**Machine Learning In Finance**

**Lablog book week 5:**

import tensorflow as tf

from tensorflow import keras

from tensorflow.keras import layers

import matplotlib.pyplot as plt

# Define the CNN model with modified parameters

model = keras.Sequential([

keras.layers.Conv1D(50, 5, padding='same', input\_shape=(50, 5),

activation=tf.nn.relu, kernel\_initializer="normal"),

keras.layers.MaxPooling1D(7),

keras.layers.Conv1D(100, 5, padding='same',

activation=tf.nn.relu, kernel\_initializer="normal"),

keras.layers.GlobalMaxPooling1D(),

keras.layers.Dense(25, activation=tf.nn.relu, kernel\_initializer="normal"),

keras.layers.Dense(2) # Output: Bid and Ask

])

# Compile the model

model.compile(optimizer='adam', loss='mean\_absolute\_error', metrics=['mae'])

# Display model summary

print(model.summary())

# Train the model

history = model.fit(X\_train, y\_train,

batch\_size=50, # Changed

epochs=14, # Based on your SID 2420077

validation\_split=0.2,

verbose=1)

# Evaluate on test data

test\_loss, test\_mae = model.evaluate(X\_test, y\_test, verbose=1)

print(f"\nTest MAE: {test\_mae:.6f}")

# Predict using trained model

pred = model.predict(X\_test)

# Plot the real vs predicted values

plt.ion()

fig = plt.figure(figsize=(23, 8))

ax1 = fig.add\_subplot(111)

line1, = ax1.plot(y\_test[:400, 0], label='Bid')

line2, = ax1.plot(pred[:400, 0], label='Predicted Bid')

line3, = ax1.plot(y\_test[:400, 1], label='Ask')

line4, = ax1.plot(pred[:400, 1], label='Predicted Ask')

plt.title('Real & Predicted Normalised Prices', size=18)

plt.ylabel('Price', size=14)

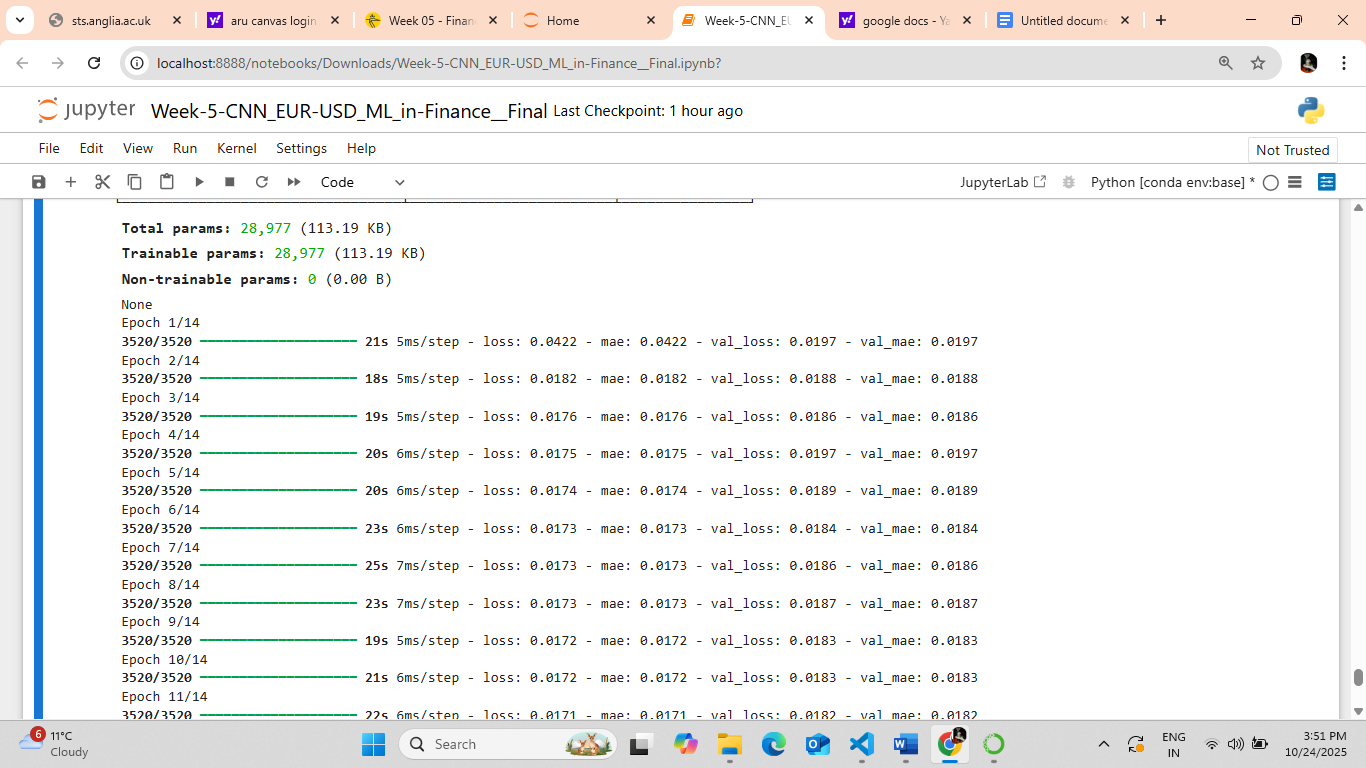
plt.xlabel('Ticks', size=14)

