Báo cáo mô hình Neural Network

Thành viên:

- Trần Minh Tú - 22715731

Mô hình AI : Dự đoán giá bạc.

1. Viết chương trình thu thập dữ liệu từ web làm database để huấn luyện mô hình AI (tự chọn). Thực hiện các bước tiền xử lý dữ liệu (nếu có) trước khi lưu thành file .csv cuối cùng.

- Dữ liệu dùng trong chương trình là một file CSV tên ‘Silver\_data’ có kèm theo.  
*- Nguồn*: <https://vn.investing.com/commodities/silver-historical-data>

- Tiền xử lí dữ liệu: Trong dữ liệu, có một số dữ liệu thiếu có giá trị là 0; số lượng dữ liệu thiếu có thể gây ra sai số trong việc dự đoán, nên nhóm đã lọc ra những cột không mang đủ giá trị.

Dữ liệu dùng trong chương trình được thu thập từ ngày 01/01/2017 đến ngày 13/11/2023.

Data trong chương trình giảm dần theo từng năm ( Từ 2023 - > 2017 )

1. Viết chương trình huấn luyện mô hình AI sử dụng các công cụ có sẵn trên tensorflow. Đọc giá trị database trong file .csv ở câu 1. Kết quả huấn luyện phải lưu lại thành file.

- Chương trình sử dụng một số công cụ có sẵn trên Tensoflow, thư viện numpy, matplotlib,pylot và pandas.

- Dữ liệu đầu vào: Dữ liệu đầu vào như câu 1.

Code:

import numpy as np

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

import matplotlib.pyplot as plt

import pandas as pd

# load the dataset

#chuẩn bị dữ liệu (dạng cột hoặc series, dataframe)

dataset = pd.read\_csv('Silver\_data.csv')

print(dataset.shape)

# Chuyển đổi về dạng cột trong đó X là dữ liệu đầu vào, Y là dữ liệu mong muốn ở ngõ ra

X1 = dataset['n-2'].values

X2 = dataset['n-1'].values

X3 = dataset['n'].values

X4 = dataset['n+1'].values

Y = dataset['n+2'].values

XX = np.column\_stack([X1, X2, X3, X4])

print(XX.shape)

print(Y.shape)

# define the keras model

#tạo mô hình AI có 3 lớp

model = Sequential()

model.add(Dense(150, input\_shape = (4,), activation='tanh'))

model.add(Dense(150, activation ='tanh'))

model.add(Dense(100, activation ='tanh'))

model.add(Dense(1, activation = "linear"))

print(model.summary())

# compile the keras model

model.compile(loss='mse', optimizer='Adamax')

# fit the keras model on the dataset

 #train 10000 lần

model.fit(XX, Y, epochs=1000, batch\_size=128)

# evaluate the keras model

Y1 = model.predict(XX) #Dữ liệu Y1 dự báo từ mô hình

print(Y1)

df = pd.DataFrame(Y1, columns=['Predicted\_Output'])

df.to\_csv('predicted\_output.csv', index=False)

fig, ax = plt.subplots()

ax.plot(Y, 'b')

ax.plot(Y1, 'r')

plt.show()

model.save('my\_model.keras')

Kết quả dự đoán giá được lưu trong file CSV tên ‘Predicted\_output’, có đính kèm.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, phần mềm

Mô tả được tạo tự động

Kết quả huấn luyện được lưu ở file có tên : ‘my\_model.keras’, có đính kèm.

1. Thực hiện các bước phân tích dữ liệu, so sánh với kết quả dự đoán của mô hình NN.

Phân tích dữ liệu có được từ file.csv:

- Dữ liệu gồm : 5 cột và 1762 dòng.

