train CNN+BiLSTM model
cnnBiLSTM_woSent.fit(
    trainX_wo_tweet, trainY_wo_tweet,
    epochs=50,
    batch_size=64,
    validation_data=(testX_wo_tweet, testY_wo_tweet),
    verbose=0,
    callbacks=[early_stop_cnn, reduce_lr_cnn]
)
trainY_pred_cnn = cnnBiLSTM_woSent.predict(trainX_wo_tweet)
# 2. Train Transformer model
transformer_woSent.fit(
    trainX_wo_tweet, trainY_wo_tweet,
    epochs=50,
    batch_size=64,
    validation_data=(testX_wo_tweet, testY_wo_tweet),
    verbose=0,
    callbacks=[early_stop_tran, reduce_lr_tran]
)
trainY_pred_transformer = transformer_woSent.predict(trainX_wo_tweet)
```

```
28/28                 0s 9ms/step

Epoch 6: ReduceLROnPlateau reducing learning rate to 0.0005000000237487257.

Epoch 10: ReduceLROnPlateau reducing learning rate to 0.0002500000118743628.

Epoch 13: ReduceLROnPlateau reducing learning rate to 0.0001250000059371814.

Epoch 18: ReduceLROnPlateau reducing learning rate to 6.25000029685907e-05.

Epoch 21: ReduceLROnPlateau reducing learning rate to 3.125000148429535e-05.
Epoch 27: ReduceLROnPlateau reducing learning rate to 1.5625000742147677e-05.

Epoch 30: ReduceLROnPlateau reducing learning rate to 1e-05.
28/28             0s 5ms/step
```

[83]:

```
print(trainY_pred_cnn.shape)
print(trainY_pred_transformer.shape)
```

```
(875, 1, 1)
(875, 1)
```

[84]:

```
from sklearn.model_selection import TimeSeriesSplit
from sklearn.linear_model import LinearRegression, Ridge
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau

tscv = TimeSeriesSplit(n_splits=5)
meta_features, meta_labels = [], []

for fold, (train_idx, val_idx) in enumerate(tscv.split(trainX_wo_tweet)):
    print(f"\n=== Fold {fold+1} ===")
    X_train, X_val = trainX_wo_tweet[train_idx], trainX_wo_tweet[val_idx]
    y_train, y_val = trainY_wo_tweet[train_idx], trainY_wo_tweet[val_idx]

    # Build and compile CNN model
    cnn_model = cnn_biLSTM((X_train.shape[1], X_train.shape[2]), 1)
    cnn_model.compile(
        optimizer=Adam(learning_rate=0.001),
        loss=Huber(0.1),
        metrics=['mae']
    )

    # Build and compile Transformer model
    transformer_model = build_transformer_model((X_train.shape[1], X_train.
 ↪shape[2]))
    transformer_model.compile(
        optimizer=Adam(0.001),
        # loss=Huber(0.05),
        loss='mse',
        metrics=['mae']
    )

    # Fresh callbacks
    early_stop_cnn = EarlyStopping(monitor='val_loss', patience=20,␣
 ↪restore_best_weights=True)
    reduce_lr_cnn = ReduceLROnPlateau(monitor='val_loss', factor=0.5,␣
 ↪patience=3, min_lr=1e-6, verbose=0)
```

```
    early_stop_tran = EarlyStopping(monitor='val_loss', patience=15,␣
↪restore_best_weights=True)
    reduce_lr_tran = ReduceLROnPlateau(monitor='val_loss', factor=0.5,␣
↪patience=3, min_lr=1e-5, verbose=0)

    # Train models
    cnn_model.fit(X_train, y_train, epochs=50, batch_size=64,
                  validation_data=(X_val, y_val), verbose=0,
                  callbacks=[early_stop_cnn, reduce_lr_cnn])

    transformer_model.fit(X_train, y_train, epochs=50, batch_size=64,
                          validation_data=(X_val, y_val), verbose=0,
                          callbacks=[early_stop_tran, reduce_lr_tran])

    # Predict fold val set
    pred_cnn = cnn_model.predict(X_val).reshape(-1, 1)
    pred_tran = transformer_model.predict(X_val).reshape(-1, 1)

    meta_features.append(np.hstack([pred_cnn, pred_tran]))
    meta_labels.append(y_val)
```

```
=== Fold 1 ===
5/5               0s 49ms/step
5/5               0s 26ms/step

=== Fold 2 ===
5/5               0s 53ms/step
5/5               0s 27ms/step

=== Fold 3 ===
5/5               0s 49ms/step
5/5               0s 28ms/step

=== Fold 4 ===
5/5               0s 50ms/step
5/5               0s 27ms/step

=== Fold 5 ===
5/5               0s 50ms/step
5/5               0s 28ms/step
```

[85]:

```
# Final meta-training data
X_meta_train = np.vstack(meta_features)
y_meta_train = np.concatenate(meta_labels)

# FIX: reshape y to 1D
```

```
if y_meta_train.ndim == 3:
    y_meta_train = y_meta_train.reshape(-1)
elif y_meta_train.ndim == 2:
    y_meta_train = y_meta_train.flatten()

# # Train meta-model (Linear Regression)
# meta_model = LinearRegression()
# meta_model.fit(X_meta_train, y_meta_train)
```

1.6.1 Weighted Averaging Ensemble

[86]:

```
# y_pred_avg = X_meta_train.mean(axis=1)  # (N,) shape
y_pred_avg = 0.9 * X_meta_train[:, 0] + 0.1 * X_meta_train[:, 1]

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

def evaluate_model(name, y_true, y_pred):
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mae = mean_absolute_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"\n {name}")
    print(f"   RMSE: {rmse:.4f}")
    print(f"   MAE : {mae:.4f}")
    print(f"   R²  : {r2:.4f}")

evaluate_model("Simple Averaging Ensemble", y_meta_train, y_pred_avg)
```

```
 Simple Averaging Ensemble
   RMSE: 0.0632
   MAE : 0.0488
   R²  : 0.9317
```

[87]:

```
# Base predictions (scaled or inverse-transformed)
y_pred_cnn = cnnBiLSTM_woSent.predict(testX_wo_tweet).reshape(-1, 1)
y_pred_tran = transformer_woSent.predict(testX_wo_tweet).reshape(-1, 1)

y_pred_cnn = scaler_target.inverse_transform(y_pred_cnn).flatten()
y_pred_tran = scaler_target.inverse_transform(y_pred_tran).flatten()
true_test = scaler_target.inverse_transform(testY_wo_tweet.reshape(-1, 1)).
 ↪flatten()

y_pred_avg = (y_pred_cnn + y_pred_tran) / 2
evaluate_model("Simple Averaging", true_test, y_pred_avg)
```

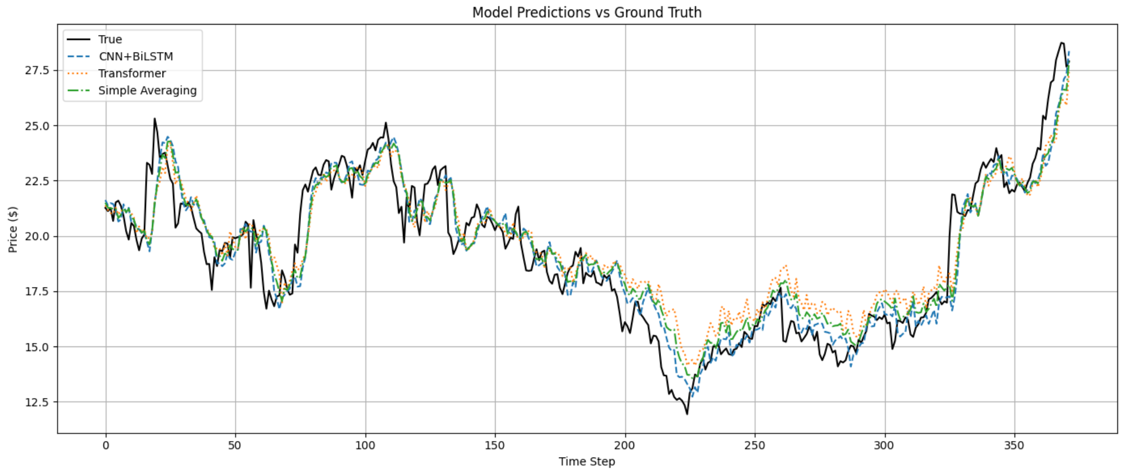
```
12/12                 0s 3ms/step
12/12                 0s 2ms/step
```

```
Simple Averaging
  RMSE: 1.2904
  MAE : 0.9884
  R²  : 0.8580
```

[88]:

```python
#  Evaluate all three models
evaluate_model("CNN+BiLSTM", true_test, y_pred_cnn)
evaluate_model("Transformer", true_test, y_pred_tran)
evaluate_model("Simple Averaging", true_test, y_pred_avg)

plt.figure(figsize=(14, 6))
plt.plot(true_test, label="True", color='black')
plt.plot(y_pred_cnn, label="CNN+BiLSTM", linestyle="--")
plt.plot(y_pred_tran, label="Transformer", linestyle=":")
plt.plot(y_pred_avg, label="Simple Averaging", linestyle="-.")

plt.title("Model Predictions vs Ground Truth")
plt.xlabel("Time Step")
plt.ylabel("Price ($)")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
CNN+BiLSTM
  RMSE: 1.2261
  MAE : 0.9125
  R²  : 0.8718

Transformer
  RMSE: 1.4602
  MAE : 1.1670
  R²  : 0.8181

Simple Averaging
  RMSE: 1.2904
  MAE : 0.9884
  R²  : 0.8580
```



1.6.2 MLP Stack Ensemble

[89]:
```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import uniform, randint
from sklearn.neural_network import MLPRegressor

param_dist = {
    'hidden_layer_sizes': [(32,), (64,), (128,), (64, 32), (128, 64), (128, 64,␣
 ↪32)],
    'activation': ['relu', 'tanh'],
    'alpha': uniform(0.00005, 0.003),  # broader float range
    'learning_rate_init': uniform(0.001, 0.05),
    'solver': ['adam'],
    'max_iter': randint(800, 1500)
}

search = RandomizedSearchCV(
    MLPRegressor(random_state=42),
    param_distributions=param_dist,
    n_iter=60,          # number of random combinations
    scoring='neg_mean_squared_error',
    cv=5,
    n_jobs=-1,
    verbose=2,
    random_state=42
)
search.fit(X_meta_train, y_meta_train)
print("  Best MLP params:", search.best_params_)
```