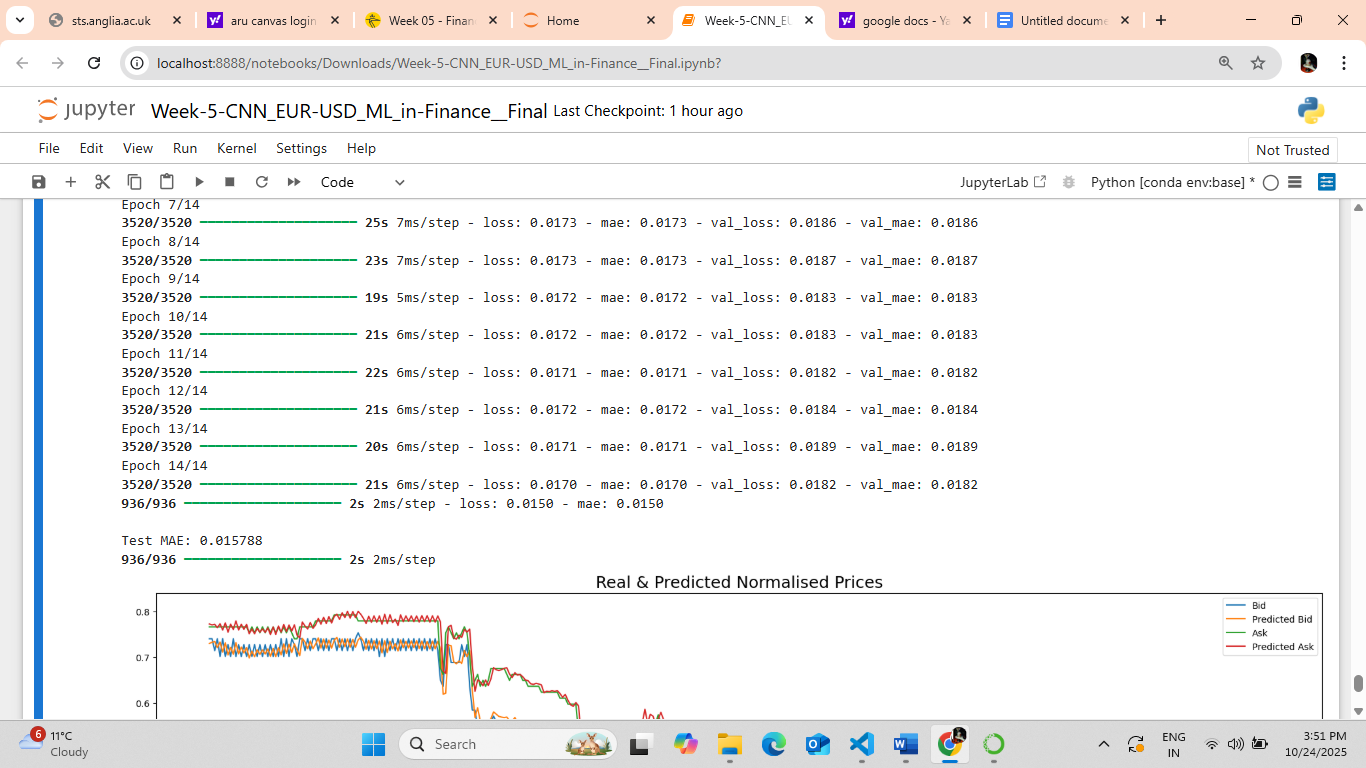
plt.legend(loc='upper right')

