The provided Python code uses the yfinance library to analyze the historical stock data of Nifty 50 stocks. The goal is to identify stocks with positive returns based on user-specified parameters, calculate various financial metrics for the selected stocks, and compare them with the Nifty index. Finally, it visualizes the cumulative returns of the Nifty index and the combined selected stocks.

Steps followed to Achieve the desired output as mentioned.

1) Importing Libraries: The script begins by importing essential libraries. yfinance is used for fetching historical stock data, pandas for data manipulation, and matplotlib for data visualization.

2) Defining Nifty 50 Tickers: A list of Nifty 50 stock tickers is initialized to specify the stocks that will be analyzed.

3) User Input: The user is prompted to input a choice date (specific date for analysis), the value of N (parameter for the analysis period), and the initial equity.

4) Date Calculation: The script calculates the start and end dates based on the user's choice date and N, defining the analysis period.

5) Fetching Historical Data: Empty lists and DataFrames are initialized to store selected stocks and their metrics. The script then iterates through each Nifty 50 stock, downloads historical data, and calculates metrics like CAGR, Sharpe Ratio, and Volatility Percent for stocks with positive returns.

6) Nifty Index Analysis: Historical data for the Nifty index is downloaded, and metrics such as CAGR, Sharpe Ratio, and Volatility Percent are calculated.

7) Comparison and Print Metrics: The script prints the list of selected stocks with positive returns and compares overall metrics of these stocks with the Nifty index.

8) Combined Stock Data and Returns: Closing prices of selected stocks are combined, and daily and cumulative returns are calculated.

9) Plotting: Using matplotlib, the script creates a visual representation of the cumulative returns of the Nifty index and the combined selected stocks over the specified period.

10) Display Plot: The resulting plot provides a comparative view of cumulative returns for both the Nifty index and the combined selected stocks.

In [6]:

```
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt
from datetime import datetime, timedelta

nifty50_tickers = ['ASIANPAINT.NS', 'BRITANNIA.NS', 'CIPLA.NS', 'EICHERMOT.NS', 'NESTLEIN
D.NS',
    'GRASIM.NS', 'HEROMOTOCO.NS', 'HINDALCO.NS', 'HINDUNILVR.NS', 'ITC.NS',
    'LT.NS', 'M&M.NS', 'RELIANCE.NS', 'TATACONSUM.NS', 'TATAMOTORS.NS', 'TATASTEEL.NS',
    'WIPRO.NS', 'APOLLOHOSP.NS', 'DRREDDY.NS', 'TITAN.NS', 'SBIN.NS',
    'BPCL.NS', 'KOTAKBANK.NS', 'UPL.NS', 'INFY.NS', 'BAJFINANCE.NS',
    'ADANIENT.NS', 'SUNPHARMA.NS', 'JSWSTEEL.NS', 'HDFCBANK.NS', 'TCS.NS',
    'ICICIBANK.NS', 'POWERGRID.NS', 'MARUTI.NS', 'INDUSINDBK.NS', 'AXISBANK.NS',
    'HCLTECH.NS', 'ONGC.NS', 'NTPC.NS', 'COALINDIA.NS', 'BHARTIARTL.NS',
    'TECHM.NS', 'LTTS.NS', 'DIVISLAB.NS', 'ADANIPORTS.NS', 'HDFCLIFE.NS']

# Get user inputs
choice_date = input("Enter the choice date (YYYY-MM-DD): ")
N = int(input("Enter the value of N: "))
Initial_Equity = int(input("Enter Initial Equity: "))

# Calculate the start date based on the user's choice date and N
end_date = datetime.strptime(choice_date, '%Y-%m-%d').strftime('%Y-%m-%d')
```

```
start_date = (datetime.strptime(choice_date, '%Y-%m-%d') - timedelta(days=N)).strftime('
%Y-%m-%d')

# Initialize an empty DataFrame to store the data
nifty50_past_data = pd.DataFrame()

# Create a new list to store selected stocks
selected_stocks = []

# Create DataFrames to store CAGR, Sharpe Ratio, and Volatility Percent for each stock
selected_stocks_metrics = pd.DataFrame(index=nifty50_tickers, columns=['CAGR', 'Sharpe R
atio', 'Volatility Percent'])

# Loop through each stock in Nifty 50
for ticker in nifty50_tickers:
    # Download historical data for the specified time period
    stock_data = yf.download(ticker, start=start_date, end=end_date)

    # Extract the opening and closing prices
    a = stock_data['Open'].iloc[0]
    b = stock_data['Close'].iloc[-1]

    # Check if b - a > 0 and append the stock to the selected_stocks list
    if b - a > 0:
        selected_stocks.append(ticker)

        # Calculate daily returns
        stock_data['Daily_Return'] = stock_data['Close'].pct_change()

        # Calculate CAGR
        stock_start_value = stock_data['Close'].iloc[0]
        stock_end_value = stock_data['Close'].iloc[-1]
        stock_cagr = ((stock_end_value / stock_start_value) ** (1 / N)) - 1

        # Calculate Sharpe Ratio
        stock_sharpe_ratio = (252 ** 0.5) * (stock_data['Daily_Return'].mean() / stock_d
ata['Daily_Return'].std())

        # Calculate Volatility Percent
        stock_volatility_percent = (252 ** 0.5) * stock_data['Daily_Return'].std() * 100

        # Store metrics in the DataFrame
        selected_stocks_metrics.loc[ticker] = [stock_cagr, stock_sharpe_ratio, stock_vol
atility_percent]

# Print the list of selected stocks
print("Stocks with positive returns:")
print(selected_stocks)

# Download historical data for Nifty index
nifty_data = yf.download('^NSEI', start=start_date, end=end_date)

# Calculate daily returns for Nifty index
nifty_data['Daily_Return'] = nifty_data['Close'].pct_change()

# Calculate CAGR for Nifty index
nifty_start_value = nifty_data['Close'].iloc[0]
nifty_end_value = nifty_data['Close'].iloc[-1]
nifty_cagr = ((nifty_end_value / nifty_start_value) ** (1 / N)) - 1

# Calculate Sharpe Ratio for Nifty index
nifty_sharpe_ratio = (252 ** 0.5) * (nifty_data['Daily_Return'].mean() / nifty_data['Dai
ly_Return'].std())

# Calculate Volatility Percent for Nifty index
nifty_volatility_percent = (252 ** 0.5) * nifty_data['Daily_Return'].std() * 100

# Find overall metrics for selected stocks
overall_cagr = selected_stocks_metrics['CAGR'].mean()
overall_sharpe_ratio = selected_stocks_metrics['Sharpe Ratio'].mean()
overall_volatility_percent = selected_stocks_metrics['Volatility Percent'].mean()
```

```python
# Compare with Nifty index
print("\nNifty Index VS Selected Stocks: ")
print(" ")
print(f"Overall CAGR for Selected Stocks: {overall_cagr:.2%}")
print(f"Overall Sharpe Ratio for Selected Stocks: {overall_sharpe_ratio:.4f}")
print(f"Overall Volatility Percent for Selected Stocks: {overall_volatility_percent:.2f}"
)
print(f"\nNifty CAGR: {nifty_cagr:.2%}")
print(f"Nifty Sharpe Ratio: {nifty_sharpe_ratio:.4f}")
print(f"Nifty Volatility Percent: {nifty_volatility_percent:.2f}")

# Initialize an empty DataFrame to store the closing prices of selected stocks
selected_stocks_data = pd.DataFrame()

# Loop through each selected stock
for ticker in selected_stocks:
    # Download historical data for the specified time period
    stock_data = yf.download(ticker, start=start_date, end=end_date)

    # Append the closing prices to the selected_stocks_data DataFrame
    selected_stocks_data[ticker] = stock_data['Close']

# Combine the closing prices of selected stocks into one unit
combined_selected_stocks = selected_stocks_data.mean(axis=1)

# Calculate daily returns for the combined selected stocks
combined_selected_stocks_returns = combined_selected_stocks.pct_change()

# Calculate cumulative returns for the combined selected stocks
combined_selected_stocks_cumulative_returns = (1 + combined_selected_stocks_returns).cump
rod()

# Plotting the equity curves
plt.figure(figsize=(12, 6))

# Plot Nifty index equity curve
nifty_data['Cumulative_Return'] = (1 + nifty_data['Daily_Return']).cumprod()
plt.plot(nifty_data.index, nifty_data['Cumulative_Return'], label='Nifty Index', color='
blue')

# Plot combined selected stocks equity curve
plt.plot(combined_selected_stocks_cumulative_returns.index, combined_selected_stocks_cumu
lative_returns, label='Combined Selected Stocks', color='green')

plt.title(f'Equity Curves - Nifty Index vs Combined Selected Stocks - {start_date} to {en
d_date}')
plt.xlabel('Date')
plt.ylabel('Cumulative Return')
plt.legend()
plt.show()
```

```
Enter the choice date (YYYY-MM-DD): 2023-05-21
Enter the value of N: 30
Enter Initial Equity: 10000
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

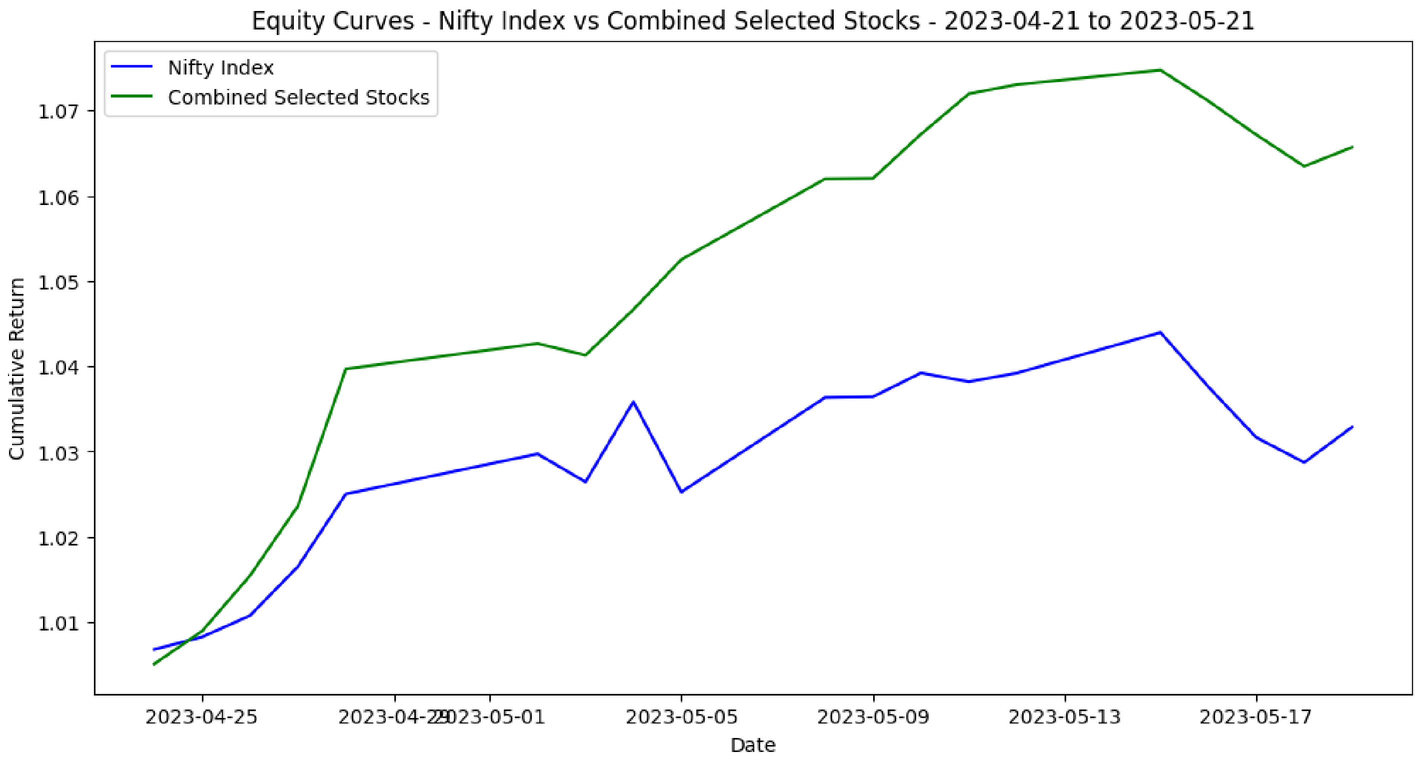
```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
Stocks with positive returns:
['ASIANPAINT.NS', 'BRITANNIA.NS', 'CIPLA.NS', 'EICHERMOT.NS', 'NESTLEIND.NS', 'GRASIM.NS'
, 'HEROMOTOCO.NS', 'HINDUNILVR.NS', 'ITC.NS', 'M&M.NS', 'RELIANCE.NS', 'TATACONSUM.NS', '
TATAMOTORS.NS', 'WIPRO.NS', 'APOLLOHOSP.NS', 'TITAN.NS', 'SBIN.NS', 'BPCL.NS', 'KOTAKBANK
.NS', 'INFY.NS', 'BAJFINANCE.NS', 'ADANIENT.NS', 'TCS.NS', 'ICICIBANK.NS', 'POWERGRID.NS'
, 'MARUTI.NS', 'INDUSINDBK.NS', 'AXISBANK.NS', 'HCLTECH.NS', 'ONGC.NS', 'NTPC.NS', 'COALI
NDIA.NS', 'BHARTIARTL.NS', 'TECHM.NS', 'LTTS.NS', 'ADANIPORTS.NS', 'HDFCLIFE.NS']
[*********************100%%**********************]  1 of 1 completed

Nifty Index VS Selected Stocks:

Overall CAGR for Selected Stocks: 0.19%
Overall Sharpe Ratio for Selected Stocks: 4.0268
Overall Volatility Percent for Selected Stocks: 19.14

Nifty CAGR: 0.11%
Nifty Sharpe Ratio: 4.8809
Nifty Volatility Percent: 8.87
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```



Extra Work

1) By giving a particular eqiuty as input,Find how many best selected stocks can be purchased(Let's say each stock should be purchased only once) so that the difference off equity and after purchasing the stocks (remaining money) should be minimum.