```
Fitting 5 folds for each of 60 candidates, totalling 300 fits
[CV] END activation=relu, alpha=0.0013874982585607733,
hidden_layer_sizes=(128,), learning_rate_init=0.02396244459829336,
max_iter=1172, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.002439628960580699, hidden_layer_sizes=(128,),
learning_rate_init=0.039984550013638466, max_iter=820, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002439628960580699, hidden_layer_sizes=(128,),
learning_rate_init=0.039984550013638466, max_iter=820, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002439628960580699, hidden_layer_sizes=(128,),
learning_rate_init=0.039984550013638466, max_iter=820, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00047860045376582237,
hidden_layer_sizes=(128,), learning_rate_init=0.0020292247147901225,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0013874982585607733,
hidden_layer_sizes=(128,), learning_rate_init=0.02396244459829336,
max_iter=1172, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0013874982585607733,
hidden_layer_sizes=(128,), learning_rate_init=0.02396244459829336,
max_iter=1172, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0013874982585607733,
hidden_layer_sizes=(128,), learning_rate_init=0.02396244459829336,
max_iter=1172, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.002865658127047251, hidden_layer_sizes=(64,),
learning_rate_init=0.010091248360355031, max_iter=1076, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002439628960580699, hidden_layer_sizes=(128,),
learning_rate_init=0.039984550013638466, max_iter=820, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002865658127047251, hidden_layer_sizes=(64,),
learning_rate_init=0.010091248360355031, max_iter=1076, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002865658127047251, hidden_layer_sizes=(64,),
learning_rate_init=0.010091248360355031, max_iter=1076, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002439628960580699, hidden_layer_sizes=(128,),
learning_rate_init=0.039984550013638466, max_iter=820, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002865658127047251, hidden_layer_sizes=(64,),
learning_rate_init=0.010091248360355031, max_iter=1076, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002865658127047251, hidden_layer_sizes=(64,),
learning_rate_init=0.010091248360355031, max_iter=1076, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0013874982585607733,
hidden_layer_sizes=(128,), learning_rate_init=0.02396244459829336,
max_iter=1172, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00047860045376582237,
hidden_layer_sizes=(128,), learning_rate_init=0.0020292247147901225,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0018855586841671384, hidden_layer_sizes=(64,),
learning_rate_init=0.003332831606807715, max_iter=1499, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0018855586841671384, hidden_layer_sizes=(64,),
learning_rate_init=0.003332831606807715, max_iter=1499, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0018855586841671384, hidden_layer_sizes=(64,),
learning_rate_init=0.003332831606807715, max_iter=1499, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0009627267288786132, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0013533152609858704, max_iter=1360, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.00047860045376582237,
hidden_layer_sizes=(128,), learning_rate_init=0.0020292247147901225,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0018855586841671384, hidden_layer_sizes=(64,),
learning_rate_init=0.003332831606807715, max_iter=1499, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0009627267288786132, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0013533152609858704, max_iter=1360, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0018855586841671384, hidden_layer_sizes=(64,),
learning_rate_init=0.003332831606807715, max_iter=1499, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0009627267288786132, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0013533152609858704, max_iter=1360, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0009627267288786132, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0013533152609858704, max_iter=1360, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.00047860045376582237,
hidden_layer_sizes=(128,), learning_rate_init=0.0020292247147901225,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0009627267288786132, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0013533152609858704, max_iter=1360, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0015927033152408347, hidden_layer_sizes=(32,),
learning_rate_init=0.003322520635999886З, max_iter=1446, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0015927033152408347, hidden_layer_sizes=(32,),
learning_rate_init=0.0033225206359998863, max_iter=1446, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0015927033152408347, hidden_layer_sizes=(32,),
learning_rate_init=0.0033225206359998863, max_iter=1446, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0015927033152408347, hidden_layer_sizes=(32,),
learning_rate_init=0.0033225206359998863, max_iter=1446, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001401497755908629, hidden_layer_sizes=(64,),
learning_rate_init=0.04844427686266667, max_iter=1115, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0015927033152408347, hidden_layer_sizes=(32,),
learning_rate_init=0.0033225206359998863, max_iter=1446, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001401497755908629, hidden_layer_sizes=(64,),
learning_rate_init=0.04844427686266667, max_iter=1115, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.001401497755908629, hidden_layer_sizes=(64,),
learning_rate_init=0.04844427686266667, max_iter=1115, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007483140212909127, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.040258798069650686, max_iter=1362, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0007483140212909127, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.040258798069650686, max_iter=1362, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.001401497755908629, hidden_layer_sizes=(64,),
learning_rate_init=0.04844427686266667, max_iter=1115, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0024751920443493837, hidden_layer_sizes=(32,),
learning_rate_init=0.0017983126110107097, max_iter=1139, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0024751920443493837, hidden_layer_sizes=(32,),
learning_rate_init=0.0017983126110107097, max_iter=1139, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007483140212909127, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.040258798069650686, max_iter=1362, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0024751920443493837, hidden_layer_sizes=(32,),
learning_rate_init=0.0017983126110107097, max_iter=1139, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001401497755908629, hidden_layer_sizes=(64,),
learning_rate_init=0.04844427686266667, max_iter=1115, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0024751920443493837, hidden_layer_sizes=(32,),
learning_rate_init=0.0017983126110107097, max_iter=1139, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007483140212909127, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.040258798069650686, max_iter=1362, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0024751920443493837, hidden_layer_sizes=(32,),
learning_rate_init=0.0017983126110107097, max_iter=1139, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00047860045376582237,
hidden_layer_sizes=(128,), learning_rate_init=0.0020292247147901225,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0012231818227197223, hidden_layer_sizes=(64,),
learning_rate_init=0.0341261142176991, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0013704574812188038, hidden_layer_sizes=(64,
32), learning_rate_init=0.02575884550556351, max_iter=834, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0013704574812188038, hidden_layer_sizes=(64,
32), learning_rate_init=0.02575884550556351, max_iter=834, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0013704574812188038, hidden_layer_sizes=(64,
32), learning_rate_init=0.02575884550556351, max_iter=834, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0012231818227197223, hidden_layer_sizes=(64,),
learning_rate_init=0.0341261142176991, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0012231818227197223, hidden_layer_sizes=(64,),
learning_rate_init=0.0341261142176991, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0012231818227197223, hidden_layer_sizes=(64,),
learning_rate_init=0.0341261142176991, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0013704574812188038, hidden_layer_sizes=(64,
32), learning_rate_init=0.02575884550556351, max_iter=834, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0016102040635334325, hidden_layer_sizes=(64,),
learning_rate_init=0.029385016390999576, max_iter=1276, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0016102040635334325, hidden_layer_sizes=(64,),
learning_rate_init=0.029385016390999576, max_iter=1276, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0016102040635334325, hidden_layer_sizes=(64,),
learning_rate_init=0.029385016390999576, max_iter=1276, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007483140212909127, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.040258798069650686, max_iter=1362, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0016102040635334325, hidden_layer_sizes=(64,),
learning_rate_init=0.029385016390999576, max_iter=1276, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0016102040635334325, hidden_layer_sizes=(64,),
learning_rate_init=0.029385016390999576, max_iter=1276, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0013704574812188038, hidden_layer_sizes=(64,
32), learning_rate_init=0.02575884550556351, max_iter=834, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0012231818227197223, hidden_layer_sizes=(64,),
learning_rate_init=0.0341261142176991, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0025768543237849957, hidden_layer_sizes=(64,
32), learning_rate_init=0.04797494707820946, max_iter=1069, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0025768543237849957, hidden_layer_sizes=(64,
32), learning_rate_init=0.04797494707820946, max_iter=1069, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0025768543237849957, hidden_layer_sizes=(64,
32), learning_rate_init=0.04797494707820946, max_iter=1069, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001216031869068446, hidden_layer_sizes=(64,),
learning_rate_init=0.02798460661945399, max_iter=1016, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0025768543237849957, hidden_layer_sizes=(64,
32), learning_rate_init=0.04797494707820946, max_iter=1069, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0025768543237849957, hidden_layer_sizes=(64,
32), learning_rate_init=0.04797494707820946, max_iter=1069, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001216031869068446, hidden_layer_sizes=(64,),
learning_rate_init=0.02798460661945399, max_iter=1016, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00027365193103931246,
hidden_layer_sizes=(32,), learning_rate_init=0.039612238464832875,
max_iter=1447, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.001216031869068446, hidden_layer_sizes=(64,),
learning_rate_init=0.02798460661945399, max_iter=1016, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.001871102743060054, hidden_layer_sizes=(128,
64), learning_rate_init=0.008046211248738132, max_iter=814, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002815622705069351, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.010799143120957262, max_iter=1095, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.001216031869068446, hidden_layer_sizes=(64,),
learning_rate_init=0.02798460661945399, max_iter=1016, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=9.223946814525337e-05,
hidden_layer_sizes=(128,), learning_rate_init=0.036342867192380855,
max_iter=1474, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=9.223946814525337e-05,
hidden_layer_sizes=(128,), learning_rate_init=0.036342867192380855,
max_iter=1474, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00027365193103931246,
hidden_layer_sizes=(32,), learning_rate_init=0.039612238464832875,
max_iter=1447, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.001871102743060054, hidden_layer_sizes=(128,
64), learning_rate_init=0.008046211248738132, max_iter=814, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.001871102743060054, hidden_layer_sizes=(128,
64), learning_rate_init=0.008046211248738132, max_iter=814, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002815622705069351, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.010799143120957262, max_iter=1095, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.00027365193103931246,
hidden_layer_sizes=(32,), learning_rate_init=0.039612238464832875,
max_iter=1447, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.002815622705069351, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.010799143120957262, max_iter=1095, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.00027365193103931246,
hidden_layer_sizes=(32,), learning_rate_init=0.039612238464832875,
max_iter=1447, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=9.223946814525337e-05,
hidden_layer_sizes=(128,), learning_rate_init=0.036342867192380855,
max_iter=1474, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.001216031869068446, hidden_layer_sizes=(64,),
learning_rate_init=0.02798460661945399, max_iter=1016, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0026393102776267807, hidden_layer_sizes=(64,
32), learning_rate_init=0.01754490124263246, max_iter=847, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002363811040057837, hidden_layer_sizes=(128,
64), learning_rate_init=0.047315043925667453, max_iter=840, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0026393102776267807, hidden_layer_sizes=(64,
32), learning_rate_init=0.01754490124263246, max_iter=847, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=9.223946814525337e-05,
hidden_layer_sizes=(128,), learning_rate_init=0.036342867192380855,
max_iter=1474, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.002363811040057837, hidden_layer_sizes=(128,
64), learning_rate_init=0.047315043925667453, max_iter=840, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0026393102776267807, hidden_layer_sizes=(64,
32), learning_rate_init=0.01754490124263246, max_iter=847, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.001871102743060054, hidden_layer_sizes=(128,
64), learning_rate_init=0.008046211248738132, max_iter=814, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002363811040057837, hidden_layer_sizes=(128,
64), learning_rate_init=0.047315043925667453, max_iter=840, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=9.223946814525337e-05,
hidden_layer_sizes=(128,), learning_rate_init=0.036342867192380855,
max_iter=1474, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.001871102743060054, hidden_layer_sizes=(128,
64), learning_rate_init=0.008046211248738132, max_iter=814, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002815622705069351, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.010799143120957262, max_iter=1095, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.002639310277626780 7, hidden_layer_sizes=(64,
32), learning_rate_init=0.01754490124263246, max_iter=847, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00027365193103931246,
hidden_layer_sizes=(32,), learning_rate_init=0.039612238464832875,
max_iter=1447, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0026393102776267807, hidden_layer_sizes=(64,
32), learning_rate_init=0.01754490124263246, max_iter=847, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002815622705069351, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.010799143120957262, max_iter=1095, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0009829469651469866, hidden_layer_sizes=(128,
64), learning_rate_init=0.0374803089169032, max_iter=1346, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001332623055075649, hidden_layer_sizes=(64,),
learning_rate_init=0.02296682509328851, max_iter=1490, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0009829469651469866, hidden_layer_sizes=(128,
64), learning_rate_init=0.0374803089169032, max_iter=1346, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002363811040057837, hidden_layer_sizes=(128,
64), learning_rate_init=0.047315043925667453, max_iter=840, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.001332623055075649, hidden_layer_sizes=(64,),
learning_rate_init=0.02296682509328851, max_iter=1490, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0011987806242613694, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.043445691213304195, max_iter=804, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0027372907870205584, hidden_layer_sizes=(32,),
learning_rate_init=0.016717799053816335, max_iter=895, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0011987806242613694, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.043445691213304195, max_iter=804, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0027372907870205584, hidden_layer_sizes=(32,),
learning_rate_init=0.016717799053816335, max_iter=895, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0011987806242613694, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.043445691213304195, max_iter=804, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0009829469651469866, hidden_layer_sizes=(128,
64), learning_rate_init=0.0374803089169032, max_iter=1346, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002363811040057837, hidden_layer_sizes=(128,
64), learning_rate_init=0.047315043925667453, max_iter=840, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007579547592468672, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.02568977981821954, max_iter=1192, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0027372907870205584, hidden_layer_sizes=(32,),
learning_rate_init=0.016717799053816335, max_iter=895, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007579547592468672, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.02568977981821954, max_iter=1192, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.001332623055075649, hidden_layer_sizes=(64,),
learning_rate_init=0.02296682509328851, max_iter=1490, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.001332623055075649, hidden_layer_sizes=(64,),
learning_rate_init=0.02296682509328851, max_iter=1490, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0009829469651469866, hidden_layer_sizes=(128,
64), learning_rate_init=0.0374803089169032, max_iter=1346, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0027372907870205584, hidden_layer_sizes=(32,),
learning_rate_init=0.016717799053816335, max_iter=895, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0027726994217782794, hidden_layer_sizes=(128,
64), learning_rate_init=0.02799205456508366, max_iter=1340, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0028785607116739434,
hidden_layer_sizes=(128,), learning_rate_init=0.03573924665198523,
max_iter=1377, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.002474361138693251, hidden_layer_sizes=(32,),
learning_rate_init=0.006274712991513531, max_iter=827, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0009829469651469866, hidden_layer_sizes=(128,
64), learning_rate_init=0.0374803089169032, max_iter=1346, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0027372907870205584, hidden_layer_sizes=(32,),
learning_rate_init=0.016717799053816335, max_iter=895, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007579547592468672, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.02568977981821954, max_iter=1192, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.001332623055075649, hidden_layer_sizes=(64,),
learning_rate_init=0.02296682509328851, max_iter=1490, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002474361138693251, hidden_layer_sizes=(32,),
learning_rate_init=0.006274712991513531, max_iter=827, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0011987806242613694, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.043445691213304195, max_iter=804, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0004162638641020201, hidden_layer_sizes=(32,),
learning_rate_init=0.016900173748593195, max_iter=1024, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0004162638641020201, hidden_layer_sizes=(32,),
learning_rate_init=0.016900173748593195, max_iter=1024, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0028785607116739434,
hidden_layer_sizes=(128,), learning_rate_init=0.03573924665198523,
max_iter=1377, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0007579547592468672, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.02568977981821954, max_iter=1192, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0011987806242613694, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.043445691213304195, max_iter=804, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0027726994217782794, hidden_layer_sizes=(128,
64), learning_rate_init=0.02799205456508366, max_iter=1340, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0016063718652300982, hidden_layer_sizes=(32,),
learning_rate_init=0.021191808552902043, max_iter=1296, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005439675594288252,
hidden_layer_sizes=(128,), learning_rate_init=0.021870550157438953,
max_iter=1476, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.002474361138693251, hidden_layer_sizes=(32,),
learning_rate_init=0.006274712991513531, max_iter=827, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002474361138693251, hidden_layer_sizes=(32,),
learning_rate_init=0.006274712991513531, max_iter=827, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0016063718652300982, hidden_layer_sizes=(32,),
learning_rate_init=0.021191808552902043, max_iter=1296, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0027726994217782794, hidden_layer_sizes=(128,
64), learning_rate_init=0.02799205456508366, max_iter=1340, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0016063718652300982, hidden_layer_sizes=(32,),
learning_rate_init=0.021191808552902043, max_iter=1296, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0004162638641020201, hidden_layer_sizes=(32,),
learning_rate_init=0.016900173748593195, max_iter=1024, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002474361138693251, hidden_layer_sizes=(32,),
learning_rate_init=0.006274712991513531, max_iter=827, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005439675594288252,
hidden_layer_sizes=(128,), learning_rate_init=0.021870550157438953,
max_iter=1476, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0016063718652300982, hidden_layer_sizes=(32,),
learning_rate_init=0.021191808552902043, max_iter=1296, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.000733805487625825, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.001026018849765791, max_iter=1058, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0004162638641020201, hidden_layer_sizes=(32,),
learning_rate_init=0.016900173748593195, max_iter=1024, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0004942607898601997, hidden_layer_sizes=(32,),
learning_rate_init=0.014339050713764252, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0016063718652300982, hidden_layer_sizes=(32,),