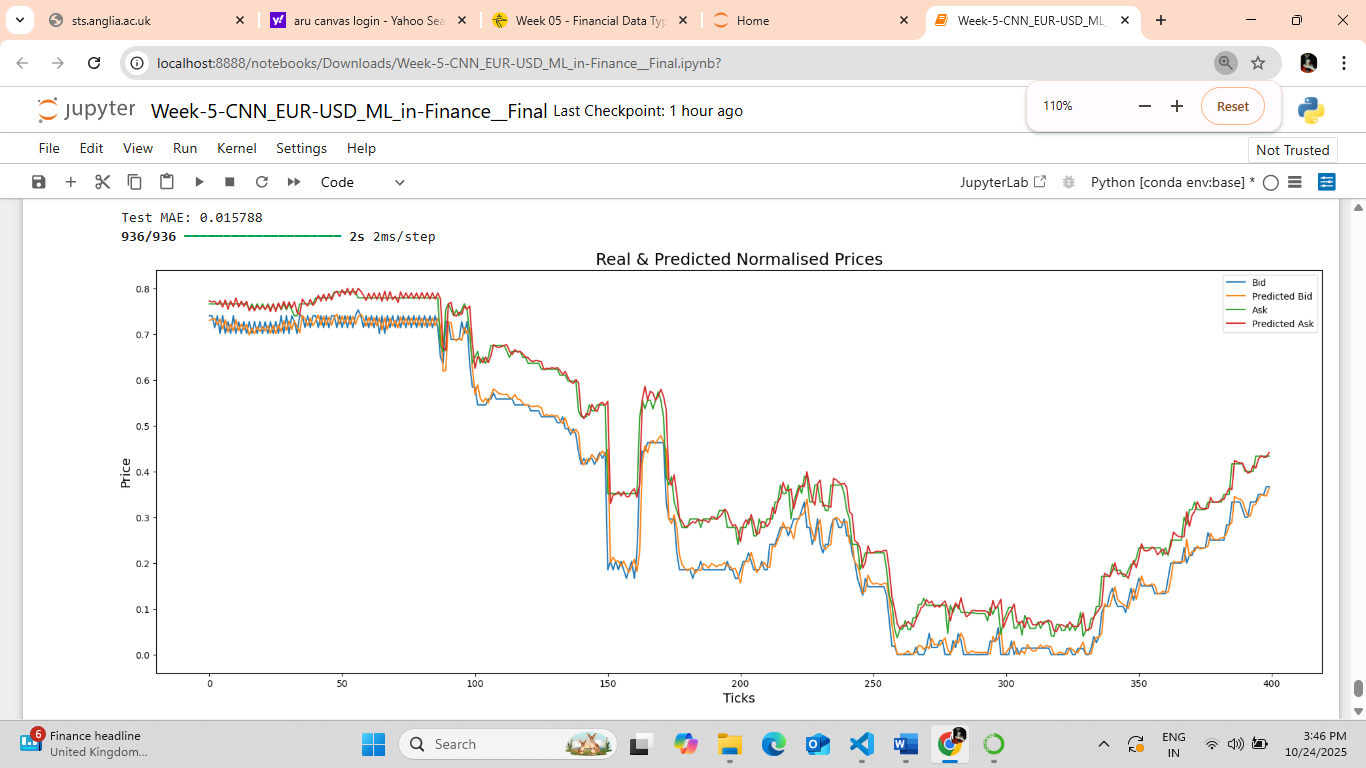
plt.show()

**Output:**

****

****

****

****