**Step-by-Step Explanation**

**Importing Necessary Libraries**

python

Copy code

import yfinance as yf

import numpy as np

import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from datetime import datetime, timedelta

* **yfinance**: Used to fetch stock data.
* **numpy**: For numerical operations.
* **pandas**: For data manipulation.
* **MinMaxScaler**: To scale the data for the LSTM model.
* **datetime, timedelta**: For handling date operations.

**Function Definition**

python

Copy code

def predict\_stock\_price(input\_date):

# Check if the input date is a valid date format

try:

input\_date = pd.to\_datetime(input\_date)

except ValueError:

print("Invalid Date Format. Please enter date in YYYY-MM-DD format.")

return

* The function predict\_stock\_price takes input\_date as an argument.
* It converts input\_date to a datetime object to ensure it is in the correct format.
* If the conversion fails, it prints an error message and exits the function.

**Fetch Historical Data**

python

Copy code

# Fetch data from yfinance

end\_date = input\_date

start\_date = input\_date - timedelta(days=90) # Fetch more days to ensure we have 60 trading days

data = yf.download('AAPL', start=start\_date, end=end\_date)

if len(data) < 60:

print("Not enough historical data to make a prediction. Try an earlier date.")

return

* Sets the end\_date as the input\_date and start\_date as 90 days before the input\_date to ensure there are at least 60 trading days of data.
* Downloads the historical stock data for AAPL using yfinance.
* If the data fetched is less than 60 days, it prints an error message and exits the function.

**Data Preparation**

python

Copy code

# Prepare the data

closing\_prices = data['Close'].values[-60:] # Last 60 days

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

scaled\_data = scaler.fit\_transform(closing\_prices.reshape(-1, 1))

* Extracts the closing prices for the last 60 days.
* Scales the closing prices to a range between 0 and 1 using MinMaxScaler.

**Making Predictions**

python

Copy code

# Make predictions

predicted\_prices = []

current\_batch = scaled\_data.reshape(1, 60, 1)

for i in range(4): # Predicting 4 days

next\_prediction = model.predict(current\_batch)

next\_prediction\_reshaped = next\_prediction.reshape(1, 1, 1)

current\_batch = np.append(current\_batch[:, 1:, :], next\_prediction\_reshaped, axis=1)

predicted\_prices.append(scaler.inverse\_transform(next\_prediction)[0, 0])

* Initializes an empty list predicted\_prices to store the predicted prices.
* Sets up current\_batch as the most recent 60 days of scaled data reshaped to fit the model's input requirements.
* Iteratively predicts the next 4 days:
  + Makes a prediction using the current batch.
  + Reshapes the prediction and appends it to current\_batch (removing the oldest entry to maintain the sequence length of 60).
  + Inversely transforms the predicted value back to the original scale and appends it to predicted\_prices.

**Output the Predictions**

python

Copy code

# Output the predictions

for i, price in enumerate(predicted\_prices, 1):

print(f"Day {i} prediction: {price}")

# Example use

user\_input = input("Enter a date (YYYY-MM-DD) to predict AAPL stock for the next 4 days: ")

predict\_stock\_price(user\_input)

* Iterates over the predicted\_prices and prints each prediction for the next 4 days.
* At the end of the script, prompts the user to enter a date and calls the predict\_stock\_price function with the input date.

**How it Works**

1. **User Input**: User provides a date for which they want predictions.
2. **Date Validation**: The code checks if the input date is valid.
3. **Data Fetching**: Retrieves historical stock data for the past 90 days from the input date.
4. **Data Preparation**: Extracts and scales the closing prices of the last 60 days.
5. **Prediction Loop**: Uses a pre-trained LSTM model to predict the next 4 days iteratively.
6. **Output**: Prints the predicted stock prices for the next 4 days.

This script integrates multiple steps, from fetching and preprocessing data to making predictions and presenting the results, demonstrating a complete pipeline for stock price prediction using machine learning.

4o