learning_rate_init=0.021191808552902043, max_iter=1296, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007579547592468672, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.02568977981821954, max_iter=1192, solver=adam;
total time=   0.1s
[CV] END activation=relu, alpha=0.0004095961020010484, hidden_layer_sizes=(64,
32), learning_rate_init=0.013206276112388709, max_iter=1173, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006097101766581076, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.04516401294594342, max_iter=1439, solver=adam;
total time=   0.1s
[CV] END activation=relu, alpha=0.000733805487625825, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0010260188497657691, max_iter=1058, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0004162638641020201, hidden_layer_sizes=(32,),
learning_rate_init=0.016900173748593195, max_iter=1024, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005439675594288252,
hidden_layer_sizes=(128,), learning_rate_init=0.021870550157438953,
max_iter=1476, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0028785607116739434,
hidden_layer_sizes=(128,), learning_rate_init=0.03573924665198523,
max_iter=1377, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0004942607898601997, hidden_layer_sizes=(32,),
learning_rate_init=0.014339050713764252, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0029373418848263337, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.03581521364198942, max_iter=1186, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0027726994217782794, hidden_layer_sizes=(128,
64), learning_rate_init=0.02799205456508366, max_iter=1340, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004095961020010484, hidden_layer_sizes=(64,
32), learning_rate_init=0.013206276112388709, max_iter=1173, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004095961020010484, hidden_layer_sizes=(64,
32), learning_rate_init=0.013206276112388709, max_iter=1173, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004942607898601997, hidden_layer_sizes=(32,),
learning_rate_init=0.014339050713764252, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.000733805487625825, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.001026018849765791, max_iter=1058, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0006097101766581076, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.04516401294594342, max_iter=1439, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0029373418848263337, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.03581521364198942, max_iter=1186, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0028785607116739434,
hidden_layer_sizes=(128,), learning_rate_init=0.03573924665198523,
max_iter=1377, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0020921163546643007, hidden_layer_sizes=(128,
64), learning_rate_init=0.025472638013878154, max_iter=1471, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0020921163546643007, hidden_layer_sizes=(128,
64), learning_rate_init=0.025472638013878154, max_iter=1471, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0012831110399546938, hidden_layer_sizes=(128,
64), learning_rate_init=0.01825356240133415, max_iter=1184, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007761658145345013, hidden_layer_sizes=(64,
32), learning_rate_init=0.005042666316635762, max_iter=983, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004095961020010484, hidden_layer_sizes=(64,
32), learning_rate_init=0.013206276112388709, max_iter=1173, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005439675594288252,
hidden_layer_sizes=(128,), learning_rate_init=0.021870550157438953,
max_iter=1476, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.000733805487625825, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.001026018849765791, max_iter=1058, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0028785607116739434,
hidden_layer_sizes=(128,), learning_rate_init=0.03573924665198523,
max_iter=1377, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0007761658145345013, hidden_layer_sizes=(64,
32), learning_rate_init=0.005042666316635762, max_iter=983, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004095961020010484, hidden_layer_sizes=(64,
32), learning_rate_init=0.013206276112388709, max_iter=1173, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0007764798148322777, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.019389156635962662, max_iter=1409, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0005439675594288252,
hidden_layer_sizes=(128,), learning_rate_init=0.021870550157438953,
max_iter=1476, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0012831110399546938, hidden_layer_sizes=(128,
64), learning_rate_init=0.01825356240133415, max_iter=1184, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0029373418848263337, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.03581521364198942, max_iter=1186, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0027726994217782794, hidden_layer_sizes=(128,
64), learning_rate_init=0.02799205456508366, max_iter=1340, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0012464733273336592, hidden_layer_sizes=(32,),
learning_rate_init=0.008535877198271473, max_iter=986, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007761658145345013, hidden_layer_sizes=(64,
32), learning_rate_init=0.005042666316635762, max_iter=983, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0020921163546643007, hidden_layer_sizes=(128,
64), learning_rate_init=0.025472638013878154, max_iter=1471, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0029373418848263337, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.03581521364198942, max_iter=1186, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0012831110399546938, hidden_layer_sizes=(128,
64), learning_rate_init=0.01825356240133415, max_iter=1184, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0012427160632625669, hidden_layer_sizes=(64,),
learning_rate_init=0.0428855052953664, max_iter=1272, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0015862791748978429,
hidden_layer_sizes=(128,), learning_rate_init=0.01843329936458647,
max_iter=1288, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0012464733273336592, hidden_layer_sizes=(32,),
learning_rate_init=0.008535877198271473, max_iter=986, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0021374384203726457,
hidden_layer_sizes=(128,), learning_rate_init=0.030544647159412092,
max_iter=1202, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0004942607898601997, hidden_layer_sizes=(32,),
learning_rate_init=0.014339050713764252, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0021374384203726457,
hidden_layer_sizes=(128,), learning_rate_init=0.030544647159412092,
max_iter=1202, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0012427160632625669, hidden_layer_sizes=(64,),
learning_rate_init=0.0428855052953664, max_iter=1272, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0012464733273336592, hidden_layer_sizes=(32,),
learning_rate_init=0.008535877198271473, max_iter=986, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007761658145345013, hidden_layer_sizes=(64,
32), learning_rate_init=0.005042666316635762, max_iter=983, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0020921163546643007, hidden_layer_sizes=(128,
64), learning_rate_init=0.025472638013878154, max_iter=1471, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0007764798148322777, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.019389156635962662, max_iter=1409, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0007764798148322777, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.019389156635962662, max_iter=1409, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0015862791748978429,
hidden_layer_sizes=(128,), learning_rate_init=0.01843329936458647,
max_iter=1288, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0015862791748978429,
hidden_layer_sizes=(128,), learning_rate_init=0.01843329936458647,
max_iter=1288, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.000733805487625825, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.001026018849765791, max_iter=1058, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0012464733273336592, hidden_layer_sizes=(32,),
learning_rate_init=0.008535877198271473, max_iter=986, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0012427160632625669, hidden_layer_sizes=(64,),
learning_rate_init=0.0428855052953664, max_iter=1272, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0004942607898601997, hidden_layer_sizes=(32,),
learning_rate_init=0.014339050713764252, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0029373418848263337, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.03581521364198942, max_iter=1186, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0007761658145345013, hidden_layer_sizes=(64,
32), learning_rate_init=0.005042666316635762, max_iter=983, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0012831110399546938, hidden_layer_sizes=(128,
64), learning_rate_init=0.01825356240133415, max_iter=1184, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0021374384203726457,
hidden_layer_sizes=(128,), learning_rate_init=0.030544647159412092,
max_iter=1202, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0012427160632625669, hidden_layer_sizes=(64,),
learning_rate_init=0.0428855052953664, max_iter=1272, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005748647812878086, hidden_layer_sizes=(32,),
learning_rate_init=0.026831794563550716, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005748647812878086, hidden_layer_sizes=(32,),
learning_rate_init=0.026831794563550716, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0015862791748978429,
hidden_layer_sizes=(128,), learning_rate_init=0.01843329936458647,
max_iter=1288, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0012427160632625669, hidden_layer_sizes=(64,),
learning_rate_init=0.0428855052953664, max_iter=1272, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005748647812878086, hidden_layer_sizes=(32,),
learning_rate_init=0.026831794563550716, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0021374384203726457,
hidden_layer_sizes=(128,), learning_rate_init=0.030544647159412092,
max_iter=1202, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0020921163546643007, hidden_layer_sizes=(128,
64), learning_rate_init=0.025472638013878154, max_iter=1471, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00032930830341769765, hidden_layer_sizes=(64,
32), learning_rate_init=0.028914672680354882, max_iter=959, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0007764798148322777, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.019389156635962662, max_iter=1409, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.00032930830341769765, hidden_layer_sizes=(64,
32), learning_rate_init=0.028914672680354882, max_iter=959, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0015862791748978429,
hidden_layer_sizes=(128,), learning_rate_init=0.01843329936458647,
max_iter=1288, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00032930830341769765, hidden_layer_sizes=(64,
32), learning_rate_init=0.028914672680354882, max_iter=959, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005748647812878086, hidden_layer_sizes=(32,),
learning_rate_init=0.026831794563550716, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006161213250241382, hidden_layer_sizes=(32,),
learning_rate_init=0.018460478730633046, max_iter=1007, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0007764798148322777, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.019389156635962662, max_iter=1409, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0006161213250241382, hidden_layer_sizes=(32,),
learning_rate_init=0.018460478730633046, max_iter=1007, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006161213250241382, hidden_layer_sizes=(32,),
learning_rate_init=0.018460478730633046, max_iter=1007, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0021374384203726457,
hidden_layer_sizes=(128,), learning_rate_init=0.030544647159412092,
max_iter=1202, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0028240808548356884, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.035789219967254114, max_iter=1062, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0028240808548356884, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.035789219967254114, max_iter=1062, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0006161213250241382, hidden_layer_sizes=(32,),
learning_rate_init=0.018460478730633046, max_iter=1007, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0012464733273336592, hidden_layer_sizes=(32,),
learning_rate_init=0.008535877198271473, max_iter=986, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0012831110399546938, hidden_layer_sizes=(128,
64), learning_rate_init=0.01825356240133415, max_iter=1184, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005748647812878086, hidden_layer_sizes=(32,),
learning_rate_init=0.026831794563550716, max_iter=801, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00032930830341769765, hidden_layer_sizes=(64,
32), learning_rate_init=0.028914672680354882, max_iter=959, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006097101766581076, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.04516401294594342, max_iter=1439, solver=adam;
total time=   0.2s
[CV] END activation=relu, alpha=0.0006161213250241382, hidden_layer_sizes=(32,),
learning_rate_init=0.018460478730633046, max_iter=1007, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002856904982662843, hidden_layer_sizes=(64,
32), learning_rate_init=0.04592770942635396, max_iter=1053, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0011668482996852292, hidden_layer_sizes=(32,),
learning_rate_init=0.03417508845540279, max_iter=1137, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002856904982662843, hidden_layer_sizes=(64,
32), learning_rate_init=0.04592770942635396, max_iter=1053, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002856904982662843, hidden_layer_sizes=(64,
32), learning_rate_init=0.04592770942635396, max_iter=1053, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0011668482996852292, hidden_layer_sizes=(32,),
learning_rate_init=0.03417508845540279, max_iter=1137, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.002856904982662843, hidden_layer_sizes=(64,
32), learning_rate_init=0.04592770942635396, max_iter=1053, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00032930830341769765, hidden_layer_sizes=(64,
32), learning_rate_init=0.028914672680354882, max_iter=959, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.002856904982662843, hidden_layer_sizes=(64,
32), learning_rate_init=0.04592770942635396, max_iter=1053, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0026189728756342776, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0399937772928812, max_iter=1427, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0026189728756342776, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0399937772928812, max_iter=1427, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0011668482996852292, hidden_layer_sizes=(32,),
learning_rate_init=0.03417508845540279, max_iter=1137, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0011668482996852292, hidden_layer_sizes=(32,),
learning_rate_init=0.03417508845540279, max_iter=1137, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0028240808548356884, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.035789219967254114, max_iter=1062, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0011668482996852292, hidden_layer_sizes=(32,),
learning_rate_init=0.03417508845540279, max_iter=1137, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0026189728756342776, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0399937772928812, max_iter=1427, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0026189728756342776, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0399937772928812, max_iter=1427, solver=adam;
total time=   0.0s
[CV] END activation=relu, alpha=0.0026189728756342776, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.0399937772928812, max_iter=1427, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.00212568559307808, hidden_layer_sizes=(128,
64), learning_rate_init=0.012213465473027992, max_iter=960, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.002282511569169687, hidden_layer_sizes=(128,),
learning_rate_init=0.03388064461501717, max_iter=832, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00212568559307808, hidden_layer_sizes=(128,
64), learning_rate_init=0.012213465473027992, max_iter=960, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00212568559307808, hidden_layer_sizes=(128,
64), learning_rate_init=0.012213465473027992, max_iter=960, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007617472624904003, hidden_layer_sizes=(128,
64), learning_rate_init=0.009941135461066442, max_iter=1303, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.002282511569169687, hidden_layer_sizes=(128,),
learning_rate_init=0.03388064461501717, max_iter=832, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007617472624904003, hidden_layer_sizes=(128,
64), learning_rate_init=0.009941135461066442, max_iter=1303, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006097101766581076, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.04516401294594342, max_iter=1439, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.00212568559307808, hidden_layer_sizes=(128,
64), learning_rate_init=0.012213465473027992, max_iter=960, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0007617472624904003, hidden_layer_sizes=(128,
64), learning_rate_init=0.009941135461066442, max_iter=1303, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00212568559307808, hidden_layer_sizes=(128,
64), learning_rate_init=0.012213465473027992, max_iter=960, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0007617472624904003, hidden_layer_sizes=(128,
64), learning_rate_init=0.009941135461066442, max_iter=1303, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.002282511569169687, hidden_layer_sizes=(128,),
learning_rate_init=0.03388064461501717, max_iter=832, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0007617472624904003, hidden_layer_sizes=(128,
64), learning_rate_init=0.009941135461066442, max_iter=1303, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.002282511569169687, hidden_layer_sizes=(128,),
learning_rate_init=0.03388064461501717, max_iter=832, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00033102430348427744,
hidden_layer_sizes=(128,), learning_rate_init=0.0328166309092947 7, max_iter=821,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.002282511569169687, hidden_layer_sizes=(128,),
learning_rate_init=0.03388064461501717, max_iter=832, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00033102430348427744,
hidden_layer_sizes=(128,), learning_rate_init=0.03281663090929477, max_iter=821,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0028240808548356884, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.035789219967254114, max_iter=1062, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.00033102430348427744,
hidden_layer_sizes=(128,), learning_rate_init=0.03281663090929477, max_iter=821,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033102430348427744,
hidden_layer_sizes=(128,), learning_rate_init=0.03281663090929477, max_iter=821,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033102430348427744,
hidden_layer_sizes=(128,), learning_rate_init=0.03281663090929477, max_iter=821,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0019434158779917886, hidden_layer_sizes=(64,),
learning_rate_init=0.018503920384733787, max_iter=1408, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0019434158779917886, hidden_layer_sizes=(64,),
learning_rate_init=0.018503920384733787, max_iter=1408, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0019434158779917886, hidden_layer_sizes=(64,),
learning_rate_init=0.018503920384733787, max_iter=1408, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002056772178989299, hidden_layer_sizes=(32,),
learning_rate_init=0.010762149389902227, max_iter=1458, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002056772178989299, hidden_layer_sizes=(32,),
learning_rate_init=0.010762149389902227, max_iter=1458, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.002056772178989299, hidden_layer_sizes=(32,),
learning_rate_init=0.010762149389902227, max_iter=1458, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0018196125426816316, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.020654886233338023, max_iter=1237, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.002056772178989299, hidden_layer_sizes=(32,),
learning_rate_init=0.010762149389902227, max_iter=1458, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0019434158779917886, hidden_layer_sizes=(64,),
learning_rate_init=0.018503920384733787, max_iter=1408, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0019434158779917886, hidden_layer_sizes=(64,),
learning_rate_init=0.018503920384733787, max_iter=1408, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0018196125426816316, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.020654886233338023, max_iter=1237, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0018196125426816316, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.020654886233338023, max_iter=1237, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.002056772178989299, hidden_layer_sizes=(32,),
learning_rate_init=0.010762149389902227, max_iter=1458, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001229478992943615, hidden_layer_sizes=(32,),
learning_rate_init=0.0031801885877216877, max_iter=971, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001229478992943615, hidden_layer_sizes=(32,),
learning_rate_init=0.0031801885877216877, max_iter=971, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001229478992943615, hidden_layer_sizes=(32,),
learning_rate_init=0.0031801885877216877, max_iter=971, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001229478992943615, hidden_layer_sizes=(32,),
learning_rate_init=0.0031801885877216877, max_iter=971, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0018196125426816316, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.020654886233338023, max_iter=1237, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0001229478992943615, hidden_layer_sizes=(32,),
learning_rate_init=0.0031801885877216877, max_iter=971, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0028240808548356884, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.035789219967254114, max_iter=1062, solver=adam;
total time=   0.0s
[CV] END activation=tanh, alpha=0.0018196125426816316, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.020654886233338023, max_iter=1237, solver=adam;
total time=   0.1s
[CV] END activation=tanh, alpha=0.0006097101766581076, hidden_layer_sizes=(128,
64, 32), learning_rate_init=0.04516401294594342, max_iter=1439, solver=adam;
total time=   0.1s
 Best MLP params: {'activation': 'tanh', 'alpha':
np.float64(0.0007761658145345013), 'hidden_layer_sizes': (64, 32),
'learning_rate_init': np.float64(0.005042666316635762), 'max_iter': 983,
'solver': 'adam'}
```

[90]:

```python
from sklearn.neural_network import MLPRegressor
from sklearn.model_selection import RandomizedSearchCV
from sklearn.preprocessing import StandardScaler
from scipy.stats import uniform, randint
import numpy as np

# === 1. (Optional but recommended) Standardize meta features ===
scaler_meta = StandardScaler()
X_meta_train_scaled = scaler_meta.fit_transform(X_meta_train)
X_meta_test_scaled = scaler_meta.transform(X_meta_test)

# === 2. Define parameter search space (narrowed around best region) ===
param_dist = {
    'hidden_layer_sizes': [(32,), (64,), (128,), (64, 32), (128, 64)],
    'activation': ['relu', 'tanh'],
    'alpha': uniform(0.0001, 0.0006),             # Regularization ~ [0.0003,␣
 ↪0.0006]
    'learning_rate_init': uniform(0.005, 0.04),       # Learning rate ~ [0.02,␣
 ↪0.04]
    'max_iter': randint(1000, 1300),
    'solver': ['adam']
}

# === 3. Setup randomized search ===
search = RandomizedSearchCV(
    estimator=MLPRegressor(random_state=42),
    param_distributions=param_dist,
    n_iter=100,
    scoring='neg_mean_squared_error',
    cv=5,
    n_jobs=-1,
    verbose=2,
    random_state=42
)

# === 4. Run search ===
search.fit(X_meta_train_scaled, y_meta_train)
print("  Best MLP params:", search.best_params_)
```

```
Fitting 5 folds for each of 100 candidates, totalling 500 fits
[CV] END activation=relu, alpha=0.0005779257921161397,
hidden_layer_sizes=(128,), learning_rate_init=0.03618764001091077,
max_iter=1020, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005779257921161397,
hidden_layer_sizes=(128,), learning_rate_init=0.03618764001091077,
max_iter=1020, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005779257921161397,
hidden_layer_sizes=(128,), learning_rate_init=0.03618764001091077,
max_iter=1020, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005779257921161397,
hidden_layer_sizes=(128,), learning_rate_init=0.03618764001091077,
max_iter=1020, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036749965171215465,
hidden_layer_sizes=(128,), learning_rate_init=0.023369955678634688,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005779257921161397,
hidden_layer_sizes=(128,), learning_rate_init=0.03618764001091077,
max_iter=1020, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036749965171215465,
hidden_layer_sizes=(128,), learning_rate_init=0.023369955678634688,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036749965171215465,
hidden_layer_sizes=(128,), learning_rate_init=0.023369955678634688,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036749965171215465,
hidden_layer_sizes=(128,), learning_rate_init=0.023369955678634688,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036749965171215465,
hidden_layer_sizes=(128,), learning_rate_init=0.023369955678634688,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0003591670111852694, hidden_layer_sizes=(32,),
learning_rate_init=0.025990986410335568, max_iter=1169, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003591670111852694, hidden_layer_sizes=(32,),
learning_rate_init=0.025990986410335568, max_iter=1169, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003591670111852694, hidden_layer_sizes=(32,),
learning_rate_init=0.025990986410335568, max_iter=1169, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003591670111852694, hidden_layer_sizes=(32,),
learning_rate_init=0.025990986410335568, max_iter=1169, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005248435466776273, hidden_layer_sizes=(128,
64), learning_rate_init=0.04379639408647977, max_iter=1293, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003591670111852694, hidden_layer_sizes=(32,),
learning_rate_init=0.025990986410335568, max_iter=1169, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002752867891211309,
hidden_layer_sizes=(128,), learning_rate_init=0.02029847965068651,
max_iter=1243, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00020909498032426035, hidden_layer_sizes=(128,
64), learning_rate_init=0.029699260385108662, max_iter=1021, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020909498032426035, hidden_layer_sizes=(128,
64), learning_rate_init=0.029699260385108662, max_iter=1021, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020909498032426035, hidden_layer_sizes=(128,
64), learning_rate_init=0.029699260385108662, max_iter=1021, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005248435466776273, hidden_layer_sizes=(128,
64), learning_rate_init=0.04379639408647977, max_iter=1293, solver=adam; total
```

```
time=   0.0s
[CV] END activation=tanh, alpha=0.0005248435466776273, hidden_layer_sizes=(128,
64), learning_rate_init=0.04379639408647977, max_iter=1293, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005248435466776273, hidden_layer_sizes=(128,
64), learning_rate_init=0.04379639408647977, max_iter=1293, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0005248435466776273, hidden_layer_sizes=(128,
64), learning_rate_init=0.04379639408647977, max_iter=1293, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0002752867891211309,
hidden_layer_sizes=(128,), learning_rate_init=0.02029847965068651,
max_iter=1243, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0002752867891211309,
hidden_layer_sizes=(128,), learning_rate_init=0.02029847965068651,
max_iter=1243, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00020909498032426035, hidden_layer_sizes=(128,
64), learning_rate_init=0.029699260385108662, max_iter=1021, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.00033124990152394966,
hidden_layer_sizes=(64,), learning_rate_init=0.008906884560255355,
max_iter=1091, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0001390309557911677, hidden_layer_sizes=(64,
32), learning_rate_init=0.04268807022739411, max_iter=1013, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020909498032426035, hidden_layer_sizes=(128,
64), learning_rate_init=0.029699260385108662, max_iter=1021, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.00033124990152394966,
hidden_layer_sizes=(64,), learning_rate_init=0.008906884560255355,
max_iter=1091, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033124990152394966,
hidden_layer_sizes=(64,), learning_rate_init=0.008906884560255355,
max_iter=1091, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005099581112952749, hidden_layer_sizes=(64,
32), learning_rate_init=0.02480707640445081, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002752867891211309,
hidden_layer_sizes=(128,), learning_rate_init=0.02029847965068651,
max_iter=1243, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005099581112952749, hidden_layer_sizes=(64,
32), learning_rate_init=0.02480707640445081, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00033124990152394966,
hidden_layer_sizes=(64,), learning_rate_init=0.008906884560255355,
max_iter=1091, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003800577359487879, hidden_layer_sizes=(128,
64), learning_rate_init=0.029301794076057538, max_iter=1020, solver=adam; total
```

```
time=   0.0s
[CV] END activation=relu, alpha=0.0001390309557911677, hidden_layer_sizes=(64,
32), learning_rate_init=0.04268807022739411, max_iter=1013, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0003346363645439445, hidden_layer_sizes=(64,),
learning_rate_init=0.031500891374159276, max_iter=1001, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002752867891211309,
hidden_layer_sizes=(128,), learning_rate_init=0.02029847965068651,
max_iter=1243, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033124990152394966,
hidden_layer_sizes=(64,), learning_rate_init=0.008906884560255355,
max_iter=1091, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003346363645439445, hidden_layer_sizes=(64,),
learning_rate_init=0.031500891374159276, max_iter=1001, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0003800577359487879, hidden_layer_sizes=(128,
64), learning_rate_init=0.029301794076057538, max_iter=1020, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00041204081270668647,
hidden_layer_sizes=(64,), learning_rate_init=0.02770801311279966, max_iter=1190,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003800577359487879, hidden_layer_sizes=(128,
64), learning_rate_init=0.029301794076057538, max_iter=1020, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005099581112952749, hidden_layer_sizes=(64,
32), learning_rate_init=0.02480707640445081, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0003346363645439445, hidden_layer_sizes=(64,),
learning_rate_init=0.031500891374159276, max_iter=1001, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0003346363645439445, hidden_layer_sizes=(64,),
learning_rate_init=0.031500891374159276, max_iter=1001, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0001390309557911677, hidden_layer_sizes=(64,
32), learning_rate_init=0.04268807022739411, max_iter=1013, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00041204081270668647,
hidden_layer_sizes=(64,), learning_rate_init=0.02770801311279966, max_iter=1190,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005099581112952749, hidden_layer_sizes=(64,
32), learning_rate_init=0.02480707640445081, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0001390309557911677, hidden_layer_sizes=(64,
32), learning_rate_init=0.04268807022739411, max_iter=1013, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005650796940166687, hidden_layer_sizes=(64,),
learning_rate_init=0.02080600944007258, max_iter=1269, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00041204081270668647,
hidden_layer_sizes=(64,), learning_rate_init=0.02770801311279966, max_iter=1190,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003346363645439445, hidden_layer_sizes=(64,),
learning_rate_init=0.031500891374159276, max_iter=1001, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005650796940166687, hidden_layer_sizes=(64,),
learning_rate_init=0.02080600944007258, max_iter=1269, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00041204081270668647,
hidden_layer_sizes=(64,), learning_rate_init=0.02770801311279966, max_iter=1190,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00041204081270668647,
hidden_layer_sizes=(64,), learning_rate_init=0.02770801311279966, max_iter=1190,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005099581112952749, hidden_layer_sizes=(64,
32), learning_rate_init=0.02480707640445081, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0001390309557911677, hidden_layer_sizes=(64,
32), learning_rate_init=0.04268807022739411, max_iter=1013, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003332063738136892, hidden_layer_sizes=(64,),
learning_rate_init=0.026587685295563192, max_iter=1279, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006531245410138701, hidden_layer_sizes=(64,
32), learning_rate_init=0.006809091556421523, max_iter=1295, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005650796940166687, hidden_layer_sizes=(64,),
learning_rate_init=0.02080600944007258, max_iter=1269, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005650796940166687, hidden_layer_sizes=(64,),
learning_rate_init=0.02080600944007258, max_iter=1269, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003332063738136892, hidden_layer_sizes=(64,),
learning_rate_init=0.026587685295563192, max_iter=1279, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003332063738136892, hidden_layer_sizes=(64,),
learning_rate_init=0.026587685295563192, max_iter=1279, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006531245410138701, hidden_layer_sizes=(64,
32), learning_rate_init=0.006809091556421523, max_iter=1295, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006531245410138701, hidden_layer_sizes=(64,
32), learning_rate_init=0.006809091556421523, max_iter=1295, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0003800577359487879, hidden_layer_sizes=(128,
64), learning_rate_init=0.029301794076057538, max_iter=1020, solver=adam; total
```

```
time=   0.0s
[CV] END activation=relu, alpha=0.0005813181884524237, hidden_layer_sizes=(32,),
learning_rate_init=0.005625456269647757, max_iter=1008, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005650796940166687, hidden_layer_sizes=(64,),
learning_rate_init=0.02080600944007258, max_iter=1269, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005813181884524237, hidden_layer_sizes=(32,),
learning_rate_init=0.005625456269647757, max_iter=1008, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003332063738136892, hidden_layer_sizes=(64,),
learning_rate_init=0.026587685295563192, max_iter=1279, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003332063738136892, hidden_layer_sizes=(64,),
learning_rate_init=0.026587685295563192, max_iter=1279, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006791531843584827, hidden_layer_sizes=(64,
32), learning_rate_init=0.02670784332632994, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005813181884524237, hidden_layer_sizes=(32,),
learning_rate_init=0.005625456269647757, max_iter=1008, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006791531843584827, hidden_layer_sizes=(64,
32), learning_rate_init=0.02670784332632994, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006531245410138701, hidden_layer_sizes=(64,
32), learning_rate_init=0.006809091556421523, max_iter=1295, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006791531843584827, hidden_layer_sizes=(64,
32), learning_rate_init=0.02670784332632994, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005813181884524237, hidden_layer_sizes=(32,),
learning_rate_init=0.005625456269647757, max_iter=1008, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005813181884524237, hidden_layer_sizes=(32,),
learning_rate_init=0.005625456269647757, max_iter=1008, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0003800577359487879, hidden_layer_sizes=(128,
64), learning_rate_init=0.029301794076057538, max_iter=1020, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.0006531245410138701, hidden_layer_sizes=(64,
32), learning_rate_init=0.006809091556421523, max_iter=1295, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00033692891090534183,
hidden_layer_sizes=(128,), learning_rate_init=0.012953696163552207,
max_iter=1162, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033692891090534183,
hidden_layer_sizes=(128,), learning_rate_init=0.012953696163552207,
max_iter=1162, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006178620555253561, hidden_layer_sizes=(64,
32), learning_rate_init=0.018235920994105967, max_iter=1047, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006791531843584827, hidden_layer_sizes=(64,
32), learning_rate_init=0.02670784332632994, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00043674605550862066,
hidden_layer_sizes=(128,), learning_rate_init=0.00978376983753207,
max_iter=1130, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005627622080115675, hidden_layer_sizes=(128,
64), learning_rate_init=0.042052035140533954, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000533037712698924, hidden_layer_sizes=(128,),
learning_rate_init=0.025909313175279763, max_iter=1014, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00033692891090534183,
hidden_layer_sizes=(128,), learning_rate_init=0.012953696163552207,
max_iter=1162, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00033692891090534183,
hidden_layer_sizes=(128,), learning_rate_init=0.012953696163552207,
max_iter=1162, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.000288613588645796, hidden_layer_sizes=(64,
32), learning_rate_init=0.04130265895704372, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00043674605550862066,
hidden_layer_sizes=(128,), learning_rate_init=0.00978376983753207,
max_iter=1130, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002865893930293973, hidden_layer_sizes=(128,
64), learning_rate_init=0.03418424713352256, max_iter=1213, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000288613588645796, hidden_layer_sizes=(64,
32), learning_rate_init=0.04130265895704372, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002865893930293973, hidden_layer_sizes=(128,
64), learning_rate_init=0.03418424713352256, max_iter=1213, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000533037712698924, hidden_layer_sizes=(128,),
learning_rate_init=0.025909313175279763, max_iter=1014, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00033692891090534183,
hidden_layer_sizes=(128,), learning_rate_init=0.012953696163552207,
max_iter=1162, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006791531843584827, hidden_layer_sizes=(64,
32), learning_rate_init=0.02670784332632994, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042390465478100384, hidden_layer_sizes=(128,
64), learning_rate_init=0.014151926619664898, max_iter=1159, solver=adam; total
```