Steps to be followed:

1) If the given input day is "Saturday" or "Sunday", We cannot find the closing prices of stocks, So I have the date of it's nearest friday

Note: If the input day is any week day then there is scope for changing the date.

In [8]:

```
import datetime
get_day_of_week = lambda choice_date: datetime.datetime.strptime(choice_date, '%Y-%m-%d')
.strftime('%A')
day_of_week = get_day_of_week(choice_date)

from datetime import datetime, timedelta
if day_of_week == "Saturday" or day_of_week == "Sunday":
  input_datetime = datetime.strptime(choice_date, '%Y-%m-%d')
  weekday = input_datetime.weekday()
  if weekday == 5:   # Saturday
        days_to_friday = 1
  else:  # Sunday
        days_to_friday = 2
  previous_friday_date = input_datetime - timedelta(days=days_to_friday)
  end_date = previous_friday_date.strftime('%Y-%m-%d')
  print(end_date)
```

2023-05-19

2) Printing Prices of Selected Stocks.

In [12]:

```python
end_date2 = end_date
from datetime import datetime, timedelta
end_date2 = datetime.strptime(end_date2, '%Y-%m-%d')
next_day_date = end_date2 + timedelta(days=1)

# Convert the next day's date back to a string in the same format
next_day_date_str = next_day_date.strftime('%Y-%m-%d')

import yfinance as yf
selected_stocks_price = []
for i in selected_stocks:
  data = yf.download(i,end_date,next_day_date_str)
  selected_stocks_price.append(data["Close"][0])
print(selected_stocks_price)
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[3084.449951171875, 4499.85009765625, 916.25, 3570.89990234375, 21690.150390625, 1716.099
9755859375, 2713.0, 2641.449951171875, 419.8500061035156, 1260.4000244140625, 2441.949951
171875, 765.4500122070312, 524.9500122070312, 386.20001220703125, 4447.14990234375, 2701.
64990234375, 575.1500244140625, 360.29998779296875, 1941.6500244140625, 1268.900024414062
5, 6784.2001953125, 1956.050048828125, 3222.85009765625, 954.2999877929688, 175.274993896
48438, 9105.9501953125, 1248.0, 924.0499877929688, 1095.3499755859375, 164.89999389648438
, 173.3000030517578, 239.89999389648438, 805.75, 1072.1500244140625, 3837.35009765625, 68
8.0999755859375, 557.5999755859375]
```

3) Finding all the possible purchase within the Intial_Equity and returning the best purhase (maximum 1 stock each company).

In [17]:

```
target_sum = Initial_Equity

# Function to find all subarrays with sum less than or equal to target_sum
def find_subarrays(selected_stocks_price, target_sum):
    result = []
    current_sum = 0
    start = 0

    for end in range(len(selected_stocks_price)):
        current_sum += selected_stocks_price[end]

        while current_sum > target_sum:
            current_sum -= selected_stocks_price[start]
            start += 1

        result.append(selected_stocks_price[start:end + 1])

    return result

# Find all subarrays
subarrays = find_subarrays(selected_stocks_price, target_sum)

# Find the subarray with the minimum difference to target_sum
min_diff = float('inf')
min_diff_subarray = None

for subarray in subarrays:
    current_diff = abs(target_sum - sum(subarray))
    if current_diff < min_diff:
        min_diff = current_diff
        min_diff_subarray = subarray


print("\nThe subarray with the minimum difference to {}: ".format(Initial_Equity))
print(" ")
print(min_diff_subarray)

best_pick_stocks = []

for i in min_diff_subarray:
  x = selected_stocks_price.index(i)
  stock = selected_stocks[x]

  best_pick_stocks.append(stock)

print(best_pick_stocks)
```

```
The subarray with the minimum difference to 10000:

[1941.6500244140625, 1268.9000244140625, 6784.2001953125]
['KOTAKBANK.NS', 'INFY.NS', 'BAJFINANCE.NS']
```

Extra Work

2) Predicting of the any Stock Prices for next 30 days (In this Case I have taken Amazon stock dataset from kaggle).

This Python script employs Long Short-Term Memory (LSTM) neural networks to predict stock prices for Amazon using historical stock data. The main steps involved in the code are as follows:

1)Data Preprocessing: The script begins by loading historical stock data for Amazon from a CSV file using the pandas library. It then uses the MinMaxScaler from scikit-learn to scale the stock data to a range between 0 and 1. This scaling is crucial for the neural network to better learn from the data.

2) Data Splitting: The dataset is split into training and testing sets. Approximately 65% of the data is used for training, and the remaining 35% is used for testing the LSTM model.

3) Creating Time Series Dataset: A custom function, create_dataset, is defined to convert the stock price data into a time series dataset. This function organizes the data into input sequences (X) and corresponding output values (Y) based on a specified time step. This is crucial for training the LSTM model.

4) Reshaping Data for LSTM: The input data is reshaped to fit the format required for LSTM models, which is in the form of [samples, time steps, features].

5) Building the LSTM Model: The script uses the Sequential model from the tensorflow.keras library to create a stacked LSTM model. Three LSTM layers with 50 units each are stacked, followed by a dense layer with one unit. The model is compiled with the mean squared error loss function and the Adam optimizer.

6) Training the Model: The model is trained using the training data, and the training process is validated using the testing data. The training is performed over 200 epochs with a batch size of 64.

7) Making Predictions: Once the model is trained, it is used to make predictions on both the training and testing datasets. These predictions are initially in the scaled format.

8) Inverse Scaling: The predictions are then transformed back to the original scale using the inverse of the MinMaxScaler. This step is crucial to interpret the predictions in the context of the original stock prices.

9) Performance Evaluation: The script calculates the Root Mean Squared Error (RMSE) for both the training and testing predictions. RMSE is a metric used to evaluate the accuracy of the model's predictions.

10) Plotting Results: The script plots the original stock prices, the predicted prices for the training set, and the predicted prices for the testing set. This visual representation helps assess the model's performance.

11) Forecasting Future Stock Prices: The script utilizes the trained model to forecast stock prices for the next 30 days. It uses a rolling prediction approach where each predicted value is fed back into the model for the subsequent prediction.

12) Plotting Future Stock Prices: Finally, the script plots the original stock prices along with the predicted prices for both the historical and forecasted periods. This provides a visual representation of the model's ability to capture trends in the stock price data.

In summary, this code demonstrates the application of LSTM neural networks for stock price prediction, including data preprocessing, model training, performance evaluation, and forecasting future stock prices. The visualizations help users understand the model's performance and its ability to generalize to unseen data.

In [38]:

```
!gdown --id 11qiYzM0Crwo2OH-azL76SPWiyb7IX-p4
```

```
/usr/local/lib/python3.10/dist-packages/gdown/cli.py:121: FutureWarning: Option `--id` wa
s deprecated in version 4.3.1 and will be removed in 5.0. You don't need to pass it anymo
re to use a file ID.
  warnings.warn(
Downloading...
From: https://drive.google.com/uc?id=11qiYzM0Crwo2OH-azL76SPWiyb7IX-p4
To: /content/Amazon_Stock.csv
100% 70.0k/70.0k [00:00<00:00, 55.8MB/s]
```

In [39]:

```
import pandas as pd
```

In [40]:

```
data = pd.read_csv("Amazon_Stock.csv")
data.head()
```

Out[40]:

	Date	Close/Last	Volume	Open	High	Low
0	09/01/2023	$138.12	40991540	$139.455	$139.96	$136.875
1	08/31/2023	$138.01	58781310	$135.06	$138.7885	$135.00

	Date	Close/Last	Volume	Open	High	Low
2	08/30/2023	$135.07	36137020	$134.93	$135.68	$133.92
3	08/29/2023	$134.91	38646090	$133.38	$135.14	$133.25
4	08/28/2023	$133.14	34108410	$133.78	$133.95	$131.85

In [41]:

```
data.tail()
```

Out[41]:

	Date	Close/Last	Volume	Open	High	Low
1253	09/10/2018	$96.9505	89621820	$98.55	$98.652	$96.5758
1254	09/07/2018	$97.6035	97139640	$96.9355	$98.76	$96.8675
1255	09/06/2018	$97.9155	149461720	$100.3253	$100.375	$96.7605
1256	09/05/2018	$99.741	163942340	$101.9055	$102.019	$99.4945
1257	09/04/2018	$101.9755	114062560	$101.325	$102.525	$100.65

In [42]:

```
data = data["Close/Last"]
data.head()
```

Out[42]:

```
0    $138.12
1    $138.01
2    $135.07
3    $134.91
4    $133.14
Name: Close/Last, dtype: object
```

In [43]:

```
sum(data.isnull())
```

Out[43]:

```
0
```

In [44]:

```
data = data.str.replace("$","")
```

```
<ipython-input-44-8bf1902f8e5a>:1: FutureWarning: The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.
  data = data.str.replace("$","")
```

In [27]:

```
import numpy as np
```

In [46]:

```
from sklearn.preprocessing import MinMaxScaler
scaler=MinMaxScaler(feature_range=(0,1))
data=scaler.fit_transform(np.array(data).reshape(-1,1))
```

In [47]:

```
print(data)
```

```
[[0.59412344]
 [0.59320195]
 [0.56857316]
 ...
 [0.25732476]
```

```
 [0.25752476]
 [0.27261723]
 [0.29133594]]
```

In [48]:

```
##splitting dataset into train and test split
training_size=int(len(data)*0.65)
test_size=len(data)-training_size
train_data,test_data=data[0:training_size,:],data[training_size:len(data),:1]
```

