

The provided Python code conducts a stock price prediction using a Long Short-Term Memory (LSTM) Recursive Neural Network model. The model is trained on historical stock price data and then used to forecast future prices. Here is a brief analysis of the code's functionality and output.

#### 1. Data Retrieval:

- The user is prompted to enter a company code, which is used to construct a query string to fetch historical stock price data from Yahoo Finance API.
- The data is downloaded and stored in a pandas DataFrame.

#### 2. Data Preprocessing:

- The DataFrame is inspected, displaying the first few rows, available columns, and last few dates.
- The 'Open', 'High', 'Low', 'Close', and 'Adj Close' columns are selected for training.
- The data is standardised using the StandardScaler to normalise the dataset.

#### 3. Data Preparation:

- The training data is divided into input (trainX) and output (trainY) sequences.
- A sliding window approach is used, where a fixed number of past days (n\_past) are used to predict the future (n\_future) stock price.
- The trainX and trainY arrays are created by iterating over the dataset.

#### 4. Model Construction:

- A Sequential model from Keras is created.
- Two LSTM layers are added, with 64 and 32 units, respectively, and a ReLU activation function.
- A Dropout layer with a dropout rate of 0.2 is applied to prevent overfitting.
- A Dense layer is added to produce the predicted output.
- The model is compiled with the Adam optimizer and mean squared error (MSE) loss.

#### 5. Model Training:

- The model is trained on the trainX and trainY data.
- The training is performed for 50 epochs with a batch size of 16.
- The training and validation loss values are plotted to monitor the training progress.

#### 6. Stock Price Prediction:

- The model is used to predict future stock prices based on the past 15 days.
- The predicted values are inverse transformed using the scaler.
- The predicted dates and corresponding stock prices are stored in a DataFrame.

#### 7. Model Performance:

- The code evaluates the model's performance by calculating the mean squared error (MSE) and mean absolute error (MAE) between the predicted and actual prices.
- The model's accuracy is computed by comparing the MSE with the variance of the actual prices.

#### 9. Visualisation:

- The original stock prices and the predicted stock prices are plotted using Matplotlib.
- The original prices are shown for the past 30 days, and the predicted prices are shown for the next 60 days.

In summary, this code allows users to input a company code, select a historical data period, and specify the number of days to forecast. It trains an LSTM model, predicts future stock prices, evaluates the model's performance using MSE and MAE, and visualises the actual and forecasted prices. This analysis provides insights into the code's workflow and highlights its key steps in performing stock price prediction.