```
time=   0.0s
[CV] END activation=tanh, alpha=0.0006178620555253561, hidden_layer_sizes=(64,
32), learning_rate_init=0.018235920994105967, max_iter=1047, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002738508717482608, hidden_layer_sizes=(64,
32), learning_rate_init=0.04218790609370292, max_iter=1044, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005627622080115675, hidden_layer_sizes=(128,
64), learning_rate_init=0.042052035140533954, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000288613588645796, hidden_layer_sizes=(64,
32), learning_rate_init=0.04130265895704372, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000533037712698924, hidden_layer_sizes=(128,),
learning_rate_init=0.025909313175279763, max_iter=1014, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00043674605550862066,
hidden_layer_sizes=(128,), learning_rate_init=0.00978376983753207,
max_iter=1130, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00011525147604645712, hidden_layer_sizes=(128,
64), learning_rate_init=0.01306876809341585, max_iter=1230, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.0005627622080115675, hidden_layer_sizes=(128,
64), learning_rate_init=0.042052035140533954, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00048004225390625406, hidden_layer_sizes=(64,
32), learning_rate_init=0.037146883075964574, max_iter=1043, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002865893930293973, hidden_layer_sizes=(128,
64), learning_rate_init=0.03418424713352256, max_iter=1213, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006355353990939867,
hidden_layer_sizes=(128,), learning_rate_init=0.017973800840210958,
max_iter=1230, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.000533037712698924, hidden_layer_sizes=(128,),
learning_rate_init=0.025909313175279763, max_iter=1014, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00043674605550862066,
hidden_layer_sizes=(128,), learning_rate_init=0.00978376983753207,
max_iter=1130, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002738508717482608, hidden_layer_sizes=(64,
32), learning_rate_init=0.04218790609370292, max_iter=1044, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006178620555253561, hidden_layer_sizes=(64,
32), learning_rate_init=0.018235920994105967, max_iter=1047, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000288613588645796, hidden_layer_sizes=(64,
32), learning_rate_init=0.04130265895704372, max_iter=1230, solver=adam; total
```

```
time=   0.0s
[CV] END activation=tanh, alpha=0.0006355353990939867,
hidden_layer_sizes=(128,), learning_rate_init=0.017973800840210958,
max_iter=1230, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.000533037712698924, hidden_layer_sizes=(128,),
learning_rate_init=0.025909313175279763, max_iter=1014, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00043674605550862066,
hidden_layer_sizes=(128,), learning_rate_init=0.00978376983753207,
max_iter=1130, solver=adam; total time=   0.0s[CV] END activation=relu,
alpha=0.0005627622080115675, hidden_layer_sizes=(128, 64),
learning_rate_init=0.042052035140533954, max_iter=1040, solver=adam; total time=
0.0s