In [49]:

```
training_size,test_size
```

Out[49]:

```
(817, 441)
```

In [50]:

```
train_data
```

Out[50]:

```
array([[0.59412344],
       [0.59320195],
       [0.56857316],
       [0.56723282],
       [0.55240529],
       [0.55341054],
       [0.54151501],
       [0.57234288],
       [0.56170391],
       [0.56530608],
       [0.55307546],
       [0.55944208],
       [0.56857316],
       [0.59035372],
       [0.61464743],
       [0.59655281],
       [0.59780938],
       [0.59186161],
       [0.60936983],
       [0.62846971],
       [0.60627029],
       [0.51696999],
       [0.51110599],
       [0.54025843],
       [0.55692894],
       [0.54461455],
       [0.51144108],
       [0.51060336],
       [0.51881296],
       [0.5160485 ],
       [0.52610107],
       [0.52576598],
       [0.57100253],
       [0.54980837],
       [0.55592368],
       [0.56530608],
       [0.56212277],
       [0.53280278],
       [0.51588096],
       [0.50205868],
       [0.5242581 ],
       [0.51236256],
       [0.52928438],
       [0.52794404],
       [0.52911684],
       [0.50850908],
       [0.51805902],
       [0.51923182],
```

```
       [0.50373411],
       [0.52048839],
       [0.52735764],
       [0.48279126],
       [0.49074954],
       [0.48832017],
       [0.50189114],
       [0.49611091],
       [0.49812143],
       [0.49736748],
       [0.47106327],
       [0.47793252],
       [0.45263356],
       [0.49770257],
       [0.48672852],
       [0.47793252],
       [0.46553436],
       [0.44718842],
       [0.45623573],
       [0.44325117],
       [0.40044399],
       [0.41510398],
       [0.40036022],
       [0.40052776],
       [0.41091541],
       [0.42683198],
       [0.40463256],
       [0.38704057],
       [0.36861086],
       [0.36073635],
       [0.37682046],
       [0.36014995],
       [0.33024357],
       [0.32362563],
       [0.32215963],
       [0.30829546],
       [0.30536346],
       [0.30519592],
       [0.29196004],
       [0.32044231],
       [0.35705041],
       [0.31650506],
       [0.29631615],
       [0.32680894],
       [0.33309179],
       [0.30670381],
       [0.3108086 ],
       [0.29405433],
       [0.29774027],
       [0.29581352],
       [0.29489204],
       [0.25660852],
       [0.27411674],
       [0.2929653 ],
       [0.29204381],
       [0.28400176],
       [0.3078766 ],
       [0.29497581],
       [0.30234769],
       [0.29154118],
       [0.27688119],
       [0.251666  ],
       [0.25836771],
       [0.25912166],
       [0.2639804 ],
       [0.26389663],
       [0.27989696],
       [0.25560326],
       [0.26599091],
       [0.27512199],
       [0.24295378],
       [0.23189596],
```

```
       [0.21137197],
       [0.19713083],
       [0.20986408],
       [0.2238539 ],
       [0.22075436],
       [0.22242979],
       [0.2320635 ],
       [0.20885882],
       [0.20919391],
       [0.22645082],
       [0.22251356],
       [0.2203355 ],
       [0.23977047],
       [0.23951915],
       [0.22938281],
       [0.25133092],
       [0.2592892 ],
       [0.28450439],
       [0.27227376],
       [0.27093342],
       [0.25476554],
       [0.26004314],
       [0.27520576],
       [0.29246267],
       [0.29304907],
       [0.30318541],
       [0.38293577],
       [0.31792917],
       [0.30100735],
       [0.27939433],
       [0.2935517 ],
       [0.26825274],
       [0.25116338],
       [0.24395904],
       [0.2540116 ],
       [0.25174977],
       [0.22184339],
       [0.2367547 ],
       [0.24169721],
       [0.25903789],
       [0.23516304],
       [0.23365516],
       [0.18992649],
       [0.16889987],
       [0.15817713],
       [0.1333808 ],
       [0.15030262],
       [0.15599908],
       [0.14075269],
       [0.14226057],
       [0.12249052],
       [0.13271063],
       [0.15122411],
       [0.13899349],
       [0.16395736],
       [0.15072148],
       [0.14845965],
       [0.17308844],
       [0.17803095],
       [0.2042514 ],
       [0.21187459],
       [0.19562294],
       [0.18339232],
       [0.19394752],
       [0.17811472],
       [0.17635553],
       [0.19947643],
       [0.2256131 ],
       [0.23708978],
       [0.24580201],
       [0.21128819],
       [0.22410522],
```

```
       [0.21958156],
       [0.2256131 ],
       [0.21782236],
       [0.21162328],
       [0.22569687],
       [0.23164464],
       [0.25066075],
       [0.26590714],
       [0.26213743],
       [0.28140485],
       [0.24655595],
       [0.15867976],
       [0.19084798],
       [0.1954554 ],
       [0.19922511],
       [0.18515152],
       [0.20877505],
       [0.24789629],
       [0.29522713],
       [0.30335295],
       [0.36660035],
       [0.4059729 ],
       [0.44735597],
       [0.4408218 ],
       [0.43663323],
       [0.40253827],
       [0.40103039],
       [0.4118369 ],
       [0.39030765],
       [0.33258916],
       [0.37975246],
       [0.382852  ],
       [0.37707177],
       [0.38930239],
       [0.39675805],
       [0.44484282],
       [0.45028796],
       [0.45146076],
       [0.40781587],
       [0.38368971],
       [0.39876856],
       [0.42565918],
       [0.39550148],
       [0.40170056],
       [0.39022388],
       [0.41979518],
       [0.43009906],
       [0.46067562],
       [0.48136715],
       [0.47190098],
       [0.49493811],
       [0.51395422],
       [0.49946177],
       [0.58013362],
       [0.55349431],
       [0.52459319],
       [0.52174496],
       [0.493514  ],
       [0.50524199],
       [0.50783891],
       [0.49904291],
       [0.5154621 ],
       [0.52434187],
       [0.53238392],
       [0.58708664],
       [0.5579342 ],
       [0.55642631],
       [0.55307546],
       [0.59504492],
       [0.62913988],
       [0.62746445],
       [0.64991518],
```

```
       [0.63651176],
       [0.6396113 ],
       [0.61523383],
       [0.63240696],
       [0.59169407],
       [0.60492995],
       [0.61657417],
       [0.6314017 ],
       [0.60585143],
       [0.56094997],
       [0.57125385],
       [0.56756791],
       [0.46142956],
       [0.45045551],
       [0.39885233],
       [0.45187962],
       [0.46260236],
       [0.48111583],
       [0.46553436],
       [0.4273346 ],
       [0.39005634],
       [0.38829714],
       [0.36383589],
       [0.36190915],
       [0.35202413],
       [0.37321829],
       [0.40496764],
       [0.41158558],
       [0.39483131],
       [0.38787828],
       [0.35487235],
       [0.32680894],
       [0.34951098],
       [0.33677773],
       [0.38553268],
       [0.41267461],
       [0.37899851],
       [0.3497623 ],
       [0.34750047],
       [0.32689271],
       [0.30544723],
       [0.33903956],
       [0.2941381 ],
       [0.30553101],
       [0.3556263 ],
       [0.4100777 ],
       [0.45221471],
       [0.4674611 ],
       [0.48245618],
       [0.46201596],
       [0.48849609],
       [0.45643678],
       [0.44408469],
       [0.40167124],
       [0.36758466],
       [0.33154202],
       [0.30913318],
       [0.33809294],
       [0.33837777],
       [0.33609919],
       [0.33436931],
       [0.40353096],
       [0.36534797],
       [0.38415045],
       [0.33284467],
       [0.3197889 ],
       [0.34899998],
       [0.34841358],
       [0.39853819],
       [0.41223062],
       [0.49199355],
       [0.47796184],
```

```
        [0.48002681],
        [0.4781964 ],
        [0.64837798],
        [0.59451716],
        [0.60477078],
        [0.6607552 ],
        [0.64631301],
        [0.6793692 ],
        [0.72713565],
        [0.76162852],
        [0.71697418],
        [0.70793943],
        [0.74006157],
        [0.70024084],
        [0.703043  ],
        [0.73101007],
        [0.75885568],
        [0.76699407],
        [0.81138453],
        [0.84733502],
        [0.80723785],
        [0.80252571],
        [0.83019959],
        [0.85544828],
        [0.8527299 ],
        [0.81740351],
        [0.8079876 ],
        [0.80596452],
        [0.81837107],
        [0.78990974],
        [0.78789085],
        [0.75428595],
        [0.71964648],
        [0.67158265],
        [0.6253953 ],
        [0.65615196],
        [0.6669836 ],
        [0.60383254],
        [0.57648537],
        [0.58853589],
        [0.6571279 ],
        [0.67603929],
        [0.71083792],
        [0.70321054],
        [0.72349159],
        [0.72538064],
        [0.70502   ],
        [0.65030891],
        [0.69529833],
        [0.71543697],
        [0.73261848],
        [0.76150286],
        [0.74818321],
        [0.73692852],
        [0.72123395],
        [0.76906742],
        [0.78737984],
        [0.78925632],
        [0.76012063],
        [0.757641  ],
        [0.60020105],
        [0.69877484],
        [0.70364196],
        [0.690071  ],
        [0.64319672],
        [0.60683575],
        [0.60042723],
        [0.60975518],
        [0.64793818],
        [0.63201324],
        [0.70761272],
        [0.74641144],
```

```
[0.76834698],
[0.79532556],
[0.78758508],
[0.821035  ],
[0.82233345],
[0.78986366],
[0.79881045],
[0.80467444],
[0.81391443],
[0.84042807],
[0.86457517],
[0.83368448],
[0.84983141],
[0.85449329],
[0.86672391],
[0.85841798],
[0.87013759],
[0.86987371],
[0.86467989],
[0.836717  ],
[0.86133322],
[0.85172883],
[0.88895684],
[0.85357599],
[0.85756351],
[0.87971685],
[0.89612767],
[0.91277304],
[0.91282749],
[0.87265074],
[0.85691009],
[0.87683512],
[0.87949905],
[0.90603363],
[0.92886134],
[0.9049823 ],
[0.9367526 ],
[0.93659763],
[0.93346876],
[0.97702989],
[0.98519341],
[0.92359631],
[0.92011979],
[0.9222057 ],
[0.91360657],
[0.89155375],
[0.89555383],
[0.93500178],
[0.89845651],
[0.91102641],
[0.89343861],
[0.85448491],
[0.82464135],
[0.82688643],
[0.84963874],
[0.88069279],
[0.85804101],
[0.85116338],
[0.82783304],
[0.83419129],
[0.8758508 ],
[0.86749461],
[0.87967916],
[0.880764  ],
[0.86496471],
[0.81924229],
[0.8127165 ],
[0.79723973],
[0.79680831],
[0.81453434],
[0.82031875],
[0.80338855],
```

```
       [0.78621123],
       [0.77313452],
       [0.81228926],
       [0.81303483],
       [0.81977005],
       [0.82598588],
       [0.86361599],
       [0.87187585],
       [0.86788833],
       [0.85283043],
       [0.83757566],
       [0.84264383],
       [0.88737356],
       [0.89814656],
       [0.89293179],
       [0.88212947],
       [0.88513267],
       [0.89015058],
       [0.89643762],
       [0.91375317],
       [0.9069635 ],
       [0.89387841],
       [0.88762487],
       [0.89427632],
       [0.8908375 ],
       [0.87022137],
       [0.8400888 ],
       [0.82600264],
       [0.81895747],
       [0.82172192],
       [0.80500534],
       [0.77739429],
       [0.77228424],
       [0.77792624],
       [0.79499047],
       [0.81887788],
       [0.81677522],
       [0.82076693],
       [0.81599615],
       [0.82796289],
       [0.83683847],
       [0.83812436],
       [0.85112987],
       [0.84222078],
       [0.84704601],
       [0.83248654],
       [0.83085719],
       [0.9449245 ],
       [0.95765775],
       [0.95601164],
       [0.98676831],
       [0.96868207],
       [0.96088714],
       [0.93875893],
       [0.93372846],
       [0.92384343],
       [0.93391275],
       [0.95802635],
       [0.97917024],
       [0.97736078],
       [0.9946135 ],
       [0.9949444 ],
       [1.        ],
       [0.98541121],
       [0.97668223],
       [0.90767136],
       [0.87499634],
       [0.87800792],
       [0.88135039],
       [0.87957025],
       [0.86179815],
       [0.88174412],
```

```
       [0.90467235],
       [0.9053509 ],
       [0.88378814],
       [0.89758529],
       [0.89856541],
       [0.86757419],
       [0.85412051],
       [0.85443046],
       [0.838916  ],
       [0.84009717],
       [0.81140547],
       [0.80426815],
       [0.77658171],
       [0.78002052],
       [0.77197428],
       [0.79165218],
       [0.78522692],
       [0.78707826],
       [0.79002702],
       [0.80470795],
       [0.80214874],
       [0.79625961],
       [0.77870531],
       [0.79738633],
       [0.79073488],
       [0.79093594],
       [0.80689857],
       [0.78700706],
       [0.76127668],
       [0.75728497],
       [0.7874301 ],
       [0.7734319 ],
       [0.81578672],
       [0.82196905],
       [0.8069614 ],
       [0.82427276],
       [0.85552786],
       [0.88942596],
       [0.89105531],
       [0.88568975],
       [0.8684873 ],
       [0.86495633],
       [0.8364238 ],
       [0.82308739],
       [0.84527844],
       [0.83383107],
       [0.84946282],
       [0.86095206],
       [0.85242832],
       [0.83312321],
       [0.86118662],
       [0.85255398],
       [0.8495424 ],
       [0.81900773],
       [0.81066829],
       [0.78739241],
       [0.78861128],
       [0.76107981],
       [0.73304991],
       [0.71680245],
       [0.72536388],
       [0.71543697],
       [0.71302017],
       [0.73011372],
       [0.75123668],
       [0.74008251],
       [0.72504136],
       [0.70536765],
       [0.7504953 ],
       [0.73212004],
       [0.72785608],
       [0.73112735],
```

```
       [0.74122181],
       [0.71778676],
       [0.719969  ],
       [0.67351777],
       [0.69383652],
       [0.68424888],
       [0.69573813],
       [0.73323839],
       [0.7548556 ],
       [0.73256822],
       [0.71758571],
       [0.76046409],
       [0.77511152],
       [0.76934805],
       [0.7983162 ],
       [0.83112526],
       [0.82291985],
       [0.80629542],
       [0.80996461],
       [0.80343882],
       [0.81367987],
       [0.82139521],
       [0.82890951],
       [0.84114432],
       [0.83228549],
       [0.82454921],
       [0.85280948],
       [0.83726151],
       [0.78001215],
       [0.79317263],
       [0.79106159],
       [0.83024566],
       [0.81678779],
       [0.81604641],
       [0.82222874],
       [0.80396239],
       [0.74422501],
       [0.73730968],
       [0.74703554],
       [0.76312802],
       [0.74425433],
       [0.7414815 ],
       [0.77016901],
       [0.76156569],
       [0.75160527],
       [0.78516828],
       [0.77181512],
       [0.80126076],
       [0.8133741 ],
       [0.82851578],
       [0.81258246],
       [0.76597625],
       [0.77124547],
       [0.78014618],
       [0.78000377],
       [0.77810635],
       [0.79252759],
       [0.79457161],
       [0.7628055 ],
       [0.75939182],
       [0.74240717],
       [0.73615364],
       [0.73728874],
       [0.76790299],
       [0.75982324],
       [0.76174161],
       [0.771857  ],
       [0.7788938 ],
       [0.78582588],
       [0.76402857],
       [0.77546336],
       [0.7711617 ],
```

```
[0.7430941 ],
[0.73485518],
[0.73527823],
[0.74265848],
[0.7378165 ],
[0.75046598],
[0.74853924],
[0.74759681],
[0.73983539],
[0.7511906 ],
[0.70831222],
[0.75385034],
[0.82406333],
[0.82851578],
[0.79465539],
[0.71392071],
[0.69552033],
[0.70878552],
[0.78202685],
[0.76182538],
[0.81357515],
[0.78036399],
[0.7792582 ],
[0.76753021],
[0.77110725],
[0.78453999],
[0.78043519],
[0.80787032],
[0.83548975],
[0.8459863 ],
[0.87946135],
[0.87916815],
[0.81370919],
[0.77345704],
[0.77560996],
[0.73551279],
[0.77708015],
[0.74600096],
[0.78632013],
[0.75594044],
[0.75432784],
[0.7665459 ],
[0.73348971],
[0.70193302],
[0.69358521],
[0.7476722 ],
[0.67708643],
[0.67475759],
[0.69730047],
[0.72635657],
[0.75903998],
[0.73677354],
[0.7423234 ],
[0.76698988],
[0.80615301],
[0.75640537],
[0.81704748],
[0.8477832 ],
[0.91624537],
[0.90270372],
[0.88253157],
[0.86194056],
[0.86118662],
[0.87871578],
[0.83877359],
[0.8224256 ],
[0.81290079],
[0.81819933],
[0.8027477 ],
[0.82453245],
[0.77004754],
[0.75564305],
```

```
       [0.76108819],
       [0.7615992 ],
       [0.72743304],
       [0.75570169],
       [0.76378563],
       [0.78788666],
       [0.77952208],
       [0.75179375],
       [0.74050975],
       [0.76262121],
       [0.71537414],
       [0.70768812],
       [0.69378207],
       [0.71676894],
       [0.69737586],
       [0.68801022],
       [0.73549184],
       [0.75156757],
       [0.77609165],
       [0.67771472],
       [0.69360196],
       [0.69735911],
       [0.72882783],
       [0.73720497],
       [0.77741523],
       [0.77013969],
       [0.72761733],
       [0.69369411],
       [0.71753545],
       [0.64769524],
       [0.6428365 ],
       [0.59262393],
       [0.55976879],
       [0.56500031],
       [0.59084798],
       [0.58239544],
       [0.59496534],
       [0.57377537],
       [0.55751953],
       [0.54871097],
       [0.54326583],
       [0.53249702],
       [0.5146579 ],
       [0.50307232],
       [0.50849232],
       [0.54597583],
       [0.52646129],
       [0.49429307],
       [0.47709481]])
```