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Cột n: giá bạc trong ngày

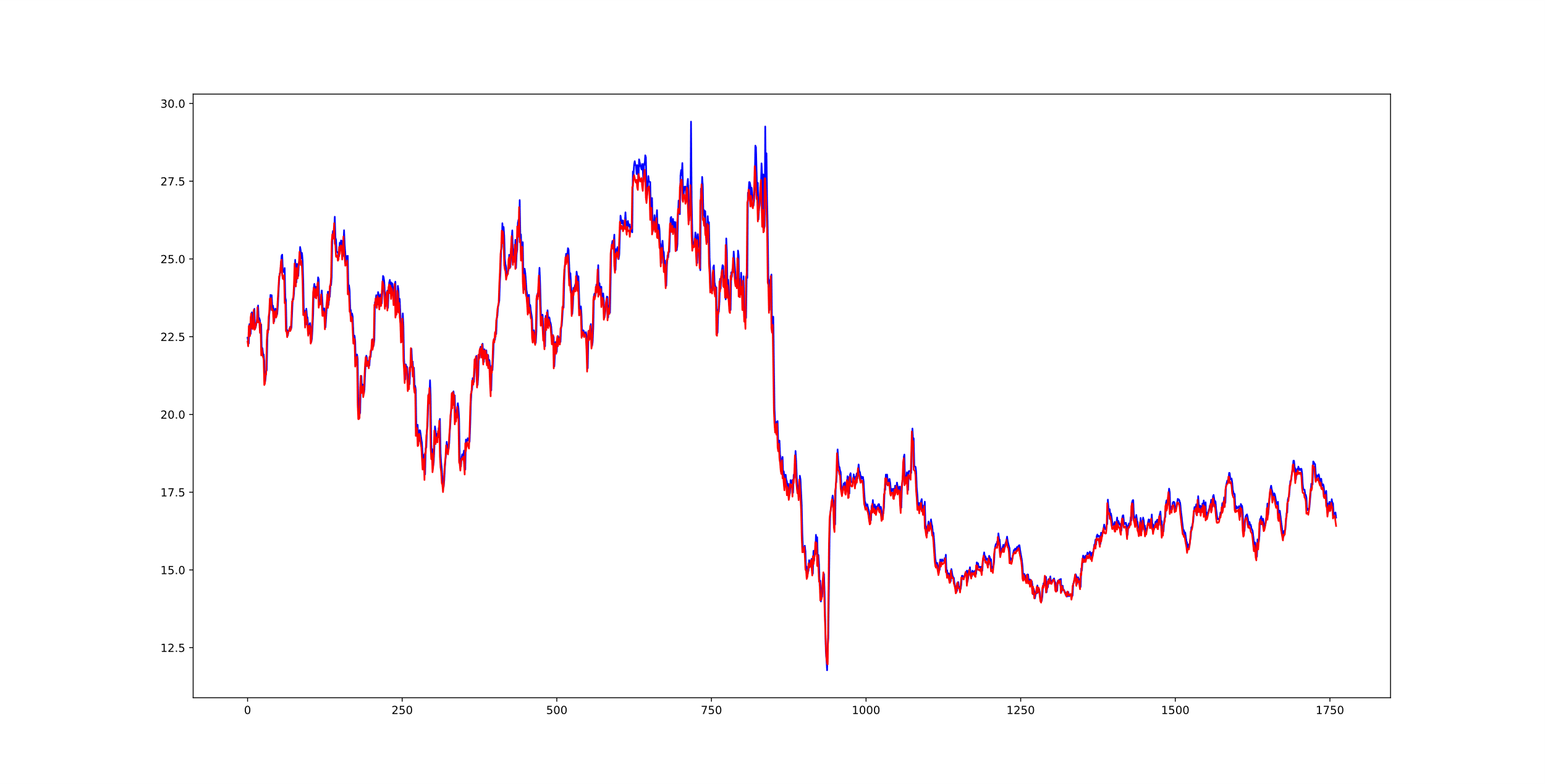
Cột n-1: giá bạc 1 ngày trước đó

Cột n-2: giá bạc 2 ngày trước đó

Cột n+1: giá bạc 1 ngày sau

Cột n+2: giá bạc 2 ngày sau

Biểu đồ so sánh giá bạc và kết quả dự đoán:



Đường màu đỏ là giá bạc thực tế. ( cột n+2 ).

Đường màu xanh là giá bạc dự đoán từ mô hình. ( cột Predicted\_output’ ).

Trục x thể hiện giá ( price ).

Trục y thể hiện các mốc thời gian ( ngày ).

Ta có thể thấy kết kết quả dự đoán còn có sự sai lệch chứ không hoàn toàn chính xác. Sự sai lệch này có thể đến từ mô hình còn sơ sài, chưa tối ưu, các trọng số chưa chính xác, vì giá bạc thay đổi theo phiên có thể là hằng ngày, cũng có thể giao động từ 1 đến 5 ngày.