[CV] END activation=tanh, alpha=0.00048004225390625406, hidden_layer_sizes=(64,
32), learning_rate_init=0.037146883075964574, max_iter=1043, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042390465478100384, hidden_layer_sizes=(128,
64), learning_rate_init=0.014151926619664898, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042390465478100384, hidden_layer_sizes=(128,
64), learning_rate_init=0.014151926619664898, max_iter=1159, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.0005627622080115675, hidden_layer_sizes=(128,
64), learning_rate_init=0.042052035140533954, max_iter=1040, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006178620555253561, hidden_layer_sizes=(64,
32), learning_rate_init=0.018235920994105967, max_iter=1047, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000288613588645796, hidden_layer_sizes=(64,
32), learning_rate_init=0.04130265895704372, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002738508717482608, hidden_layer_sizes=(64,
32), learning_rate_init=0.04218790609370292, max_iter=1044, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002865893930293973, hidden_layer_sizes=(128,
64), learning_rate_init=0.03418424713352256, max_iter=1213, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00016603115471660606,
hidden_layer_sizes=(32,), learning_rate_init=0.030907604821654496,
max_iter=1120, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00020097462530375835,
hidden_layer_sizes=(32,), learning_rate_init=0.021153446842321635,
max_iter=1051, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002738508717482608, hidden_layer_sizes=(64,
32), learning_rate_init=0.04218790609370292, max_iter=1044, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00011525147604645712, hidden_layer_sizes=(128,
64), learning_rate_init=0.01306876809341585, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006355353990939867,
hidden_layer_sizes=(128,), learning_rate_init=0.017973800840210958,
max_iter=1230, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003115413138005014,
hidden_layer_sizes=(128,), learning_rate_init=0.005278085221247628,
max_iter=1136, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00048004225390625406, hidden_layer_sizes=(64,
32), learning_rate_init=0.037146883075964574, max_iter=1043, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00011525147604645712, hidden_layer_sizes=(128,
64), learning_rate_init=0.01306876809341585, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00023326468628243817,
hidden_layer_sizes=(128,), learning_rate_init=0.01577649335194086,
max_iter=1151, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006355353990939867,
hidden_layer_sizes=(128,), learning_rate_init=0.017973800840210958,
max_iter=1230, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00020097462530375835,
hidden_layer_sizes=(32,), learning_rate_init=0.021153446842321635,
max_iter=1051, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00016603115471660606,
hidden_layer_sizes=(32,), learning_rate_init=0.030907604821654496,
max_iter=1120, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00042390465478100384, hidden_layer_sizes=(128,
64), learning_rate_init=0.014151926619664898, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006178620555253561, hidden_layer_sizes=(64,
32), learning_rate_init=0.018235920994105967, max_iter=1047, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002738508717482608, hidden_layer_sizes=(64,
32), learning_rate_init=0.04218790609370292, max_iter=1044, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006355353990939867,
hidden_layer_sizes=(128,), learning_rate_init=0.017973800840210958,
max_iter=1230, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00023326468628243817,
hidden_layer_sizes=(128,), learning_rate_init=0.01577649335194086,
max_iter=1151, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00020097462530375835,
hidden_layer_sizes=(32,), learning_rate_init=0.021153446842321635,
max_iter=1051, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00048004225390625406, hidden_layer_sizes=(64,
32), learning_rate_init=0.037146883075964574, max_iter=1043, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0003115413138005014,
hidden_layer_sizes=(128,), learning_rate_init=0.005278085221247628,
max_iter=1136, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00048004225390625406, hidden_layer_sizes=(64,
32), learning_rate_init=0.037146883075964574, max_iter=1043, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00016603115471660606,
hidden_layer_sizes=(32,), learning_rate_init=0.030907604821654496,
max_iter=1120, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00020097462530375835,
hidden_layer_sizes=(32,), learning_rate_init=0.021153446842321635,
max_iter=1051, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00020097462530375835,
hidden_layer_sizes=(32,), learning_rate_init=0.021153446842321635,
max_iter=1051, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003115413138005014,
hidden_layer_sizes=(128,), learning_rate_init=0.005278085221247628,
max_iter=1136, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00023326468628243817,
hidden_layer_sizes=(128,), learning_rate_init=0.01577649335194086,
max_iter=1151, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004185607499902818, hidden_layer_sizes=(64,),
learning_rate_init=0.04442601816442403, max_iter=1246, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0002865893930293973, hidden_layer_sizes=(128,
64), learning_rate_init=0.03418424713352256, max_iter=1213, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002510693774952185, hidden_layer_sizes=(64,),
learning_rate_init=0.03349082359697768, max_iter=1100, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00016603115471660606,
hidden_layer_sizes=(32,), learning_rate_init=0.030907604821654496,
max_iter=1120, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003115413138005014,
hidden_layer_sizes=(128,), learning_rate_init=0.005278085221247628,
max_iter=1136, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004185607499902818, hidden_layer_sizes=(64,),
learning_rate_init=0.04442601816442403, max_iter=1246, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006986442910293651,
hidden_layer_sizes=(128,), learning_rate_init=0.029382573359195877,
max_iter=1129, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0002510693774952185, hidden_layer_sizes=(64,),
learning_rate_init=0.03349082359697768, max_iter=1100, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00014851199579962916,
hidden_layer_sizes=(128,), learning_rate_init=0.014686397531097035,
max_iter=1293, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00011525147604645712, hidden_layer_sizes=(128,
64), learning_rate_init=0.01306876809341585, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00013088725074999363, hidden_layer_sizes=(128,
64), learning_rate_init=0.01880284992106732, max_iter=1125, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00016603115471660606,
hidden_layer_sizes=(32,), learning_rate_init=0.030907604821654496,
max_iter=1120, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00042390465478100384, hidden_layer_sizes=(128,
64), learning_rate_init=0.014151926619664898, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.000382180380667623, hidden_layer_sizes=(64,),
learning_rate_init=0.03034118843043579, max_iter=1239, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00023326468628243817,
hidden_layer_sizes=(128,), learning_rate_init=0.01577649335194086,
max_iter=1151, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003115413138005014,
hidden_layer_sizes=(128,), learning_rate_init=0.005278085221247628,
max_iter=1136, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004185607499902818, hidden_layer_sizes=(64,),
learning_rate_init=0.04442601816442403, max_iter=1246, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00014851199579962916,
hidden_layer_sizes=(128,), learning_rate_init=0.014686397531097035,
max_iter=1293, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0002510693774952185, hidden_layer_sizes=(64,),
learning_rate_init=0.03349082359697768, max_iter=1100, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00029557534312113086, hidden_layer_sizes=(64,
32), learning_rate_init=0.005663513157114246, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006986442910293651,
hidden_layer_sizes=(128,), learning_rate_init=0.029382573359195877,
max_iter=1129, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.000382180380667623, hidden_layer_sizes=(64,),
learning_rate_init=0.03034118843043579, max_iter=1239, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00014851199579962916,
hidden_layer_sizes=(128,), learning_rate_init=0.014686397531097035,
max_iter=1293, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004185607499902818, hidden_layer_sizes=(64,),
learning_rate_init=0.04442601816442403, max_iter=1246, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006620379932420407, hidden_layer_sizes=(32,),
learning_rate_init=0.03202760468157123, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002510693774952185, hidden_layer_sizes=(64,),
learning_rate_init=0.03349082359697768, max_iter=1100, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006011814973535428, hidden_layer_sizes=(64,
32), learning_rate_init=0.02532795106962875, max_iter=1258, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00014851199579962916,
hidden_layer_sizes=(128,), learning_rate_init=0.014686397531097035,
max_iter=1293, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.000382180380667623, hidden_layer_sizes=(64,),
learning_rate_init=0.03034118843043579, max_iter=1239, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00023326468628243817,
hidden_layer_sizes=(128,), learning_rate_init=0.01577649335194086,
max_iter=1151, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004185607499902818, hidden_layer_sizes=(64,),
learning_rate_init=0.04442601816442403, max_iter=1246, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.000382180380667623, hidden_layer_sizes=(64,),
learning_rate_init=0.03034118843043579, max_iter=1239, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002510693774952185, hidden_layer_sizes=(64,),
learning_rate_init=0.03349082359697768, max_iter=1100, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006620379932420407, hidden_layer_sizes=(32,),
learning_rate_init=0.03202760468157123, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006986442910293651,
hidden_layer_sizes=(128,), learning_rate_init=0.029382573359195877,
max_iter=1129, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006986442910293651,
hidden_layer_sizes=(128,), learning_rate_init=0.029382573359195877,
max_iter=1129, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00011525147604645712, hidden_layer_sizes=(128,
64), learning_rate_init=0.01306876809341585, max_iter=1230, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00013088725074999363, hidden_layer_sizes=(128,
64), learning_rate_init=0.01880284992106732, max_iter=1125, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006011814973535428, hidden_layer_sizes=(64,
32), learning_rate_init=0.02532795106962875, max_iter=1258, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006137945751268555, hidden_layer_sizes=(32,),
learning_rate_init=0.03068126584617151, max_iter=1074, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.000382180380667623, hidden_layer_sizes=(64,),
learning_rate_init=0.03034118843043579, max_iter=1239, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006893010059976614, hidden_layer_sizes=(64,
32), learning_rate_init=0.02618602313424026, max_iter=1001, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00019697722845676824, hidden_layer_sizes=(64,
32), learning_rate_init=0.02822746485745819, max_iter=1111, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004871036742456699, hidden_layer_sizes=(64,
32), learning_rate_init=0.04262093057958416, max_iter=1127, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006620379932420407, hidden_layer_sizes=(32,),
learning_rate_init=0.03202760468157123, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006986442910293651,
hidden_layer_sizes=(128,), learning_rate_init=0.029382573359195877,
max_iter=1129, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006137945751268555, hidden_layer_sizes=(32,),
learning_rate_init=0.03068126584617151, max_iter=1074, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00014851199579962916,
hidden_layer_sizes=(128,), learning_rate_init=0.014686397531097035,
max_iter=1293, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00015586166068353954, hidden_layer_sizes=(64,
32), learning_rate_init=0.027331738144283906, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006640800654746671, hidden_layer_sizes=(32,),
learning_rate_init=0.005202463353848747, max_iter=1193, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006011814973535428, hidden_layer_sizes=(64,
32), learning_rate_init=0.02532795106962875, max_iter=1258, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000494567735380206, hidden_layer_sizes=(32,),
learning_rate_init=0.026701609222195975, max_iter=1021, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006011814973535428, hidden_layer_sizes=(64,
32), learning_rate_init=0.02532795106962875, max_iter=1258, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006137945751268555, hidden_layer_sizes=(32,),
learning_rate_init=0.03068126584617151, max_iter=1074, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.000494567735380206, hidden_layer_sizes=(32,),
learning_rate_init=0.026701609222195975, max_iter=1021, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006893010059976614, hidden_layer_sizes=(64,
32), learning_rate_init=0.02618602313424026, max_iter=1001, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006640800654746671, hidden_layer_sizes=(32,),
learning_rate_init=0.005202463353848747, max_iter=1193, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006264036120285886,
hidden_layer_sizes=(128,), learning_rate_init=0.014142000871891987,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004871036742456699, hidden_layer_sizes=(64,
32), learning_rate_init=0.04262093057958416, max_iter=1127, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0001145895798588723, hidden_layer_sizes=(32,),
learning_rate_init=0.00674415087017735, max_iter=1171, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00015586166068353954, hidden_layer_sizes=(64,
32), learning_rate_init=0.027331738144283906, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00036905448579174836, hidden_layer_sizes=(128,
64), learning_rate_init=0.01397077237842239, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006137945751268555, hidden_layer_sizes=(32,),
learning_rate_init=0.03068126584617151, max_iter=1074, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00013088725074999363, hidden_layer_sizes=(128,
64), learning_rate_init=0.01880284992106732, max_iter=1125, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.000494567735380206, hidden_layer_sizes=(32,),
learning_rate_init=0.026701609222195975, max_iter=1021, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006640800654746671, hidden_layer_sizes=(32,),
learning_rate_init=0.005202463353848747, max_iter=1193, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00021322426500482763,
hidden_layer_sizes=(32,), learning_rate_init=0.018968382984506437,
max_iter=1207, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006264036120285886,
hidden_layer_sizes=(128,), learning_rate_init=0.014142000871891987,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006011814973535428, hidden_layer_sizes=(64,
32), learning_rate_init=0.02532795106962875, max_iter=1258, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00045392250853632627, hidden_layer_sizes=(128,
64), learning_rate_init=0.024469686118378207, max_iter=1263, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00013088725074999363, hidden_layer_sizes=(128,
64), learning_rate_init=0.01880284992106732, max_iter=1125, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0006893010059976614, hidden_layer_sizes=(64,
32), learning_rate_init=0.02618602313424026, max_iter=1001, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003955106162913183, hidden_layer_sizes=(128,
64), learning_rate_init=0.014207410729662213, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00019697722845676824, hidden_layer_sizes=(64,
32), learning_rate_init=0.02822746485745819, max_iter=1111, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004871036742456699, hidden_layer_sizes=(64,
32), learning_rate_init=0.04262093057958416, max_iter=1127, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006640800654746671, hidden_layer_sizes=(32,),
learning_rate_init=0.005202463353848747, max_iter=1193, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00021322426500482763,
hidden_layer_sizes=(32,), learning_rate_init=0.018968382984506437,
max_iter=1207, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006569911375526352, hidden_layer_sizes=(32,),
learning_rate_init=0.027110598673419598, max_iter=1094, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.000494567735380206, hidden_layer_sizes=(32,),
learning_rate_init=0.026701609222195975, max_iter=1021, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006264036120285886,
hidden_layer_sizes=(128,), learning_rate_init=0.014142000871891987,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00029557534312113086, hidden_layer_sizes=(64,
32), learning_rate_init=0.005663513157114246, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00036905448579174836, hidden_layer_sizes=(128,
64), learning_rate_init=0.01397077237842239, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006569911375526352, hidden_layer_sizes=(32,),
learning_rate_init=0.027110598673419598, max_iter=1094, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.000494567735380206, hidden_layer_sizes=(32,),
learning_rate_init=0.026701609222195975, max_iter=1021, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00021322426500482763,
hidden_layer_sizes=(32,), learning_rate_init=0.018968382984506437,
max_iter=1207, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006893010059976614, hidden_layer_sizes=(64,
32), learning_rate_init=0.02618602313424026, max_iter=1001, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00019697722845676824, hidden_layer_sizes=(64,
32), learning_rate_init=0.02822746485745819, max_iter=1111, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006264036120285886,
hidden_layer_sizes=(128,), learning_rate_init=0.014142000871891987,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006640800654746671, hidden_layer_sizes=(32,),
learning_rate_init=0.005202463353848747, max_iter=1193, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00029557534312113086, hidden_layer_sizes=(64,
32), learning_rate_init=0.005663513157114246, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004871036742456699, hidden_layer_sizes=(64,
32), learning_rate_init=0.04262093057958416, max_iter=1127, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00038196670839456574, hidden_layer_sizes=(128,
64), learning_rate_init=0.040339760890650354, max_iter=1232, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.0006569911375526352, hidden_layer_sizes=(32,),
learning_rate_init=0.027110598673419598, max_iter=1094, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00045392250853632627, hidden_layer_sizes=(128,
64), learning_rate_init=0.024469686118378207, max_iter=1263, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00019697722845676824, hidden_layer_sizes=(64,
32), learning_rate_init=0.02822746485745819, max_iter=1111, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00021322426500482763,
hidden_layer_sizes=(32,), learning_rate_init=0.018968382984506437,
max_iter=1207, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00029557534312113086, hidden_layer_sizes=(64,
32), learning_rate_init=0.005663513157114246, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00013088725074999363, hidden_layer_sizes=(128,
64), learning_rate_init=0.01880284992106732, max_iter=1125, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003955106162913183, hidden_layer_sizes=(128,
64), learning_rate_init=0.014207410729662213, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00036905448579174836, hidden_layer_sizes=(128,
64), learning_rate_init=0.01397077237842239, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006264036120285886,
hidden_layer_sizes=(128,), learning_rate_init=0.014142000871891987,
max_iter=1143, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00021322426500482763,
hidden_layer_sizes=(32,), learning_rate_init=0.018968382984506437,
max_iter=1207, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00045392250853632627, hidden_layer_sizes=(128,
64), learning_rate_init=0.024469686118378207, max_iter=1263, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004871036742456699, hidden_layer_sizes=(64,
32), learning_rate_init=0.04262093057958416, max_iter=1127, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00019697722845676824, hidden_layer_sizes=(64,
32), learning_rate_init=0.02822746485745819, max_iter=1111, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005176178780049838, hidden_layer_sizes=(64,),
learning_rate_init=0.02960028906796679, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006893010059976614, hidden_layer_sizes=(64,
32), learning_rate_init=0.02618602313424026, max_iter=1001, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006881989502296274, hidden_layer_sizes=(64,
32), learning_rate_init=0.016777955682783428, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006137945751268555, hidden_layer_sizes=(32,),
learning_rate_init=0.03068126584617151, max_iter=1074, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00029557534312113086, hidden_layer_sizes=(64,
32), learning_rate_init=0.005663513157114246, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006620379932420407, hidden_layer_sizes=(32,),
learning_rate_init=0.03202760468157123, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00036905448579174836, hidden_layer_sizes=(128,
64), learning_rate_init=0.01397077237842239, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005176178780049838, hidden_layer_sizes=(64,),
learning_rate_init=0.02960028906796679, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00039116792456578937,
hidden_layer_sizes=(64,), learning_rate_init=0.02727205049833401, max_iter=1222,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00024234945249808001, hidden_layer_sizes=(128,
64), learning_rate_init=0.012152908368853153, max_iter=1134, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00036063661930625716, hidden_layer_sizes=(128,
64), learning_rate_init=0.025105483724207685, max_iter=1283, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00015586166068353954, hidden_layer_sizes=(64,
32), learning_rate_init=0.027331738144283906, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002779060861886791, hidden_layer_sizes=(64,),
learning_rate_init=0.03740453578716723, max_iter=1012, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00038196670839456574, hidden_layer_sizes=(128,
64), learning_rate_init=0.040339760890650354, max_iter=1232, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006620379932420407, hidden_layer_sizes=(32,),
learning_rate_init=0.03202760468157123, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005182094445971608,
hidden_layer_sizes=(128,), learning_rate_init=0.03335643987640474,
max_iter=1038, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00025249818944184327,
hidden_layer_sizes=(64,), learning_rate_init=0.040094922877118216,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00014256455019995658, hidden_layer_sizes=(64,
32), learning_rate_init=0.040464685958026395, max_iter=1142, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00039116792456578937,
hidden_layer_sizes=(64,), learning_rate_init=0.02727205049833401, max_iter=1222,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0003955106162913183, hidden_layer_sizes=(128,
64), learning_rate_init=0.014207410729662213, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00040680543931656267,
hidden_layer_sizes=(32,), learning_rate_init=0.030115776597945444,
max_iter=1019, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006881989502296274, hidden_layer_sizes=(64,
32), learning_rate_init=0.016777955682783428, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00027192475127697064,
hidden_layer_sizes=(64,), learning_rate_init=0.03941618473246701, max_iter=1179,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00039116792456578937,
hidden_layer_sizes=(64,), learning_rate_init=0.02727205049833401, max_iter=1222,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0005182094445971608,
hidden_layer_sizes=(128,), learning_rate_init=0.03335643987640474,
max_iter=1038, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036063661930625716, hidden_layer_sizes=(128,
64), learning_rate_init=0.025105483724207685, max_iter=1283, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00025249818944184327,
hidden_layer_sizes=(64,), learning_rate_init=0.040094922877118216,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00015586166068353954, hidden_layer_sizes=(64,
32), learning_rate_init=0.027331738144283906, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00040680543931656267,
hidden_layer_sizes=(32,), learning_rate_init=0.030115776597945444,
max_iter=1019, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006881989502296274, hidden_layer_sizes=(64,
32), learning_rate_init=0.016777955682783428, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00036905448579174836, hidden_layer_sizes=(128,
64), learning_rate_init=0.01397077237842239, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00024234945249808001, hidden_layer_sizes=(128,
64), learning_rate_init=0.012152908368853153, max_iter=1134, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00045392250853632627, hidden_layer_sizes=(128,
64), learning_rate_init=0.024469686118378207, max_iter=1263, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.00014256455019995658, hidden_layer_sizes=(64,
32), learning_rate_init=0.040464685958026395, max_iter=1142, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00039116792456578937,
hidden_layer_sizes=(64,), learning_rate_init=0.02727205049833401, max_iter=1222,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00025249818944184327,
hidden_layer_sizes=(64,), learning_rate_init=0.040094922877118216,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00027192475127697064,
hidden_layer_sizes=(64,), learning_rate_init=0.03941618473246701, max_iter=1179,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00040680543931656267,
hidden_layer_sizes=(32,), learning_rate_init=0.030115776597945444,
max_iter=1019, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006881989502296274, hidden_layer_sizes=(64,
32), learning_rate_init=0.016777955682783428, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005935603363957949,
hidden_layer_sizes=(128,), learning_rate_init=0.010082420506075392,
max_iter=1263, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00039116792456578937,
hidden_layer_sizes=(64,), learning_rate_init=0.02727205049833401, max_iter=1222,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00014256455019995658, hidden_layer_sizes=(64,
32), learning_rate_init=0.040464685958026395, max_iter=1142, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00025249818944184327,
hidden_layer_sizes=(64,), learning_rate_init=0.040094922877118216,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036063661930625716, hidden_layer_sizes=(128,
64), learning_rate_init=0.025105483724207685, max_iter=1283, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003955106162913183, hidden_layer_sizes=(128,
64), learning_rate_init=0.014207410729662213, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005182094445971608,
hidden_layer_sizes=(128,), learning_rate_init=0.03335643987640474,
max_iter=1038, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00015586166068353954, hidden_layer_sizes=(64,
32), learning_rate_init=0.027331738144283906, max_iter=1159, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00027192475127697064,
hidden_layer_sizes=(64,), learning_rate_init=0.03941618473246701, max_iter=1179,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00024234945249808001, hidden_layer_sizes=(128,
64), learning_rate_init=0.012152908368853153, max_iter=1134, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00038196670839456574, hidden_layer_sizes=(128,
64), learning_rate_init=0.040339760890650354, max_iter=1232, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.00025249818944184327,
hidden_layer_sizes=(64,), learning_rate_init=0.040094922877118216,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00040680543931656267,
hidden_layer_sizes=(32,), learning_rate_init=0.030115776597945444,
