//Google Stock

import numpy as np import pandas as pd

import matplotlib.pyplot as plt import yfinance as yf

from sklearn.preprocessing import MinMaxScaler from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import SimpleRNN, Dense

data = yf.download('GOOG', start='2015-01-01', end='2024-01-01') print(data.head())

close\_prices = data['Close'].values.reshape(-1, 1)

scaler = MinMaxScaler()

scaled\_close = scaler.fit\_transform(close\_prices)

X\_train = [] y\_train = []

for i in range(60, len(scaled\_close)): X\_train.append(scaled\_close[i-60:i, 0])

y\_train.append(scaled\_close[i, 0])

X\_train, y\_train = np.array(X\_train), np.array(y\_train)

X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1))

model = Sequential()

model.add(SimpleRNN(units=50, activation='tanh', return\_sequences=False, input\_shape=(X\_train.shape[1], 1))) model.add(Dense(1))

X\_train, y\_train = np.array(X\_train), np.array(y\_train)

X\_train = np.reshape(X\_train, (X\_train.shape[0], X\_train.shape[1], 1)) model.compile(optimizer='adam', loss='mean\_squared\_error')

history = model.fit(X\_train, y\_train, epochs=20, batch\_size=32)

predicted\_stock\_price = model.predict(X\_train)

predicted\_stock\_price = scaler.inverse\_transform(predicted\_stock\_price) real\_stock\_price = scaler.inverse\_transform(y\_train.reshape(-1,1))

plt.figure(figsize=(12,6))

plt.plot(real\_stock\_price, color='blue', label='Real Google Stock Price') plt.plot(predicted\_stock\_price, color='red', label='Predicted Google Stock Price') plt.title('Google Stock Price Prediction')

plt.xlabel('Time') plt.ylabel('Google Stock Price') plt.legend()

plt.show()