In [51]:

```
# convert an array of values into a dataset matrix
def create_dataset(dataset, time_step=1):
    dataX, dataY = [], []
    for i in range(len(dataset)-time_step-1):
        a = dataset[i:(i+time_step), 0]    ###i=0, 0,1,2,3-----99   100
        dataX.append(a)
        dataY.append(dataset[i + time_step, 0])
    return numpy.array(dataX), numpy.array(dataY)
```

In [52]:

```
# reshape into X=t,t+1,t+2,t+3 and Y=t+4
time_step = 100
X_train, y_train = create_dataset(train_data, time_step)
X_test, ytest = create_dataset(test_data, time_step)
```

In [53]:

```
print(X_train.shape), print(y_train.shape)
```

```
(716, 100)
(716,)
```

Out[53]:

```
(None, None)
```

In [54]:

```
print(X_test.shape), print(ytest.shape)
```

```
(340, 100)
(340,)
```

Out[54]:

```
(None, None)
```

In [55]:

```
# reshape input to be [samples, time steps, features] which is required for LSTM
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)
```

In [56]:

```
### Create the Stacked LSTM model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import LSTM
```

In [57]:

```
model=Sequential()
model.add(LSTM(50,return_sequences=True,input_shape=(100,1)))
model.add(LSTM(50,return_sequences=True))
model.add(LSTM(50))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')
```

In [58]:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 100, 50)           10400