max_iter=1019, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0001145895798588723, hidden_layer_sizes=(32,),
learning_rate_init=0.00674415087017735, max_iter=1171, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0006881989502296274, hidden_layer_sizes=(64,
32), learning_rate_init=0.016777955682783428, max_iter=1160, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004473189373045352,
hidden_layer_sizes=(128,), learning_rate_init=0.023623920725298408,
max_iter=1062, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0005176178780049838, hidden_layer_sizes=(64,),
learning_rate_init=0.02960028906796679, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00027192475127697064,
hidden_layer_sizes=(64,), learning_rate_init=0.03941618473246701, max_iter=1179,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0005182094445971608,
hidden_layer_sizes=(128,), learning_rate_init=0.03335643987640474,
max_iter=1038, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00036063661930625716, hidden_layer_sizes=(128,
64), learning_rate_init=0.025105483724207685, max_iter=1283, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00040680543931656267,
hidden_layer_sizes=(32,), learning_rate_init=0.030115776597945444,
max_iter=1019, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006817221202684954, hidden_layer_sizes=(64,),
learning_rate_init=0.017918258917649837, max_iter=1144, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00045392250853632627, hidden_layer_sizes=(128,
64), learning_rate_init=0.024469686118378207, max_iter=1263, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005935603363957949,
hidden_layer_sizes=(128,), learning_rate_init=0.010082420506075392,
max_iter=1263, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0001145895798588723, hidden_layer_sizes=(32,),
learning_rate_init=0.00674415087017735, max_iter=1171, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005182094445971608,
hidden_layer_sizes=(128,), learning_rate_init=0.03335643987640474,
max_iter=1038, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0005176178780049838, hidden_layer_sizes=(64,),
learning_rate_init=0.02960028906796679, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00027192475127697064,
hidden_layer_sizes=(64,), learning_rate_init=0.0394161847324670l, max_iter=1179,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002779060861886791, hidden_layer_sizes=(64,),
learning_rate_init=0.03740453578716723, max_iter=1012, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001310090327011646, hidden_layer_sizes=(64,
32), learning_rate_init=0.021454156202267143, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00014256455019995658, hidden_layer_sizes=(64,
32), learning_rate_init=0.040464685958026395, max_iter=1142, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.000677589048806755, hidden_layer_sizes=(32,),
learning_rate_init=0.03253999603061466, max_iter=1128, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0004473189373045352,
hidden_layer_sizes=(128,), learning_rate_init=0.023623920725298408,
max_iter=1062, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00024234945249808001, hidden_layer_sizes=(128,
64), learning_rate_init=0.012152908368853153, max_iter=1134, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002629257494918451, hidden_layer_sizes=(64,
32), learning_rate_init=0.02991561903276001, max_iter=1195, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005176178780049838, hidden_layer_sizes=(64,),
learning_rate_init=0.02960028906796679, max_iter=1150, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002624993507572445, hidden_layer_sizes=(64,
32), learning_rate_init=0.03476170571996461, max_iter=1228, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002779060861886791, hidden_layer_sizes=(64,),
learning_rate_init=0.03740453578716723, max_iter=1012, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005935603363957949,
hidden_layer_sizes=(128,), learning_rate_init=0.010082420506075392,
max_iter=1263, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00038196670839456574, hidden_layer_sizes=(128,
64), learning_rate_init=0.040339760890650354, max_iter=1232, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0001145895798588723, hidden_layer_sizes=(32,),
learning_rate_init=0.00674415087017735, max_iter=1171, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00036063661930625716, hidden_layer_sizes=(128,
64), learning_rate_init=0.025105483724207685, max_iter=1283, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.000677589048806755, hidden_layer_sizes=(32,),
learning_rate_init=0.03253999603061466, max_iter=1128, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0004473189373045352,
hidden_layer_sizes=(128,), learning_rate_init=0.023623920725298408,
max_iter=1062, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00014256455019995658, hidden_layer_sizes=(64,
32), learning_rate_init=0.040464685958026395, max_iter=1142, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002629257494918451, hidden_layer_sizes=(64,
32), learning_rate_init=0.02991561903276001, max_iter=1195, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002779060861886791, hidden_layer_sizes=(64,),
learning_rate_init=0.03740453578716723, max_iter=1012, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0001145895798588723, hidden_layer_sizes=(32,),
learning_rate_init=0.00674415087017735, max_iter=1171, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002097195982643844, hidden_layer_sizes=(128,
64), learning_rate_init=0.016197355877837712, max_iter=1271, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005935603363957949,
hidden_layer_sizes=(128,), learning_rate_init=0.010082420506075392,
max_iter=1263, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00024234945249808001, hidden_layer_sizes=(128,
64), learning_rate_init=0.012152908368853153, max_iter=1134, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.000677589048806755, hidden_layer_sizes=(32,),
learning_rate_init=0.03253999603061466, max_iter=1128, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002624993507572445, hidden_layer_sizes=(64,
32), learning_rate_init=0.03476170571996461, max_iter=1228, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0001310090327011646, hidden_layer_sizes=(64,
32), learning_rate_init=0.021454156202267143, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002779060861886791, hidden_layer_sizes=(64,),
learning_rate_init=0.03740453578716723, max_iter=1012, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005427381500174611, hidden_layer_sizes=(128,
64), learning_rate_init=0.02946882984937409, max_iter=1246, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0004473189373045352,
hidden_layer_sizes=(128,), learning_rate_init=0.023623920725298408,
max_iter=1062, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002629257494918451, hidden_layer_sizes=(64,
32), learning_rate_init=0.02991561903276001, max_iter=1195, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006491282365758882, hidden_layer_sizes=(64,),
learning_rate_init=0.014591494366296129, max_iter=1117, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005935603363957949,
hidden_layer_sizes=(128,), learning_rate_init=0.010082420506075392,
max_iter=1263, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0003955106162913183, hidden_layer_sizes=(128,
64), learning_rate_init=0.014207410729662213, max_iter=1146, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033910284063842403,
hidden_layer_sizes=(64,), learning_rate_init=0.042965829263655435,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00016964358430414974, hidden_layer_sizes=(128,
64), learning_rate_init=0.008766279530742405, max_iter=1093, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006491282365758882, hidden_layer_sizes=(64,),
learning_rate_init=0.014591494366296129, max_iter=1117, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0002629257494918451, hidden_layer_sizes=(64,
32), learning_rate_init=0.02991561903276001, max_iter=1195, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002624993507572445, hidden_layer_sizes=(64,
32), learning_rate_init=0.03476170571996461, max_iter=1228, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00038196670839456574, hidden_layer_sizes=(128,
64), learning_rate_init=0.040339760890650354, max_iter=1232, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033910284063842403,
hidden_layer_sizes=(64,), learning_rate_init=0.042965829263655435,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0004473189373045352,
hidden_layer_sizes=(128,), learning_rate_init=0.023623920725298408,
max_iter=1062, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006877063171729051, hidden_layer_sizes=(128,
64), learning_rate_init=0.02396695316349301, max_iter=1217, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0001310090327011646, hidden_layer_sizes=(64,
32), learning_rate_init=0.021454156202267143, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005427381500174611, hidden_layer_sizes=(128,
64), learning_rate_init=0.02946882984937409, max_iter=1246, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006569911375526352, hidden_layer_sizes=(32,),
learning_rate_init=0.027110598673419598, max_iter=1094, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00042362646904003747, hidden_layer_sizes=(128,
64), learning_rate_init=0.011517377083257188, max_iter=1279, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002097195982643844, hidden_layer_sizes=(128,
64), learning_rate_init=0.016197355877837712, max_iter=1271, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0006491282365758882, hidden_layer_sizes=(64,),
learning_rate_init=0.014591494366296129, max_iter=1117, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00033910284063842403,
hidden_layer_sizes=(64,), learning_rate_init=0.042965829263655435,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002629257494918451, hidden_layer_sizes=(64,
32), learning_rate_init=0.02991561903276001, max_iter=1195, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006569911375526352, hidden_layer_sizes=(32,),
learning_rate_init=0.027110598673419598, max_iter=1094, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0002624993507572445, hidden_layer_sizes=(64,
32), learning_rate_init=0.03476170571996461, max_iter=1228, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042362646904003747, hidden_layer_sizes=(128,
64), learning_rate_init=0.011517377083257188, max_iter=1279, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002486385937006945,
hidden_layer_sizes=(128,), learning_rate_init=0.03531384441857476,
max_iter=1184, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003949695250700994, hidden_layer_sizes=(64,
32), learning_rate_init=0.011928074796400607, max_iter=1221, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0001310090327011646, hidden_layer_sizes=(64,
32), learning_rate_init=0.021454156202267143, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042362646904003747, hidden_layer_sizes=(128,
64), learning_rate_init=0.011517377083257188, max_iter=1279, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006491282365758882, hidden_layer_sizes=(64,),
learning_rate_init=0.014591494366296129, max_iter=1117, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00025653617400502835,
hidden_layer_sizes=(64,), learning_rate_init=0.03003439662856946, max_iter=1095,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00016964358430414974, hidden_layer_sizes=(128,
64), learning_rate_init=0.008766279530742405, max_iter=1093, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00048545156692378935,
hidden_layer_sizes=(32,), learning_rate_init=0.039687957342200146,
max_iter=1159, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00042362646904003747, hidden_layer_sizes=(128,
64), learning_rate_init=0.011517377083257188, max_iter=1279, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00048545156692378935,
hidden_layer_sizes=(32,), learning_rate_init=0.039687957342200146,
max_iter=1159, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006817221202684954, hidden_layer_sizes=(64,),
learning_rate_init=0.017918258917649837, max_iter=1144, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00025653617400502835,
hidden_layer_sizes=(64,), learning_rate_init=0.03003439662856946, max_iter=1095,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0002624993507572445, hidden_layer_sizes=(64,
32), learning_rate_init=0.03476170571996461, max_iter=1228, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006491282365758882, hidden_layer_sizes=(64,),
learning_rate_init=0.014591494366296129, max_iter=1117, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00048545156692378935,
hidden_layer_sizes=(32,), learning_rate_init=0.039687957342200146,
max_iter=1159, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0001310090327011646, hidden_layer_sizes=(64,
32), learning_rate_init=0.021454156202267143, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0003949695250700994, hidden_layer_sizes=(64,
32), learning_rate_init=0.011928074796400607, max_iter=1221, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00042362646904003747, hidden_layer_sizes=(128,
64), learning_rate_init=0.011517377083257188, max_iter=1279, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00048545156692378935,
hidden_layer_sizes=(32,), learning_rate_init=0.039687957342200146,
max_iter=1159, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002486385937006945,
hidden_layer_sizes=(128,), learning_rate_init=0.03531384441857476,
max_iter=1184, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00048545156692378935,
hidden_layer_sizes=(32,), learning_rate_init=0.039687957342200146,
max_iter=1159, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00025653617400502835,
hidden_layer_sizes=(64,), learning_rate_init=0.03003439662856946, max_iter=1095,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006817221202684954, hidden_layer_sizes=(64,),
learning_rate_init=0.017918258917649837, max_iter=1144, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005805695768094398, hidden_layer_sizes=(64,
32), learning_rate_init=0.020526797048260877, max_iter=1032, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.000677589048806755, hidden_layer_sizes=(32,),
learning_rate_init=0.03253999603061466, max_iter=1128, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005805695768094398, hidden_layer_sizes=(64,
32), learning_rate_init=0.020526797048260877, max_iter=1032, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005867225060416018,
hidden_layer_sizes=(128,), learning_rate_init=0.042658592351061, max_iter=1064,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005805695768094398, hidden_layer_sizes=(64,
32), learning_rate_init=0.020526797048260877, max_iter=1032, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002486385937006945,
hidden_layer_sizes=(128,), learning_rate_init=0.03531384441857476,
max_iter=1184, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005867225060416018,
hidden_layer_sizes=(128,), learning_rate_init=0.042658592351061, max_iter=1064,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005805695768094398, hidden_layer_sizes=(64,
32), learning_rate_init=0.020526797048260877, max_iter=1032, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002097195982643844, hidden_layer_sizes=(128,
64), learning_rate_init=0.016197355877837712, max_iter=1271, solver=adam; total
time=   0.1s
[CV] END activation=tanh, alpha=0.0003949695250700994, hidden_layer_sizes=(64,
32), learning_rate_init=0.011928074796400607, max_iter=1221, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006817221202684954, hidden_layer_sizes=(64,),
learning_rate_init=0.017918258917649837, max_iter=1144, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.00025653617400502835,
hidden_layer_sizes=(64,), learning_rate_init=0.03003439662856946, max_iter=1095,
solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.000677589048806755, hidden_layer_sizes=(32,),
learning_rate_init=0.03253999603061466, max_iter=1128, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005867225060416018,
hidden_layer_sizes=(128,), learning_rate_init=0.042658592351061, max_iter=1064,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005805695768094398, hidden_layer_sizes=(64,
32), learning_rate_init=0.020526797048260877, max_iter=1032, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005867225060416018,
hidden_layer_sizes=(128,), learning_rate_init=0.042658592351061, max_iter=1064,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005867225060416018,
hidden_layer_sizes=(128,), learning_rate_init=0.042658592351061, max_iter=1064,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00016964358430414974, hidden_layer_sizes=(128,
64), learning_rate_init=0.008766279530742405, max_iter=1093, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004908908631146727,
hidden_layer_sizes=(128,), learning_rate_init=0.012831645391571857,
max_iter=1011, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006817221202684954, hidden_layer_sizes=(64,),
learning_rate_init=0.017918258917649837, max_iter=1144, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0004908908631146727,
hidden_layer_sizes=(128,), learning_rate_init=0.012831645391571857,
max_iter=1011, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0004908908631146727,
hidden_layer_sizes=(128,), learning_rate_init=0.012831645391571857,
max_iter=1011, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0004908908631146727,
hidden_layer_sizes=(128,), learning_rate_init=0.012831645391571857,
max_iter=1011, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0002486385937006945,
hidden_layer_sizes=(128,), learning_rate_init=0.03531384441857476,
max_iter=1184, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00025653617400502835,
hidden_layer_sizes=(64,), learning_rate_init=0.03003439662856946, max_iter=1095,
solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0004908908631146727,
hidden_layer_sizes=(128,), learning_rate_init=0.012831645391571857,
max_iter=1011, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003949695250700994, hidden_layer_sizes=(64,
32), learning_rate_init=0.011928074796400607, max_iter=1221, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0002097195982643844, hidden_layer_sizes=(128,
64), learning_rate_init=0.016197355877837712, max_iter=1271, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003263554499367855, hidden_layer_sizes=(128,
64), learning_rate_init=0.008777718430237136, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00016964358430414974, hidden_layer_sizes=(128,
64), learning_rate_init=0.008766279530742405, max_iter=1093, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003263554499367855, hidden_layer_sizes=(128,
64), learning_rate_init=0.008777718430237136, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003263554499367855, hidden_layer_sizes=(128,
64), learning_rate_init=0.008777718430237136, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0003263554499367855, hidden_layer_sizes=(128,
64), learning_rate_init=0.008777718430237136, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00014271318907613738, hidden_layer_sizes=(128,
64), learning_rate_init=0.038795012438778184, max_iter=1089, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002486385937006945,
hidden_layer_sizes=(128,), learning_rate_init=0.03531384441857476,
max_iter=1184, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0003949695250700994, hidden_layer_sizes=(64,
32), learning_rate_init=0.011928074796400607, max_iter=1221, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005886810895533614, hidden_layer_sizes=(32,),
learning_rate_init=0.024841498171736247, max_iter=1113, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005886810895533614, hidden_layer_sizes=(32,),
learning_rate_init=0.024841498171736247, max_iter=1113, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00014271318907613738, hidden_layer_sizes=(128,
64), learning_rate_init=0.038795012438778184, max_iter=1089, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005886810895533614, hidden_layer_sizes=(32,),
learning_rate_init=0.024841498171736247, max_iter=1113, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0003263554499367855, hidden_layer_sizes=(128,
64), learning_rate_init=0.008777718430237136, max_iter=1034, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0005886810895533614, hidden_layer_sizes=(32,),
learning_rate_init=0.024841498171736247, max_iter=1113, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0005886810895533614, hidden_layer_sizes=(32,),
learning_rate_init=0.024841498171736247, max_iter=1113, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.00014271318907613738, hidden_layer_sizes=(128,
64), learning_rate_init=0.038795012438778184, max_iter=1089, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00014271318907613738, hidden_layer_sizes=(128,
64), learning_rate_init=0.038795012438778184, max_iter=1089, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00014271318907613738, hidden_layer_sizes=(128,
64), learning_rate_init=0.038795012438778184, max_iter=1089, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033910284063842403,
hidden_layer_sizes=(64,), learning_rate_init=0.042965829263655435,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006619888911134243, hidden_layer_sizes=(64,
32), learning_rate_init=0.021717841268623153, max_iter=1248, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006619888911134243, hidden_layer_sizes=(64,
32), learning_rate_init=0.021717841268623153, max_iter=1248, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020646372626783368,
hidden_layer_sizes=(32,), learning_rate_init=0.027740811349998333,
max_iter=1226, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.00020646372626783368,
hidden_layer_sizes=(32,), learning_rate_init=0.027740811349998333,
max_iter=1226, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006619888911134243, hidden_layer_sizes=(64,
32), learning_rate_init=0.021717841268623153, max_iter=1248, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020646372626783368,
hidden_layer_sizes=(32,), learning_rate_init=0.027740811349998333,
max_iter=1226, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0006619888911134243, hidden_layer_sizes=(64,
32), learning_rate_init=0.021717841268623153, max_iter=1248, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020646372626783368,
hidden_layer_sizes=(32,), learning_rate_init=0.027740811349998333,
max_iter=1226, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0002097195982643844, hidden_layer_sizes=(128,
64), learning_rate_init=0.016197355877837712, max_iter=1271, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0006619888911134243, hidden_layer_sizes=(64,
32), learning_rate_init=0.021717841268623153, max_iter=1248, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.00020646372626783368,
hidden_layer_sizes=(32,), learning_rate_init=0.027740811349998333,
max_iter=1226, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00033910284063842403,
hidden_layer_sizes=(64,), learning_rate_init=0.042965829263655435,
max_iter=1099, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006877063171729051, hidden_layer_sizes=(128,
64), learning_rate_init=0.02396695316349301, max_iter=1217, solver=adam; total
time=   0.1s
[CV] END activation=relu, alpha=0.0005389886531797841, hidden_layer_sizes=(64,
32), learning_rate_init=0.019880723431711326, max_iter=1095, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005389886531797841, hidden_layer_sizes=(64,
32), learning_rate_init=0.019880723431711326, max_iter=1095, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005389886531797841, hidden_layer_sizes=(64,
32), learning_rate_init=0.019880723431711326, max_iter=1095, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005389886531797841, hidden_layer_sizes=(64,
32), learning_rate_init=0.019880723431711326, max_iter=1095, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00016964358430414974, hidden_layer_sizes=(128,
64), learning_rate_init=0.008766279530742405, max_iter=1093, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005389886531797841, hidden_layer_sizes=(64,
32), learning_rate_init=0.019880723431711326, max_iter=1095, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004031514234687142, hidden_layer_sizes=(32,),
learning_rate_init=0.01780198404122447, max_iter=1186, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005439452562684247,
hidden_layer_sizes=(128,), learning_rate_init=0.009124954753437303,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005427381500174611, hidden_layer_sizes=(128,
64), learning_rate_init=0.02946882984937409, max_iter=1246, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005439452562684247,
hidden_layer_sizes=(128,), learning_rate_init=0.009124954753437303,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.0005439452562684247,
hidden_layer_sizes=(128,), learning_rate_init=0.009124954753437303,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006202699643258333, hidden_layer_sizes=(128,
64), learning_rate_init=0.014543874391613608, max_iter=1182, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005439452562684247,
hidden_layer_sizes=(128,), learning_rate_init=0.009124954753437303,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=tanh, alpha=0.0006202699643258333, hidden_layer_sizes=(128,
64), learning_rate_init=0.014543874391613608, max_iter=1182, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004031514234687142, hidden_layer_sizes=(32,),
learning_rate_init=0.01780198404122447, max_iter=1186, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006202699643258333, hidden_layer_sizes=(128,
64), learning_rate_init=0.014543874391613608, max_iter=1182, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004031514234687142, hidden_layer_sizes=(32,),
learning_rate_init=0.01780198404122447, max_iter=1186, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0004031514234687142, hidden_layer_sizes=(32,),
learning_rate_init=0.01780198404122447, max_iter=1186, solver=adam; total time=
0.0s
[CV] END activation=relu, alpha=0.0005439452562684247,
hidden_layer_sizes=(128,), learning_rate_init=0.009124954753437303,
max_iter=1272, solver=adam; total time=   0.0s
[CV] END activation=relu, alpha=0.00033352100724049784, hidden_layer_sizes=(64,
32), learning_rate_init=0.04334165987532793, max_iter=1075, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006202699643258333, hidden_layer_sizes=(128,
64), learning_rate_init=0.014543874391613608, max_iter=1182, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0004031514234687142, hidden_layer_sizes=(32,),
learning_rate_init=0.01780198404122447, max_iter=1186, solver=adam; total time=
0.0s
[CV] END activation=tanh, alpha=0.0006877063171729051, hidden_layer_sizes=(128,
64), learning_rate_init=0.02396695316349301, max_iter=1217, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033352100724049784, hidden_layer_sizes=(64,
32), learning_rate_init=0.04334165987532793, max_iter=1075, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006202699643258333, hidden_layer_sizes=(128,
64), learning_rate_init=0.014543874391613608, max_iter=1182, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033352100724049784, hidden_layer_sizes=(64,
32), learning_rate_init=0.04334165987532793, max_iter=1075, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005427381500174611, hidden_layer_sizes=(128,
64), learning_rate_init=0.02946882984937409, max_iter=1246, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002915881825542489, hidden_layer_sizes=(64,
32), learning_rate_init=0.027937515524931445, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002915881825542489, hidden_layer_sizes=(64,
32), learning_rate_init=0.027937515524931445, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002915881825542489, hidden_layer_sizes=(64,
32), learning_rate_init=0.027937515524931445, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033352100724049784, hidden_layer_sizes=(64,
32), learning_rate_init=0.04334165987532793, max_iter=1075, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002915881825542489, hidden_layer_sizes=(64,
32), learning_rate_init=0.027937515524931445, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.00033352100724049784, hidden_layer_sizes=(64,
32), learning_rate_init=0.04334165987532793, max_iter=1075, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0002915881825542489, hidden_layer_sizes=(64,
32), learning_rate_init=0.027937515524931445, max_iter=1143, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006877063171729051, hidden_layer_sizes=(128,
64), learning_rate_init=0.02396695316349301, max_iter=1217, solver=adam; total
time=   0.0s
[CV] END activation=relu, alpha=0.0005427381500174611, hidden_layer_sizes=(128,
64), learning_rate_init=0.02946882984937409, max_iter=1246, solver=adam; total
time=   0.0s
[CV] END activation=tanh, alpha=0.0006877063171729051, hidden_layer_sizes=(128,
64), learning_rate_init=0.02396695316349301, max_iter=1217, solver=adam; total
time=   0.0s
 Best MLP params: {'activation': 'tanh', 'alpha':
```

