**STOCK MARKET PREDICTION**

**CODE:**

import numpy as np

import pandas as pd

import yfinance as yf

import matplotlib.pyplot as plt

from sklearn.preprocessing import MinMaxScaler

from sklearn.metrics import mean\_squared\_error

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import GRU, Dense, Dropout

import tensorflow as tf

# ---------------- PARAMETERS ----------------

TICKER = "AAPL"

START = "2015-01-01"

END = "2024-12-31"

LOOKBACK = 50 # past days for prediction

TEST\_RATIO = 0.25

EPOCHS = 40

BATCH\_SIZE = 32

SEED = 123

# --------------------------------------------

np.random.seed(SEED)

tf.random.set\_seed(SEED)

# ---------- Download Data ----------

def load\_stock(ticker, start, end):

df = yf.download(ticker, start=start, end=end, progress=False)

if df.empty:

raise ValueError("No data found.")

return df[["Close"]]

# ---------- Preprocessing ----------

def create\_sequences(data, lookback):

X, y = [], []

for i in range(len(data) - lookback):

X.append(data[i:i+lookback])

y.append(data[i+lookback])

return np.array(X), np.array(y)

def prepare\_dataset(df, lookback, test\_ratio):

scaler = MinMaxScaler(feature\_range=(0,1))

scaled = scaler.fit\_transform(df)

X, y = create\_sequences(scaled, lookback)

split = int(len(X) \* (1 - test\_ratio))

X\_train, X\_test = X[:split], X[split:]

y\_train, y\_test = y[:split], y[split:]

# reshape for GRU input

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

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

return X\_train, X\_test, y\_train, y\_test, scaler

# ---------- Model ----------

def build\_gru(input\_shape):

model = Sequential()

model.add(GRU(64, return\_sequences=True, input\_shape=input\_shape))

model.add(Dropout(0.3))

model.add(GRU(64))

model.add(Dropout(0.3))

model.add(Dense(1))

model.compile(optimizer="adam", loss="mse")

return model

# ---------- Plot ----------

def plot\_prediction(dates, true, pred, ticker):

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

plt.plot(dates, true, label="Actual Price")

plt.plot(dates, pred, label="Predicted Price")

plt.title(f"{ticker} Stock Price Prediction (GRU)")

plt.xlabel("Date")

plt.ylabel("Price")

plt.legend()

plt.show()

# ---------- MAIN ----------

if \_\_name\_\_ == "\_\_main\_\_":

df = load\_stock(TICKER, START, END)

df = df.dropna()

dates = df.index

X\_train, X\_test, y\_train, y\_test, scaler = prepare\_dataset(df.values, LOOKBACK, TEST\_RATIO)

test\_dates = dates[LOOKBACK + len(y\_train):]

model = build\_gru((LOOKBACK, 1))

model.summary()

history = model.fit(

X\_train, y\_train,

epochs=EPOCHS,

batch\_size=BATCH\_SIZE,

validation\_split=0.1,

verbose=1

)

# Predictions

y\_pred\_scaled = model.predict(X\_test)

y\_pred = scaler.inverse\_transform(y\_pred\_scaled)

y\_true = scaler.inverse\_transform(y\_test.reshape(-1,1))

rmse = np.sqrt(mean\_squared\_error(y\_true, y\_pred))

print(f"Test RMSE: {rmse:.4f}")

plot\_prediction(test\_dates, y\_true.flatten(), y\_pred.flatten(), TICKER)

# Next-day forecast

last\_seq = X\_test[-1].reshape(1, LOOKBACK, 1)

next\_pred = model.predict(last\_seq)

next\_price = scaler.inverse\_transform(next\_pred)[0][0]

print(f"Next predicted close price for {TICKER}: {next\_price:.2f}")

**OUTPUT:**

Model: "sequential"

Total params: 37,889 (148.00 KB)

Trainable params: 37,889 (148.00 KB)

Non-trainable params: 0 (0.00 B)

Epoch 1/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 15s 75ms/step - loss: 0.0146 - val\_loss: 5.7959e-04

Epoch 2/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 59ms/step - loss: 9.9044e-04 - val\_loss: 4.1145e-04

Epoch 3/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 73ms/step - loss: 8.5761e-04 - val\_loss: 3.9705e-04

Epoch 4/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 60ms/step - loss: 6.2095e-04 - val\_loss: 5.1721e-04

Epoch 5/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 60ms/step - loss: 6.7003e-04 - val\_loss: 3.8901e-04

Epoch 6/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 78ms/step - loss: 6.0648e-04 - val\_loss: 3.9292e-04

Epoch 7/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 5.0625e-04 - val\_loss: 4.7093e-04

Epoch 8/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 6.0036e-04 - val\_loss: 6.3560e-04

Epoch 9/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 5.1435e-04 - val\_loss: 6.5710e-04

Epoch 10/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 78ms/step - loss: 4.4805e-04 - val\_loss: 3.8172e-04

Epoch 11/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 4.7645e-04 - val\_loss: 7.8705e-04