 lstm_1 (LSTM)               (None, 100, 50)           20200

 lstm_2 (LSTM)               (None, 50)                20200

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 50851 (198.64 KB)
Trainable params: 50851 (198.64 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [59]:

```
model.fit(X_train,y_train,validation_data=(X_test,ytest),epochs=200,batch_size=64,verbos
e=1)
```

```
Epoch 1/200
12/12 [==============================] - 11s 315ms/step - loss: 0.1112 - val_loss: 0.0052
Epoch 2/200
12/12 [==============================] - 3s 223ms/step - loss: 0.0178 - val_loss: 0.0265
Epoch 3/200
```

```
12/12 [==============================] - 4s 295ms/step - loss: 0.0100 - val_loss: 0.0106
Epoch 4/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0075 - val_loss: 0.0054
Epoch 5/200
12/12 [==============================] - 3s 221ms/step - loss: 0.0061 - val_loss: 0.0032
Epoch 6/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0055 - val_loss: 0.0024
Epoch 7/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0054 - val_loss: 0.0019
Epoch 8/200
12/12 [==============================] - 4s 322ms/step - loss: 0.0052 - val_loss: 0.0017
Epoch 9/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0051 - val_loss: 0.0020
Epoch 10/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0050 - val_loss: 0.0023
Epoch 11/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0048 - val_loss: 0.0023
Epoch 12/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0045 - val_loss: 0.0013
Epoch 13/200
12/12 [==============================] - 3s 280ms/step - loss: 0.0046 - val_loss: 0.0015
Epoch 14/200
12/12 [==============================] - 3s 223ms/step - loss: 0.0042 - val_loss: 0.0014
Epoch 15/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0040 - val_loss: 0.0011
Epoch 16/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0041 - val_loss: 0.0020
Epoch 17/200
12/12 [==============================] - 2s 190ms/step - loss: 0.0039 - val_loss: 0.0011
Epoch 18/200
12/12 [==============================] - 3s 238ms/step - loss: 0.0038 - val_loss: 0.0012
Epoch 19/200
12/12 [==============================] - 3s 272ms/step - loss: 0.0035 - val_loss: 0.0011
Epoch 20/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0033 - val_loss: 0.0011
Epoch 21/200
12/12 [==============================] - 2s 190ms/step - loss: 0.0032 - val_loss: 0.0014
Epoch 22/200
12/12 [==============================] - 2s 189ms/step - loss: 0.0032 - val_loss: 9.3796e
-04
Epoch 23/200
12/12 [==============================] - 2s 189ms/step - loss: 0.0031 - val_loss: 0.0013
Epoch 24/200
12/12 [==============================] - 4s 318ms/step - loss: 0.0031 - val_loss: 8.7044e
-04
Epoch 25/200
12/12 [==============================] - 2s 189ms/step - loss: 0.0030 - val_loss: 8.9277e
-04
Epoch 26/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0030 - val_loss: 0.0010
Epoch 27/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0028 - val_loss: 0.0010
Epoch 28/200
12/12 [==============================] - 2s 190ms/step - loss: 0.0028 - val_loss: 8.1183e
-04
Epoch 29/200
12/12 [==============================] - 3s 276ms/step - loss: 0.0028 - val_loss: 8.4437e
-04
Epoch 30/200
12/12 [==============================] - 3s 232ms/step - loss: 0.0029 - val_loss: 8.4345e
-04
Epoch 31/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0028 - val_loss: 8.7914e
-04
Epoch 32/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0026 - val_loss: 9.3342e
-04
Epoch 33/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0026 - val_loss: 7.8148e
-04
Epoch 34/200
12/12 [==============================] - 3s 212ms/step - loss: 0.0026 - val_loss: 8.8172e
```

```
-04
Epoch 35/200
12/12 [==============================] - 3s 288ms/step - loss: 0.0025 - val_loss: 0.0010
Epoch 36/200
12/12 [==============================] - 2s 189ms/step - loss: 0.0025 - val_loss: 7.7745e
-04
Epoch 37/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0024 - val_loss: 7.5501e
-04
Epoch 38/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0024 - val_loss: 0.0013
Epoch 39/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0030 - val_loss: 7.9592e
-04
Epoch 40/200
12/12 [==============================] - 3s 296ms/step - loss: 0.0026 - val_loss: 7.1824e
-04
Epoch 41/200
12/12 [==============================] - 3s 215ms/step - loss: 0.0024 - val_loss: 8.8900e
-04
Epoch 42/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0023 - val_loss: 6.9534e
-04
Epoch 43/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0025 - val_loss: 8.2952e
-04
Epoch 44/200
12/12 [==============================] - 3s 223ms/step - loss: 0.0023 - val_loss: 7.7904e
-04
Epoch 45/200
12/12 [==============================] - 3s 285ms/step - loss: 0.0023 - val_loss: 9.0990e
-04
Epoch 46/200
12/12 [==============================] - 3s 237ms/step - loss: 0.0022 - val_loss: 7.6449e
-04
Epoch 47/200
12/12 [==============================] - 2s 200ms/step - loss: 0.0022 - val_loss: 7.3711e
-04
Epoch 48/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0021 - val_loss: 8.5639e
-04
Epoch 49/200
12/12 [==============================] - 3s 225ms/step - loss: 0.0021 - val_loss: 6.6984e
-04
Epoch 50/200
12/12 [==============================] - 3s 276ms/step - loss: 0.0021 - val_loss: 7.2356e
-04
Epoch 51/200
12/12 [==============================] - 3s 248ms/step - loss: 0.0021 - val_loss: 8.4385e
-04
Epoch 52/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0020 - val_loss: 9.3870e
-04
Epoch 53/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0022 - val_loss: 7.3219e
-04
Epoch 54/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0020 - val_loss: 8.2504e
-04
Epoch 55/200
12/12 [==============================] - 3s 236ms/step - loss: 0.0020 - val_loss: 6.6639e
-04
Epoch 56/200
12/12 [==============================] - 3s 279ms/step - loss: 0.0019 - val_loss: 6.2923e
-04
Epoch 57/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0020 - val_loss: 6.1872e
-04
Epoch 58/200
12/12 [==============================] - 3s 218ms/step - loss: 0.0021 - val_loss: 5.9428e
-04
Epoch 59/200
```

```
12/12 [==============================] - 2s 194ms/step - loss: 0.0021 - val_loss: 7.3725e
-04
Epoch 60/200
12/12 [==============================] - 3s 219ms/step - loss: 0.0019 - val_loss: 6.7221e
-04
Epoch 61/200
12/12 [==============================] - 4s 299ms/step - loss: 0.0018 - val_loss: 8.9676e
-04
Epoch 62/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0022 - val_loss: 9.7795e
-04
Epoch 63/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0020 - val_loss: 5.8988e
-04
Epoch 64/200
12/12 [==============================] - 3s 222ms/step - loss: 0.0017 - val_loss: 5.6792e
-04
Epoch 65/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0018 - val_loss: 5.9357e
-04
Epoch 66/200
12/12 [==============================] - 4s 326ms/step - loss: 0.0018 - val_loss: 6.7447e
-04
Epoch 67/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0017 - val_loss: 5.8439e
-04
Epoch 68/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0018 - val_loss: 5.8979e
-04
Epoch 69/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0018 - val_loss: 5.8000e
-04
Epoch 70/200
12/12 [==============================] - 2s 201ms/step - loss: 0.0016 - val_loss: 5.5233e
-04
Epoch 71/200
12/12 [==============================] - 3s 303ms/step - loss: 0.0021 - val_loss: 7.4803e
-04
Epoch 72/200
12/12 [==============================] - 3s 221ms/step - loss: 0.0017 - val_loss: 6.8487e
-04
Epoch 73/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0017 - val_loss: 6.1766e
-04
Epoch 74/200
12/12 [==============================] - 2s 198ms/step - loss: 0.0016 - val_loss: 6.5619e
-04
Epoch 75/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0015 - val_loss: 6.0406e
-04
Epoch 76/200
12/12 [==============================] - 3s 272ms/step - loss: 0.0015 - val_loss: 5.8139e
-04
Epoch 77/200
12/12 [==============================] - 3s 244ms/step - loss: 0.0015 - val_loss: 4.9878e
-04
Epoch 78/200
12/12 [==============================] - 2s 202ms/step - loss: 0.0017 - val_loss: 5.2887e
-04
Epoch 79/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0016 - val_loss: 5.3506e
-04
Epoch 80/200
12/12 [==============================] - 2s 197ms/step - loss: 0.0014 - val_loss: 4.8324e
-04
Epoch 81/200
12/12 [==============================] - 3s 236ms/step - loss: 0.0016 - val_loss: 4.7924e
-04
Epoch 82/200
12/12 [==============================] - 3s 271ms/step - loss: 0.0016 - val_loss: 5.4998e
-04
Epoch 83/200
```

```
12/12 [==============================] - 2s 192ms/step - loss: 0.0015 - val_loss: 5.5226e
-04
Epoch 84/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0015 - val_loss: 6.2062e
-04
Epoch 85/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0015 - val_loss: 5.0170e
-04
Epoch 86/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0014 - val_loss: 4.5411e
-04
Epoch 87/200
12/12 [==============================] - 4s 323ms/step - loss: 0.0016 - val_loss: 4.5606e
-04
Epoch 88/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0014 - val_loss: 4.5882e
-04
Epoch 89/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0013 - val_loss: 4.9549e
-04
Epoch 90/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0013 - val_loss: 5.0172e
-04
Epoch 91/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0012 - val_loss: 5.9305e
-04
Epoch 92/200
12/12 [==============================] - 3s 294ms/step - loss: 0.0013 - val_loss: 5.9567e
-04
Epoch 93/200
12/12 [==============================] - 3s 218ms/step - loss: 0.0012 - val_loss: 4.6154e
-04
Epoch 94/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0012 - val_loss: 4.9117e
-04
Epoch 95/200
12/12 [==============================] - 2s 190ms/step - loss: 0.0012 - val_loss: 6.1723e
-04
Epoch 96/200
12/12 [==============================] - 2s 200ms/step - loss: 0.0013 - val_loss: 4.9011e
-04
Epoch 97/200
12/12 [==============================] - 3s 261ms/step - loss: 0.0013 - val_loss: 3.9908e
-04
Epoch 98/200
12/12 [==============================] - 4s 367ms/step - loss: 0.0012 - val_loss: 4.7833e
-04
Epoch 99/200
12/12 [==============================] - 3s 220ms/step - loss: 0.0011 - val_loss: 4.0900e
-04
Epoch 100/200
12/12 [==============================] - 2s 199ms/step - loss: 0.0011 - val_loss: 3.9580e
-04
Epoch 101/200
12/12 [==============================] - 2s 198ms/step - loss: 0.0012 - val_loss: 6.4360e
-04
Epoch 102/200
12/12 [==============================] - 2s 198ms/step - loss: 0.0011 - val_loss: 3.9954e
-04
Epoch 103/200
12/12 [==============================] - 3s 303ms/step - loss: 0.0010 - val_loss: 4.2054e
-04
Epoch 104/200
12/12 [==============================] - 3s 214ms/step - loss: 0.0011 - val_loss: 4.3569e
-04
Epoch 105/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0012 - val_loss: 3.7896e
-04
Epoch 106/200
12/12 [==============================] - 2s 190ms/step - loss: 0.0011 - val_loss: 4.2505e
-04
Epoch 107/200
```

```
12/12 [==============================] - 2s 196ms/step - loss: 0.0010 - val_loss: 3.7135e
-04
Epoch 108/200
12/12 [==============================] - 3s 250ms/step - loss: 0.0012 - val_loss: 6.2967e
-04
Epoch 109/200
12/12 [==============================] - 3s 260ms/step - loss: 0.0012 - val_loss: 4.1599e
-04
Epoch 110/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0011 - val_loss: 3.3622e
-04
Epoch 111/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0011 - val_loss: 4.9461e
-04
Epoch 112/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0011 - val_loss: 3.3632e
-04
Epoch 113/200
12/12 [==============================] - 3s 223ms/step - loss: 0.0015 - val_loss: 3.7339e
-04
Epoch 114/200
12/12 [==============================] - 3s 293ms/step - loss: 0.0011 - val_loss: 6.4028e
-04
Epoch 115/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0015 - val_loss: 5.7667e
-04
Epoch 116/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0012 - val_loss: 5.1300e
-04
Epoch 117/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0012 - val_loss: 4.6902e
-04
Epoch 118/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0011 - val_loss: 6.3387e
-04
Epoch 119/200
12/12 [==============================] - 3s 297ms/step - loss: 0.0012 - val_loss: 5.3059e
-04
Epoch 120/200
12/12 [==============================] - 3s 210ms/step - loss: 9.8564e-04 - val_loss: 3.5
449e-04
Epoch 121/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0011 - val_loss: 3.2277e
-04
Epoch 122/200
12/12 [==============================] - 2s 189ms/step - loss: 9.5817e-04 - val_loss: 3.2
036e-04
Epoch 123/200
12/12 [==============================] - 2s 191ms/step - loss: 9.5735e-04 - val_loss: 3.5
384e-04
Epoch 124/200
12/12 [==============================] - 3s 239ms/step - loss: 9.4264e-04 - val_loss: 3.9
300e-04
Epoch 125/200
12/12 [==============================] - 3s 265ms/step - loss: 0.0010 - val_loss: 4.2625e
-04
Epoch 126/200
12/12 [==============================] - 2s 192ms/step - loss: 9.4477e-04 - val_loss: 6.1
372e-04
Epoch 127/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0011 - val_loss: 4.7582e
-04
Epoch 128/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0010 - val_loss: 4.7361e
-04
Epoch 129/200
12/12 [==============================] - 2s 190ms/step - loss: 9.0033e-04 - val_loss: 3.2
385e-04
Epoch 130/200
12/12 [==============================] - 4s 325ms/step - loss: 9.6767e-04 - val_loss: 3.2
094e-04
Epoch 131/200
```

```
12/12 [==============================] - 2s 189ms/step - loss: 8.6573e-04 - val_loss: 3.1
088e-04
Epoch 132/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0011 - val_loss: 2.9152e
-04
Epoch 133/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0011 - val_loss: 3.1854e
-04
Epoch 134/200
12/12 [==============================] - 2s 192ms/step - loss: 9.1068e-04 - val_loss: 4.1
422e-04
Epoch 135/200
12/12 [==============================] - 3s 297ms/step - loss: 9.8577e-04 - val_loss: 3.8
638e-04
Epoch 136/200
12/12 [==============================] - 3s 218ms/step - loss: 9.0229e-04 - val_loss: 2.8
438e-04
Epoch 137/200
12/12 [==============================] - 2s 192ms/step - loss: 8.8780e-04 - val_loss: 3.1
717e-04
Epoch 138/200
12/12 [==============================] - 2s 192ms/step - loss: 8.3073e-04 - val_loss: 3.7
117e-04
Epoch 139/200
12/12 [==============================] - 2s 194ms/step - loss: 8.3620e-04 - val_loss: 3.5
745e-04
Epoch 140/200
12/12 [==============================] - 3s 241ms/step - loss: 8.3775e-04 - val_loss: 4.2
948e-04
Epoch 141/200
12/12 [==============================] - 3s 267ms/step - loss: 9.2655e-04 - val_loss: 3.4
715e-04
Epoch 142/200
12/12 [==============================] - 2s 192ms/step - loss: 8.4657e-04 - val_loss: 2.8
053e-04
Epoch 143/200
12/12 [==============================] - 2s 193ms/step - loss: 8.4058e-04 - val_loss: 4.7
057e-04
Epoch 144/200
12/12 [==============================] - 2s 193ms/step - loss: 9.6828e-04 - val_loss: 5.4
818e-04
Epoch 145/200
12/12 [==============================] - 2s 190ms/step - loss: 8.4893e-04 - val_loss: 2.7
721e-04
Epoch 146/200
12/12 [==============================] - 4s 308ms/step - loss: 8.4368e-04 - val_loss: 2.8
174e-04
Epoch 147/200
12/12 [==============================] - 2s 191ms/step - loss: 8.0640e-04 - val_loss: 4.3
506e-04
Epoch 148/200
12/12 [==============================] - 2s 192ms/step - loss: 0.0011 - val_loss: 3.1167e
-04
Epoch 149/200
12/12 [==============================] - 2s 194ms/step - loss: 8.9446e-04 - val_loss: 2.6
782e-04
Epoch 150/200
12/12 [==============================] - 2s 190ms/step - loss: 8.1620e-04 - val_loss: 3.0
162e-04
Epoch 151/200
12/12 [==============================] - 3s 264ms/step - loss: 7.9965e-04 - val_loss: 2.7
070e-04
Epoch 152/200
12/12 [==============================] - 3s 238ms/step - loss: 8.7238e-04 - val_loss: 2.6
151e-04
Epoch 153/200
12/12 [==============================] - 2s 193ms/step - loss: 8.9855e-04 - val_loss: 4.9
225e-04
Epoch 154/200
12/12 [==============================] - 2s 192ms/step - loss: 8.9539e-04 - val_loss: 2.6
121e-04
Epoch 155/200
```

```
12/12 [==============================] - 2s 191ms/step - loss: 9.1225e-04 - val_loss: 2.6
047e-04
Epoch 156/200
12/12 [==============================] - 2s 208ms/step - loss: 0.0010 - val_loss: 4.3222e
-04
Epoch 157/200
12/12 [==============================] - 4s 323ms/step - loss: 9.0448e-04 - val_loss: 2.6
445e-04
Epoch 158/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0010 - val_loss: 2.6088e
-04
Epoch 159/200
12/12 [==============================] - 2s 191ms/step - loss: 0.0010 - val_loss: 2.6124e
-04
Epoch 160/200
12/12 [==============================] - 2s 192ms/step - loss: 9.6493e-04 - val_loss: 3.2
619e-04
Epoch 161/200
12/12 [==============================] - 2s 192ms/step - loss: 8.5032e-04 - val_loss: 2.7
692e-04
Epoch 162/200
12/12 [==============================] - 4s 317ms/step - loss: 9.3280e-04 - val_loss: 3.8
531e-04
Epoch 163/200
12/12 [==============================] - 2s 196ms/step - loss: 9.3046e-04 - val_loss: 2.9
781e-04
Epoch 164/200
12/12 [==============================] - 2s 191ms/step - loss: 8.0474e-04 - val_loss: 3.5
686e-04
Epoch 165/200
12/12 [==============================] - 2s 191ms/step - loss: 8.2469e-04 - val_loss: 3.9
672e-04
Epoch 166/200
12/12 [==============================] - 2s 193ms/step - loss: 9.1216e-04 - val_loss: 3.9
019e-04
Epoch 167/200
12/12 [==============================] - 3s 274ms/step - loss: 8.1169e-04 - val_loss: 3.3
872e-04
Epoch 168/200
12/12 [==============================] - 3s 238ms/step - loss: 8.9763e-04 - val_loss: 2.8
719e-04
Epoch 169/200
12/12 [==============================] - 2s 194ms/step - loss: 0.0011 - val_loss: 2.5650e
-04
Epoch 170/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0011 - val_loss: 2.5271e
-04
Epoch 171/200
12/12 [==============================] - 2s 195ms/step - loss: 8.6392e-04 - val_loss: 2.5
396e-04
Epoch 172/200
12/12 [==============================] - 3s 221ms/step - loss: 8.5656e-04 - val_loss: 2.9
347e-04
Epoch 173/200
12/12 [==============================] - 3s 287ms/step - loss: 7.8678e-04 - val_loss: 2.4
890e-04
Epoch 174/200
12/12 [==============================] - 2s 191ms/step - loss: 8.9285e-04 - val_loss: 2.8
055e-04
Epoch 175/200
12/12 [==============================] - 2s 195ms/step - loss: 7.9591e-04 - val_loss: 2.8
486e-04
Epoch 176/200
12/12 [==============================] - 2s 191ms/step - loss: 8.0566e-04 - val_loss: 4.0
098e-04
Epoch 177/200
12/12 [==============================] - 2s 195ms/step - loss: 8.4285e-04 - val_loss: 3.9
629e-04
Epoch 178/200
12/12 [==============================] - 4s 317ms/step - loss: 8.5144e-04 - val_loss: 2.6
553e-04
Epoch 179/200
```

```
12/12 [==============================] - 2s 200ms/step - loss: 8.4932e-04 - val_loss: 3.0
005e-04
Epoch 180/200
12/12 [==============================] - 2s 193ms/step - loss: 0.0010 - val_loss: 2.7986e
-04
Epoch 181/200
12/12 [==============================] - 2s 197ms/step - loss: 8.8520e-04 - val_loss: 2.5
165e-04
Epoch 182/200
12/12 [==============================] - 2s 196ms/step - loss: 0.0012 - val_loss: 2.7631e
-04
Epoch 183/200
12/12 [==============================] - 3s 289ms/step - loss: 0.0011 - val_loss: 2.6308e
-04
Epoch 184/200
12/12 [==============================] - 3s 225ms/step - loss: 9.5332e-04 - val_loss: 2.5
126e-04
Epoch 185/200
12/12 [==============================] - 2s 196ms/step - loss: 9.6163e-04 - val_loss: 2.7
652e-04
Epoch 186/200
12/12 [==============================] - 2s 194ms/step - loss: 8.1653e-04 - val_loss: 2.7
248e-04
Epoch 187/200
12/12 [==============================] - 2s 195ms/step - loss: 8.8407e-04 - val_loss: 2.7
494e-04
Epoch 188/200
12/12 [==============================] - 3s 237ms/step - loss: 7.9319e-04 - val_loss: 2.5
475e-04
Epoch 189/200
12/12 [==============================] - 3s 271ms/step - loss: 8.2081e-04 - val_loss: 2.4
427e-04
Epoch 190/200
12/12 [==============================] - 2s 195ms/step - loss: 8.9235e-04 - val_loss: 3.0
857e-04
Epoch 191/200
12/12 [==============================] - 2s 193ms/step - loss: 9.4093e-04 - val_loss: 2.8
700e-04
Epoch 192/200
12/12 [==============================] - 2s 190ms/step - loss: 8.8747e-04 - val_loss: 3.8
238e-04
Epoch 193/200
12/12 [==============================] - 2s 192ms/step - loss: 8.0264e-04 - val_loss: 2.3
867e-04
Epoch 194/200
12/12 [==============================] - 4s 326ms/step - loss: 8.8898e-04 - val_loss: 2.5
479e-04
Epoch 195/200
12/12 [==============================] - 2s 195ms/step - loss: 0.0011 - val_loss: 2.6333e
-04
Epoch 196/200
12/12 [==============================] - 2s 194ms/step - loss: 9.9177e-04 - val_loss: 2.3
762e-04
Epoch 197/200
12/12 [==============================] - 2s 195ms/step - loss: 8.5082e-04 - val_loss: 2.7
717e-04
Epoch 198/200
12/12 [==============================] - 2s 195ms/step - loss: 8.5878e-04 - val_loss: 2.3
452e-04
Epoch 199/200
12/12 [==============================] - 3s 296ms/step - loss: 8.3729e-04 - val_loss: 2.4
918e-04
Epoch 200/200
12/12 [==============================] - 4s 354ms/step - loss: 8.1999e-04 - val_loss: 4.8
578e-04
```