```
np.float64(0.0002510693774952185), 'hidden_layer_sizes': (64,),
'learning_rate_init': np.float64(0.03349082359697768), 'max_iter': 1100,
'solver': 'adam'}
```

[91]:

```python
from sklearn.neural_network import MLPRegressor

meta_model = MLPRegressor(
    hidden_layer_sizes=(512,128),
    activation='tanh',
    alpha=0.0004, # 0.0004
    learning_rate_init=0.03, # 0.04
    max_iter=1200,
    solver='adam',
    # early_stopping=True,
    # n_iter_no_change=400,
    # validation_fraction=0.2,
    random_state=42
)

meta_model.fit(X_meta_train, y_meta_train)
```

[91]:

```
MLPRegressor(activation='tanh', alpha=0.0004, hidden_layer_sizes=(512, 128),
             learning_rate_init=0.03, max_iter=1200, random_state=42)
```

[92]:

```python
# Retrain base models on full training set
cnn_full = cnn_biLSTM((trainX_wo_tweet.shape[1], trainX_wo_tweet.shape[2]), 1)
cnn_full.compile(optimizer=Adam(0.001), loss=Huber(0.1), metrics=['mae'])

transformer_full = build_transformer_model((trainX_wo_tweet.shape[1],␣
 ↪trainX_wo_tweet.shape[2]))
transformer_full.compile(optimizer=Adam(0.001), loss=Huber(0.05),␣
 ↪metrics=['mae'])

cnn_full.fit(trainX_wo_tweet, trainY_wo_tweet, epochs=50, batch_size=64,
             validation_split=0.1, verbose=0, callbacks=[early_stop_cnn,␣
 ↪reduce_lr_cnn])

transformer_full.fit(trainX_wo_tweet, trainY_wo_tweet, epochs=50, batch_size=64,
                     validation_split=0.1, verbose=0,␣
 ↪callbacks=[early_stop_tran, reduce_lr_tran])

# Predict on test
testY_pred_cnn = cnn_full.predict(testX_wo_tweet).reshape(-1, 1)
testY_pred_tran = transformer_full.predict(testX_wo_tweet).reshape(-1, 1)

X_meta_test = np.hstack([testY_pred_cnn, testY_pred_tran])
y_meta_test = testY_wo_tweet.reshape(-1)
```

```
# Final ensemble prediction
y_pred_ensemble = meta_model.predict(X_meta_test)
```

```
12/12            0s 18ms/step
12/12            0s 10ms/step
```

[93]:

```
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np

y_pred_ensemble = scaler_target.inverse_transform(y_pred_ensemble.reshape(-1,␣
 ↪1)).flatten()
y_meta_test     = scaler_target.inverse_transform(y_meta_test.reshape(-1, 1)).
 ↪flatten()

rmse = np.sqrt(mean_squared_error(y_meta_test, y_pred_ensemble))
mae = mean_absolute_error(y_meta_test, y_pred_ensemble)
r2 = r2_score(y_meta_test, y_pred_ensemble)

print(f" RMSE: {rmse:.4f}")
print(f" MAE: {mae:.4f}")
print(f" R² Score: {r2:.4f}")
```

```
 RMSE: 1.4604
 MAE: 1.1673
 R² Score: 0.8181
```

[94]:

```
# Retrain full CNN model
cnn_final = cnn_biLSTM((trainX_wo_tweet.shape[1], trainX_wo_tweet.shape[2]), 1)
cnn_final.compile(optimizer=Adam(0.001), loss=Huber(0.1), metrics=['mae'])
cnn_final.fit(trainX_wo_tweet, trainY_wo_tweet, epochs=50, batch_size=64,
              validation_split=0.1, verbose=0,
              callbacks=[early_stop_cnn, reduce_lr_cnn])

# Retrain full Transformer model
transformer_final = build_transformer_model((trainX_wo_tweet.shape[1],␣
 ↪trainX_wo_tweet.shape[2]))
transformer_final.compile(optimizer=Adam(0.001), loss=Huber(0.05),␣
 ↪metrics=['mae'])
transformer_final.fit(trainX_wo_tweet, trainY_wo_tweet, epochs=50,␣
 ↪batch_size=64,
                      validation_split=0.1, verbose=0,
                      callbacks=[early_stop_tran, reduce_lr_tran])
```

[94]:

```
<keras.src.callbacks.history.History at 0x481225290>
```

[95]:

```
# Predict from base models
pred_cnn = cnn_final.predict(testX_wo_tweet).reshape(-1, 1)
```

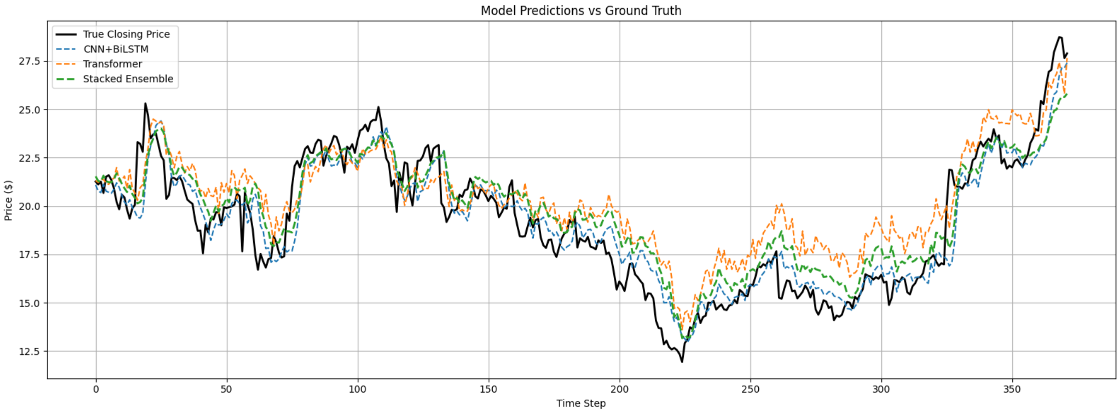
```
pred_tran = transformer_final.predict(testX_wo_tweet).reshape(-1, 1)

# Inverse transform all predictions and ground truth
pred_cnn = scaler_target.inverse_transform(pred_cnn).flatten()
pred_tran = scaler_target.inverse_transform(pred_tran).flatten()
true_test = scaler_target.inverse_transform(testY_wo_tweet.reshape(-1, 1)).
 ↪flatten()
```

```
12/12           0s 18ms/step
12/12           0s 11ms/step
```

[96]:

```
from sklearn.neural_network import MLPRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
import numpy as np
import matplotlib.pyplot as plt

def evaluate_model(name, y_true, y_pred):
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    mae = mean_absolute_error(y_true, y_pred)
    r2 = r2_score(y_true, y_pred)
    print(f"  {name}")
    print(f"   RMSE: {rmse:.4f}")
    print(f"   MAE : {mae:.4f}")
    print(f"   R²  : {r2:.4f}\n")
    return y_pred.flatten()  # return prediction for plotting

def compare_all_models(meta_model, cnn_model, transformer_model,
                       testX_wo_tweet, testY_wo_tweet, scaler_target):
    # === Predict from base models ===
    raw_pred_cnn = cnn_model.predict(testX_wo_tweet).reshape(-1, 1)
    raw_pred_tran = transformer_model.predict(testX_wo_tweet).reshape(-1, 1)

    # === Build meta input ===
    X_meta_test = np.hstack([raw_pred_cnn, raw_pred_tran])
    raw_pred_meta = meta_model.predict(X_meta_test).reshape(-1, 1)

    # === Inverse transform all ===
    pred_cnn = scaler_target.inverse_transform(raw_pred_cnn).flatten()
    pred_tran = scaler_target.inverse_transform(raw_pred_tran).flatten()
    pred_meta = scaler_target.inverse_transform(raw_pred_meta).flatten()
    y_true = scaler_target.inverse_transform(testY_wo_tweet.reshape(-1, 1)).
 ↪flatten()

    # === Evaluate ===
    y1 = evaluate_model("CNN+BiLSTM", y_true, pred_cnn)
    y2 = evaluate_model("Transformer", y_true, pred_tran)
    y3 = evaluate_model("Stacked MLP Ensemble", y_true, pred_meta)
    # === Plot all together ===
    plt.figure(figsize=(16, 6))
    plt.plot(y_true, label="True Closing Price", color="black", linewidth=2)
    plt.plot(y1, label="CNN+BiLSTM", linestyle='--')
    plt.plot(y2, label="Transformer", linestyle='--')
    plt.plot(y3, label="Stacked Ensemble", linestyle='--', linewidth=2)
    plt.title("Model Predictions vs Ground Truth")
    plt.xlabel("Time Step")
    plt.ylabel("Price ($)")
    plt.legend()
    plt.grid(True)
    plt.tight_layout()
    plt.show()

# === Call it ===
compare_all_models(
    meta_model=meta_model,
    cnn_model=cnn_final,
    transformer_model=transformer_final,
    testX_wo_tweet=testX_wo_tweet,
    testY_wo_tweet=testY_wo_tweet,
    scaler_target=scaler_target
)
```

```
12/12                 0s 3ms/step
12/12                 0s 2ms/step
 CNN+BiLSTM
  RMSE: 1.2649
  MAE : 0.9473
  R²  : 0.8635

 Transformer
  RMSE: 1.9425
  MAE : 1.6738
  R²  : 0.6781

 Stacked MLP Ensemble
  RMSE: 1.4907
  MAE : 1.2119
  R²  : 0.8104
```


Model Predictions vs Ground Truth
True Closing Price
CNN+BiLSTM
Transformer
Stacked Ensemble
Price ($)
27.5
25.0
22.5
20.0
17.5
15.0
12.5
Time Step
0
50
100
150
200
250
300
350