Epoch 12/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 58ms/step - loss: 4.0648e-04 - val\_loss: 4.1241e-04

Epoch 13/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 77ms/step - loss: 5.4330e-04 - val\_loss: 0.0012

Epoch 14/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 4.1635e-04 - val\_loss: 3.6548e-04

Epoch 15/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 58ms/step - loss: 4.0568e-04 - val\_loss: 3.6368e-04

Epoch 16/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 6s 70ms/step - loss: 4.4378e-04 - val\_loss: 3.7971e-04

Epoch 17/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 58ms/step - loss: 4.3345e-04 - val\_loss: 3.8492e-04

Epoch 18/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 62ms/step - loss: 4.0861e-04 - val\_loss: 5.7920e-04

Epoch 19/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 58ms/step - loss: 4.3909e-04 - val\_loss: 5.2999e-04

Epoch 20/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 59ms/step - loss: 4.5481e-04 - val\_loss: 3.7621e-04

Epoch 21/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 6s 68ms/step - loss: 3.6640e-04 - val\_loss: 6.8428e-04

Epoch 22/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 4.5984e-04 - val\_loss: 4.2597e-04

Epoch 23/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 3.8838e-04 - val\_loss: 7.2947e-04

Epoch 24/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 7s 92ms/step - loss: 4.1375e-04 - val\_loss: 5.0792e-04

Epoch 25/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 58ms/step - loss: 3.2127e-04 - val\_loss: 9.8265e-04

Epoch 26/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 62ms/step - loss: 4.1042e-04 - val\_loss: 4.3027e-04

Epoch 27/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 5s 60ms/step - loss: 3.5152e-04 - val\_loss: 3.6833e-04

Epoch 28/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 3.5864e-04 - val\_loss: 3.4303e-04

Epoch 29/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 3.9675e-04 - val\_loss: 5.5358e-04

Epoch 30/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 78ms/step - loss: 3.5574e-04 - val\_loss: 3.9889e-04

Epoch 31/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 58ms/step - loss: 3.8598e-04 - val\_loss: 3.3404e-04

Epoch 32/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 60ms/step - loss: 4.1411e-04 - val\_loss: 4.2534e-04

Epoch 33/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 64ms/step - loss: 3.1403e-04 - val\_loss: 4.7279e-04

Epoch 34/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 72ms/step - loss: 3.8134e-04 - val\_loss: 3.2708e-04

Epoch 35/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 58ms/step - loss: 3.4361e-04 - val\_loss: 3.3351e-04

Epoch 36/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 60ms/step - loss: 3.1950e-04 - val\_loss: 0.0011

Epoch 37/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 4s 79ms/step - loss: 3.3657e-04 - val\_loss: 5.5977e-04

Epoch 38/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 60ms/step - loss: 4.0941e-04 - val\_loss: 3.3976e-04

Epoch 39/40

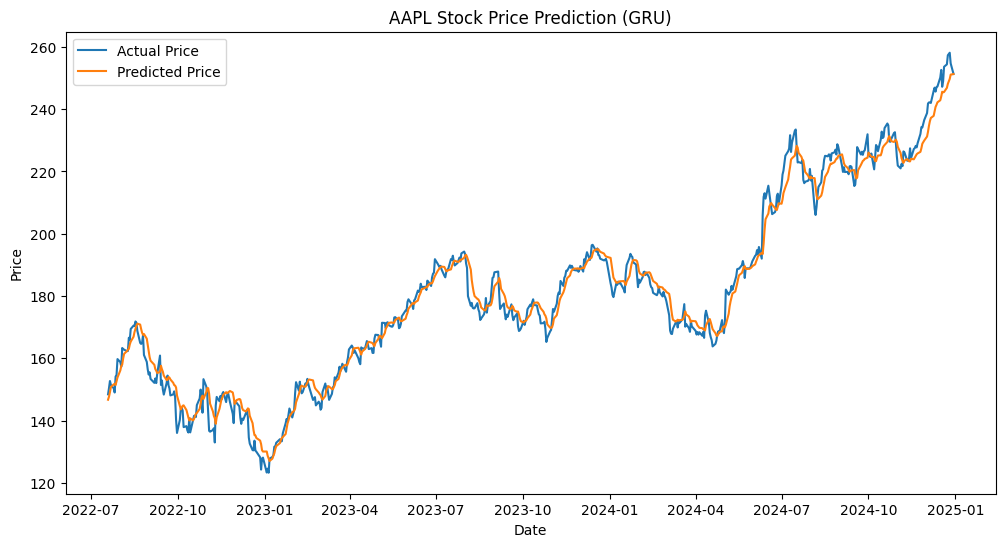
52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 59ms/step - loss: 3.4998e-04 - val\_loss: 3.3637e-04

Epoch 40/40

52/52 ━━━━━━━━━━━━━━━━━━━━ 3s 60ms/step - loss: 2.8300e-04 - val\_loss: 3.5902e-04

20/20 ━━━━━━━━━━━━━━━━━━━━ 1s 31ms/step

Test RMSE: 4.0367



1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 44ms/step

Next predicted close price for AAPL: 251.25