Out[59]:

```
<keras.src.callbacks.History at 0x7a1b18185a50>
```

In [60]:

```
import tensorflow as tf
```

In [61]:

```
tf.__version__
```

Out[61]:

```
'2.15.0'
```

In [62]:

```
### Lets Do the prediction and check performance metrics
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

```
23/23 [==============================] - 2s 39ms/step
11/11 [==============================] - 1s 62ms/step
```

In [63]:

```
##Transformback to original form
train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)
```

In [64]:

```
### Calculate RMSE performance metrics
import math
from sklearn.metrics import mean_squared_error
math.sqrt(mean_squared_error(y_train,train_predict))
```

Out[64]:

```
147.6678441996265
```

In [65]:

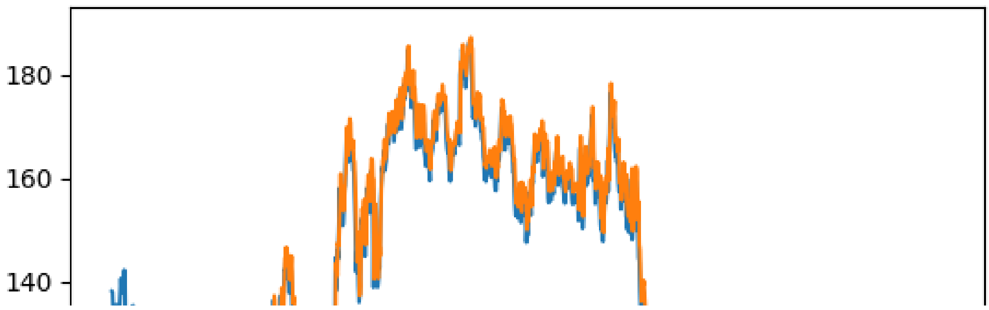
```
### Test Data RMSE
math.sqrt(mean_squared_error(ytest,test_predict))
```

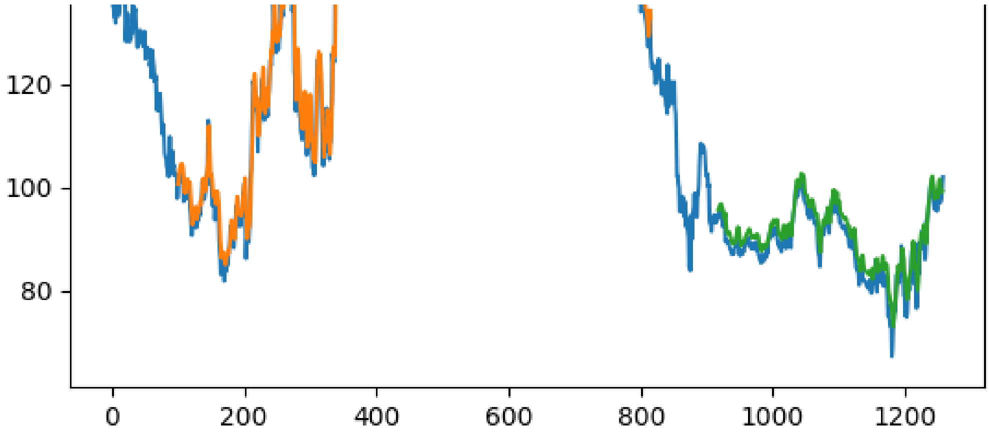
Out[65]:

```
90.78082321584522
```

In [66]:

```
### Plotting
# shift train predictions for plotting
look_back=100
trainPredictPlot = numpy.empty_like(data)
trainPredictPlot[:, :] = np.nan
trainPredictPlot[look_back:len(train_predict)+look_back, :] = train_predict
# shift test predictions for plotting
testPredictPlot = numpy.empty_like(data)
testPredictPlot[:, :] = numpy.nan
testPredictPlot[len(train_predict)+(look_back*2)+1:len(data)-1, :] = test_predict
# plot baseline and predictions
plt.plot(scaler.inverse_transform(data))
plt.plot(trainPredictPlot)
plt.plot(testPredictPlot)
plt.show()
```





In [67]:

```
x_input=test_data[341:].reshape(1,-1)
x_input.shape
```

Out[67]:

```
(1, 100)
```

In [69]:

```
temp_input=list(x_input)
temp_input=temp_input[0].tolist()
```

In [70]:

```
# demonstrate prediction for next 10 days
from numpy import array

lst_output=[]
n_steps=100
i=0
while(i<30):

    if(len(temp_input)>100):
        #print(temp_input)
        x_input=np.array(temp_input[1:])
        print("{} day input {}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1, n_steps, 1))
        #print(x_input)
        yhat = model.predict(x_input, verbose=0)
        print("{} day output {}".format(i,yhat))
        temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]
        #print(temp_input)
        lst_output.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1, n_steps,1))
        yhat = model.predict(x_input, verbose=0)
        print(yhat[0])
        temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        lst_output.extend(yhat.tolist())
        i=i+1


print(lst_output)
```

```
[0.30032226]
101
1 day input [0.13680286 0.13025194 0.12400679 0.12071876 0.14753817 0.14628997
 0.14233597 0.1384741  0.11445266 0.12423297 0.13079227 0.13213261
 0.13094306 0.1196046  0.09693606 0.06547572 0.08174831 0.06618358
 0.05615196 0.04929108 0.0531697  0.         0.01402752 0.04895181
 0.06329766 0.08692119 0.07411674 0.10385558 0.131697   0.1338583
```

```
 0.1253555  0.12442983 0.11944543 0.14879055 0.13589395 0.17943831
 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128
 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946
 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722  0.1251754
 0.11888835 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359
 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375 0.17594923
 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389
 0.17227167 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001
 0.2627699  0.27661312 0.27604348 0.28022367 0.26425265 0.264127
 0.24729314 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463
 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109
 0.25732476 0.27261723 0.29133594 0.30032226]
1 day output [[0.3140608]]
2 day input [0.13025194 0.12400679 0.12071876 0.14753817 0.14628997 0.14233597
 0.1384741  0.11445266 0.12423297 0.13079227 0.13213261 0.13094306
 0.1196046  0.09693606 0.06547572 0.08174831 0.06618358 0.05615196
 0.04929108 0.0531697  0.         0.01402752 0.04895181 0.06329766
 0.08692119 0.07411674 0.10385558 0.131697   0.1338583  0.1253555
 0.12442983 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246
 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591
 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999
 0.12267901 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835
 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754
 0.1835473  0.13413475 0.1779053  0.18653375 0.17594923 0.17875139
 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167
 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699
 0.27661312 0.27604348 0.28022367 0.26425265 0.264127   0.24729314
 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078
 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476
 0.27261723 0.29133594 0.30032226 0.31406081]
2 day output [[0.3265723]]
3 day input [0.12400679 0.12071876 0.14753817 0.14628997 0.14233597 0.1384741
 0.11445266 0.12423297 0.13079227 0.13213261 0.13094306 0.1196046
 0.09693606 0.06547572 0.08174831 0.06618358 0.05615196 0.04929108
 0.0531697  0.         0.01402752 0.04895181 0.06329766 0.08692119
 0.07411674 0.10385558 0.131697   0.1338583  0.1253555  0.12442983
 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944
 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683
 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901
 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183
 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473
 0.13413475 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585
 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167 0.22046954
 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312
 0.27604348 0.28022367 0.26425265 0.264127   0.24729314 0.23918826
 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389
 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723
 0.29133594 0.30032226 0.31406081 0.3265723 ]
3 day output [[0.33851972]]
4 day input [0.12071876 0.14753817 0.14628997 0.14233597 0.1384741  0.11445266
 0.12423297 0.13079227 0.13213261 0.13094306 0.1196046  0.09693606
 0.06547572 0.08174831 0.06618358 0.05615196 0.04929108 0.0531697
 0.         0.01402752 0.04895181 0.06329766 0.08692119 0.07411674
 0.10385558 0.131697   0.1338583  0.1253555  0.12442983 0.11944543
 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026
 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619
 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622
 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183
 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475
 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585 0.1993759
 0.17465916 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828
 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348
 0.28022367 0.26425265 0.264127   0.24729314 0.23918826 0.25145658
 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834
 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594
 0.30032226 0.31406081 0.3265723  0.33851972]
4 day output [[0.3499886]]
5 day input [0.14753817 0.14628997 0.14233597 0.1384741  0.11445266 0.12423297
 0.13079227 0.13213261 0.13094306 0.1196046  0.09693606 0.06547572
 0.08174831 0.06618358 0.05615196 0.04929108 0.0531697  0.
 0.01402752 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558
 0.131697   0.1338583  0.1253555  0.12442983 0.11944543 0.14879055
```

```
 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177
 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386
 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926
 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183 0.10641061
 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053
 0.18653375 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916
 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604
 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367
 0.26425265 0.264127   0.24729314 0.23918826 0.25145658 0.24396741
 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459
 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226
 0.31406081 0.3265723  0.33851972 0.34998861]
5 day output [[0.3611464]]
6 day input [0.14628997 0.14233597 0.1384741  0.11445266 0.12423297 0.13079227
 0.13213261 0.13094306 0.1196046  0.09693606 0.06547572 0.08174831
 0.06618358 0.05615196 0.04929108 0.0531697  0.         0.01402752
 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558 0.131697
 0.1338583  0.1253555  0.12442983 0.11944543 0.14879055 0.13589395
 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407
 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671
 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722
 0.1251754  0.11888835 0.13469183 0.13469183 0.10641061 0.07810006
 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375
 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474
 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604 0.23684684
 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367 0.26425265
 0.264127   0.24729314 0.23918826 0.25145658 0.24396741 0.25009529
 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082
 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081
 0.3265723  0.33851972 0.34998861 0.36114639]
6 day output [[0.37221733]]
7 day input [0.14233597 0.1384741  0.11445266 0.12423297 0.13079227 0.13213261
 0.13094306 0.1196046  0.09693606 0.06547572 0.08174831 0.06618358
 0.05615196 0.04929108 0.0531697  0.         0.01402752 0.04895181
 0.06329766 0.08692119 0.07411674 0.10385558 0.131697   0.1338583
 0.1253555  0.12442983 0.11944543 0.14879055 0.13589395 0.17943831
 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128
 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946
 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722  0.1251754
 0.11888835 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359
 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375 0.17594923
 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389
 0.17227167 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001
 0.2627699  0.27661312 0.27604348 0.28022367 0.26425265 0.264127
 0.24729314 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463
 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109
 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723
 0.33851972 0.34998861 0.36114639 0.37221733]
7 day output [[0.3834219]]
8 day input [0.1384741  0.11445266 0.12423297 0.13079227 0.13213261 0.13094306
 0.1196046  0.09693606 0.06547572 0.08174831 0.06618358 0.05615196
 0.04929108 0.0531697  0.         0.01402752 0.04895181 0.06329766
 0.08692119 0.07411674 0.10385558 0.131697   0.1338583  0.1253555
 0.12442983 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246
 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591
 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999
 0.12267901 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835
 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754
 0.1835473  0.13413475 0.1779053  0.18653375 0.17594923 0.17875139
 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167
 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699
 0.27661312 0.27604348 0.28022367 0.26425265 0.264127   0.24729314
 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078
 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476
 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972
 0.34998861 0.36114639 0.37221733 0.3834219 ]
8 day output [[0.39494038]]
9 day input [0.11445266 0.12423297 0.13079227 0.13213261 0.13094306 0.1196046
 0.09693606 0.06547572 0.08174831 0.06618358 0.05615196 0.04929108
 0.0531697  0.         0.01402752 0.04895181 0.06329766 0.08692119
 0.07411674 0.10385558 0.131697   0.1338583  0.1253555  0.12442983
 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944
```

```
 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683
 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901
 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183
 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473
 0.13413475 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585
 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167 0.22046954
 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312
 0.27604348 0.28022367 0.26425265 0.264127   0.24729314 0.23918826
 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389
 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723
 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861
 0.36114639 0.37221733 0.3834219  0.39494038]
9 day output [[0.4068972]]
10 day input [0.12423297 0.13079227 0.13213261 0.13094306 0.1196046  0.09693606
 0.06547572 0.08174831 0.06618358 0.05615196 0.04929108 0.0531697
 0.         0.01402752 0.04895181 0.06329766 0.08692119 0.07411674
 0.10385558 0.131697   0.1338583  0.1253555  0.12442983 0.11944543
 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026
 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619
 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622
 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183
 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475
 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585 0.1993759
 0.17465916 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828
 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348
 0.28022367 0.26425265 0.264127   0.24729314 0.23918826 0.25145658
 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834
 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594
 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639
 0.37221733 0.3834219  0.39494038 0.40689719]
10 day output [[0.41935974]]
11 day input [0.13079227 0.13213261 0.13094306 0.1196046  0.09693606 0.06547572
 0.08174831 0.06618358 0.05615196 0.04929108 0.0531697  0.
 0.01402752 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558
 0.131697   0.1338583  0.1253555  0.12442983 0.11944543 0.14879055
 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177
 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386
 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926
 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183 0.10641061
 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053
 0.18653375 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916
 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604
 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367
 0.26425265 0.264127   0.24729314 0.23918826 0.25145658 0.24396741
 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459
 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226
 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733
 0.3834219  0.39494038 0.40689719 0.41935974]
11 day output [[0.43234965]]
12 day input [0.13213261 0.13094306 0.1196046  0.09693606 0.06547572 0.08174831
 0.06618358 0.05615196 0.04929108 0.0531697  0.         0.01402752
 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558 0.131697
 0.1338583  0.1253555  0.12442983 0.11944543 0.14879055 0.13589395
 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407
 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671
 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722
 0.1251754  0.11888835 0.13469183 0.13469183 0.10641061 0.07810006
 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375
 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474
 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604 0.23684684
 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367 0.26425265
 0.264127   0.24729314 0.23918826 0.25145658 0.24396741 0.25009529
 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082
 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081
 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733 0.3834219
 0.39494038 0.40689719 0.41935974 0.43234965]
12 day output [[0.4458574]]
13 day input [0.13094306 0.1196046  0.09693606 0.06547572 0.08174831 0.06618358
 0.05615196 0.04929108 0.0531697  0.         0.01402752 0.04895181
 0.06329766 0.08692119 0.07411674 0.10385558 0.131697   0.1338583
 0.1253555  0.12442983 0.11944543 0.14879055 0.13589395 0.17943831
 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128
```

```
 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946
 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722  0.1251754
 0.11888835 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359
 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375 0.17594923
 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389
 0.17227167 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001
 0.2627699  0.27661312 0.27604348 0.28022367 0.26425265 0.264127
 0.24729314 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463
 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109
 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723
 0.33851972 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038
 0.40689719 0.41935974 0.43234965 0.44585741]
13 day output [[0.45985937]]
14 day input [0.1196046  0.09693606 0.06547572 0.08174831 0.06618358 0.05615196
 0.04929108 0.0531697  0.         0.01402752 0.04895181 0.06329766
 0.08692119 0.07411674 0.10385558 0.131697   0.1338583  0.1253555
 0.12442983 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246
 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591
 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999
 0.12267901 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835
 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754
 0.1835473  0.13413475 0.1779053  0.18653375 0.17594923 0.17875139
 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167
 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699
 0.27661312 0.27604348 0.28022367 0.26425265 0.264127   0.24729314
 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078
 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476
 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972
 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719
 0.41935974 0.43234965 0.44585741 0.45985937]
14 day output [[0.47433043]]
15 day input [0.09693606 0.06547572 0.08174831 0.06618358 0.05615196 0.04929108
 0.0531697  0.         0.01402752 0.04895181 0.06329766 0.08692119
 0.07411674 0.10385558 0.131697   0.1338583  0.1253555  0.12442983
 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944
 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683
 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901
 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183
 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473
 0.13413475 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585
 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167 0.22046954
 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312
 0.27604348 0.28022367 0.26425265 0.264127   0.24729314 0.23918826
 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389
 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723
 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861
 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974
 0.43234965 0.44585741 0.45985937 0.47433043]
15 day output [[0.48925337]]
16 day input [0.06547572 0.08174831 0.06618358 0.05615196 0.04929108 0.0531697
 0.         0.01402752 0.04895181 0.06329766 0.08692119 0.07411674
 0.10385558 0.131697   0.1338583  0.1253555  0.12442983 0.11944543
 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026
 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619
 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622
 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183
 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475
 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585 0.1993759
 0.17465916 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828
 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348
 0.28022367 0.26425265 0.264127   0.24729314 0.23918826 0.25145658
 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834
 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594
 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639
 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965
 0.44585741 0.45985937 0.47433043 0.48925337]
16 day output [[0.5046225]]
17 day input [0.08174831 0.06618358 0.05615196 0.04929108 0.0531697  0.
 0.01402752 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558
 0.131697   0.1338583  0.1253555  0.12442983 0.11944543 0.14879055
 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177
 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386
```

```
 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926
 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183 0.10641061
 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053
 0.18653375 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916
 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604
 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367
 0.26425265 0.264127   0.24729314 0.23918826 0.25145658 0.24396741
 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459
 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226
 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733
 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965 0.44585741
 0.45985937 0.47433043 0.48925337 0.50462252]
17 day output [[0.5204436]]
18 day input [0.06618358 0.05615196 0.04929108 0.0531697  0.         0.01402752
 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558 0.131697
 0.1338583  0.1253555  0.12442983 0.11944543 0.14879055 0.13589395
 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407
 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671
 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722
 0.1251754  0.11888835 0.13469183 0.13469183 0.10641061 0.07810006
 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375
 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474
 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604 0.23684684
 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367 0.26425265
 0.264127   0.24729314 0.23918826 0.25145658 0.24396741 0.25009529
 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082
 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081
 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733 0.3834219
 0.39494038 0.40689719 0.41935974 0.43234965 0.44585741 0.45985937
 0.47433043 0.48925337 0.50462252 0.52044362]
18 day output [[0.53672975]]
19 day input [0.05615196 0.04929108 0.0531697  0.         0.01402752 0.04895181
 0.06329766 0.08692119 0.07411674 0.10385558 0.131697   0.1338583
 0.1253555  0.12442983 0.11944543 0.14879055 0.13589395 0.17943831
 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128
 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946
 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722  0.1251754
 0.11888835 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359
 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375 0.17594923
 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389
 0.17227167 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001
 0.2627699  0.27661312 0.27604348 0.28022367 0.26425265 0.264127
 0.24729314 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463
 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109
 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723
 0.33851972 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038
 0.40689719 0.41935974 0.43234965 0.44585741 0.45985937 0.47433043
 0.48925337 0.50462252 0.52044362 0.53672975]
19 day output [[0.5534962]]
20 day input [0.04929108 0.0531697  0.         0.01402752 0.04895181 0.06329766
 0.08692119 0.07411674 0.10385558 0.131697   0.1338583  0.1253555
 0.12442983 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246
 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591
 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999
 0.12267901 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835
 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754
 0.1835473  0.13413475 0.1779053  0.18653375 0.17594923 0.17875139
 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167
 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699
 0.27661312 0.27604348 0.28022367 0.26425265 0.264127   0.24729314
 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078
 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476
 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972
 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719
 0.41935974 0.43234965 0.44585741 0.45985937 0.47433043 0.48925337
 0.50462252 0.52044362 0.53672975 0.55349618]
20 day output [[0.57075435]]
21 day input [0.0531697  0.         0.01402752 0.04895181 0.06329766 0.08692119
 0.07411674 0.10385558 0.131697   0.1338583  0.1253555  0.12442983
 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944
 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683
 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901
```

```
 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183
 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473
 0.13413475 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585
 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167 0.22046954
 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312
 0.27604348 0.28022367 0.26425265 0.264127   0.24729314 0.23918826
 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389
 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723
 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861
 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974
 0.43234965 0.44585741 0.45985937 0.47433043 0.48925337 0.50462252
 0.52044362 0.53672975 0.55349618 0.57075435]
21 day output [[0.58850735]]
22 day input [0.         0.01402752 0.04895181 0.06329766 0.08692119 0.07411674
 0.10385558 0.131697   0.1338583  0.1253555  0.12442983 0.11944543
 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026
 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619
 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622
 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183
 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475
 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585 0.1993759
 0.17465916 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828
 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348
 0.28022367 0.26425265 0.264127   0.24729314 0.23918826 0.25145658
 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834
 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594
 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639
 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965
 0.44585741 0.45985937 0.47433043 0.48925337 0.50462252 0.52044362
 0.53672975 0.55349618 0.57075435 0.58850735]
22 day output [[0.6067462]]
23 day input [0.01402752 0.04895181 0.06329766 0.08692119 0.07411674 0.10385558
 0.131697   0.1338583  0.1253555  0.12442983 0.11944543 0.14879055
 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177
 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386
 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926
 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183 0.10641061
 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053
 0.18653375 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916
 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604
 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367
 0.26425265 0.264127   0.24729314 0.23918826 0.25145658 0.24396741
 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459
 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226
 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733
 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965 0.44585741
 0.45985937 0.47433043 0.48925337 0.50462252 0.52044362 0.53672975
 0.55349618 0.57075435 0.58850735 0.6067462 ]
23 day output [[0.625447]]
24 day input [0.04895181 0.06329766 0.08692119 0.07411674 0.10385558 0.131697
 0.1338583  0.1253555  0.12442983 0.11944543 0.14879055 0.13589395
 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407
 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671
 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722
 0.1251754  0.11888835 0.13469183 0.13469183 0.10641061 0.07810006
 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375
 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474
 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604 0.23684684
 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367 0.26425265
 0.264127   0.24729314 0.23918826 0.25145658 0.24396741 0.25009529
 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082
 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081
 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733 0.3834219
 0.39494038 0.40689719 0.41935974 0.43234965 0.44585741 0.45985937
 0.47433043 0.48925337 0.50462252 0.52044362 0.53672975 0.55349618
 0.57075435 0.58850735 0.6067462  0.62544698]
24 day output [[0.6445707]]
25 day input [0.06329766 0.08692119 0.07411674 0.10385558 0.131697   0.1338583
 0.1253555  0.12442983 0.11944543 0.14879055 0.13589395 0.17943831
 0.14501246 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128
 0.07236591 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946
 0.1202999  0.12267901 0.15433622 0.17212926 0.1723722  0.1251754
```

```
 0.11888835 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359
 0.1251754  0.1835473  0.13413475 0.1779053  0.18653375 0.17594923
 0.17875139 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389
 0.17227167 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001
 0.2627699  0.27661312 0.27604348 0.28022367 0.26425265 0.264127
 0.24729314 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463
 0.26230078 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109
 0.25732476 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723
 0.33851972 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038
 0.40689719 0.41935974 0.43234965 0.44585741 0.45985937 0.47433043
 0.48925337 0.50462252 0.52044362 0.53672975 0.55349618 0.57075435
 0.58850735 0.6067462  0.62544698 0.64457071]
25 day output [[0.6640633]]
26 day input [0.08692119 0.07411674 0.10385558 0.131697   0.1338583  0.1253555
 0.12442983 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246
 0.13805944 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591
 0.06345683 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999
 0.12267901 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835
 0.13469183 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754
 0.1835473  0.13413475 0.1779053  0.18653375 0.17594923 0.17875139
 0.20430585 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167
 0.22046954 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699
 0.27661312 0.27604348 0.28022367 0.26425265 0.264127   0.24729314
 0.23918826 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078
 0.27054389 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476
 0.27261723 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972
 0.34998861 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719
 0.41935974 0.43234965 0.44585741 0.45985937 0.47433043 0.48925337
 0.50462252 0.52044362 0.53672975 0.55349618 0.57075435 0.58850735
 0.6067462  0.62544698 0.64457071 0.66406327]
26 day output [[0.6838557]]
27 day input [0.07411674 0.10385558 0.131697   0.1338583  0.1253555  0.12442983
 0.11944543 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944
 0.13981026 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683
 0.07050619 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901
 0.15433622 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183
 0.13469183 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473
 0.13413475 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585
 0.1993759  0.17465916 0.18624474 0.1572389  0.17227167 0.22046954
 0.21799828 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312
 0.27604348 0.28022367 0.26425265 0.264127   0.24729314 0.23918826
 0.25145658 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389
 0.27059834 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723
 0.29133594 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861
 0.36114639 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974
 0.43234965 0.44585741 0.45985937 0.47433043 0.48925337 0.50462252
 0.52044362 0.53672975 0.55349618 0.57075435 0.58850735 0.6067462
 0.62544698 0.64457071 0.66406327 0.68385571]
27 day output [[0.7038667]]
28 day input [0.10385558 0.131697   0.1338583  0.1253555  0.12442983 0.11944543
 0.14879055 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026
 0.09946177 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619
 0.10448386 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622
 0.17212926 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183
 0.10641061 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475
 0.1779053  0.18653375 0.17594923 0.17875139 0.20430585 0.1993759
 0.17465916 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828
 0.22856604 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348
 0.28022367 0.26425265 0.264127   0.24729314 0.23918826 0.25145658
 0.24396741 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834
 0.26940459 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594
 0.30032226 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639
 0.37221733 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965
 0.44585741 0.45985937 0.47433043 0.48925337 0.50462252 0.52044362
 0.53672975 0.55349618 0.57075435 0.58850735 0.6067462  0.62544698
 0.64457071 0.66406327 0.68385571 0.70386672]
28 day output [[0.724003]]
29 day input [0.131697   0.1338583  0.1253555  0.12442983 0.11944543 0.14879055
 0.13589395 0.17943831 0.14501246 0.13805944 0.13981026 0.09946177
 0.09942407 0.06622128 0.07236591 0.06345683 0.07050619 0.10448386
 0.11538671 0.10682946 0.1202999  0.12267901 0.15433622 0.17212926
 0.1723722  0.1251754  0.11888835 0.13469183 0.13469183 0.10641061
```

```
 0.07810006 0.08164359 0.1251754  0.1835473  0.13413475 0.1779053
 0.18653375 0.17594923 0.17875139 0.20430585 0.1993759  0.17465916
 0.18624474 0.1572389  0.17227167 0.22046954 0.21799828 0.22856604
 0.23684684 0.2550001  0.2627699  0.27661312 0.27604348 0.28022367
 0.26425265 0.264127   0.24729314 0.23918826 0.25145658 0.24396741
 0.25009529 0.23626463 0.26230078 0.27054389 0.27059834 0.26940459
 0.24924082 0.25471109 0.25732476 0.27261723 0.29133594 0.30032226
 0.31406081 0.3265723  0.33851972 0.34998861 0.36114639 0.37221733
 0.3834219  0.39494038 0.40689719 0.41935974 0.43234965 0.44585741
 0.45985937 0.47433043 0.48925337 0.50462252 0.52044362 0.53672975
 0.55349618 0.57075435 0.58850735 0.6067462  0.62544698 0.64457071
 0.66406327 0.68385571 0.70386672 0.72400302]
29 day output [[0.74416214]]
[[0.3003222644329071], [0.3140608072280884], [0.3265722990036011], [0.33851972222328186],
[0.3499886095523834], [0.3611463904380798], [0.3722173273563385], [0.3834218978881836],
[0.3949403762817383], [0.4068971872329712], [0.4193597435951233], [0.43234965205192566],
[0.4458574056625366], [0.4598593711853027], [0.47433042526245117], [0.48925337195396423]
, [0.5046225190162659], [0.5204436182975769], [0.5367297530174255], [0.5534961819648743],
[0.57075434923172], [0.588507354259491], [0.6067461967468262], [0.6254469752311707], [0.6
445707082748413], [0.6640632748603821], [0.6838557124137878], [0.703866720199585], [0.724
0030169487], [0.7441621422767639]]
```

In [71]:

```
day_new=np.arange(1,101)
day_pred=np.arange(101,131)
```

In [72]:

```
import matplotlib.pyplot as plt
```

In [73]:

```
len(data)
```

Out[73]:

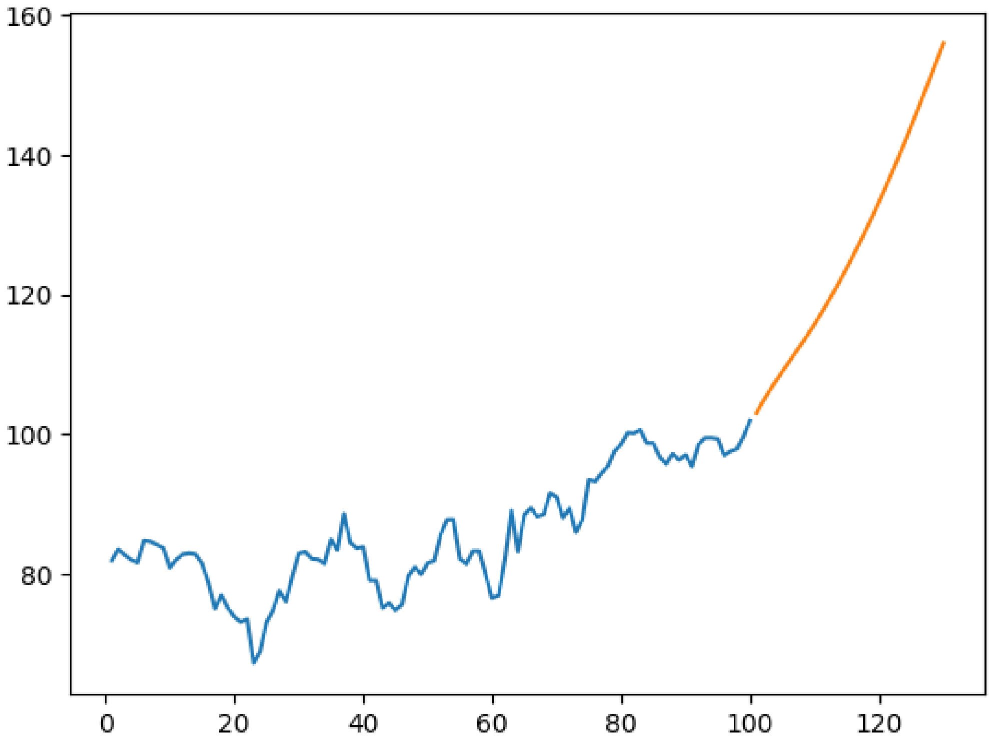
```
1258
```

In [74]:

```
plt.plot(day_new,scaler.inverse_transform(data[1158:]))
plt.plot(day_pred,scaler.inverse_transform(lst_output))
```

Out[74]:

```
[<matplotlib.lines.Line2D at 0x7a1b12962e00>]
```

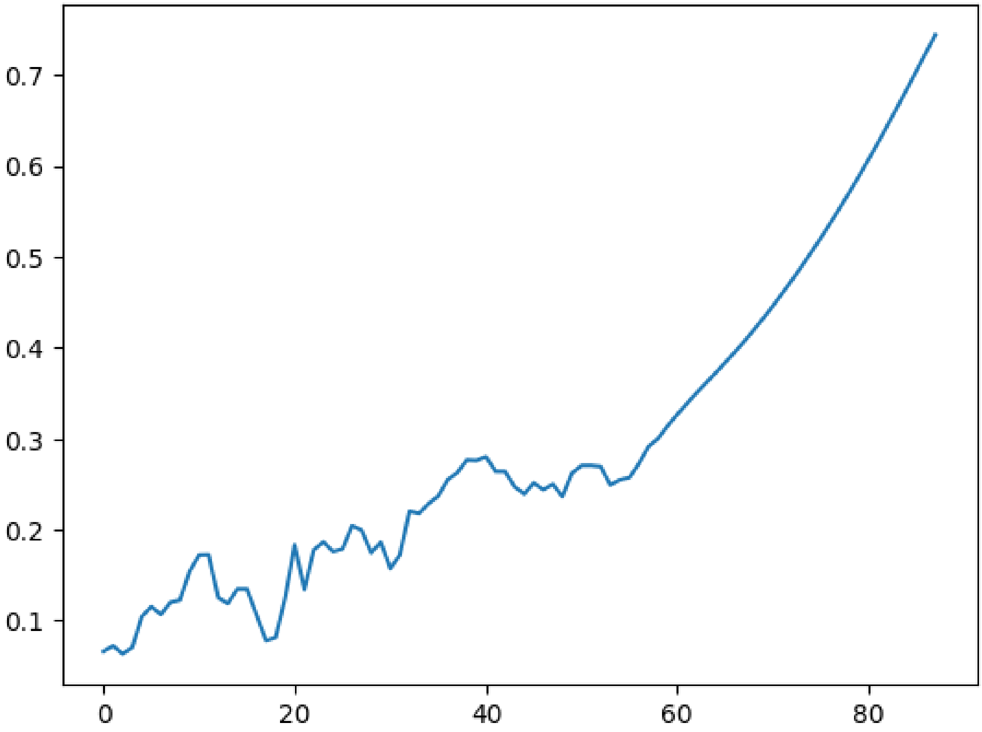


In [75]:

```
data2=data.tolist()
data2.extend(lst_output)
plt.plot(data2[1200:])
```

Out[75]:

```
[<matplotlib.lines.Line2D at 0x7a1b12af4250>]
```



In [76]:

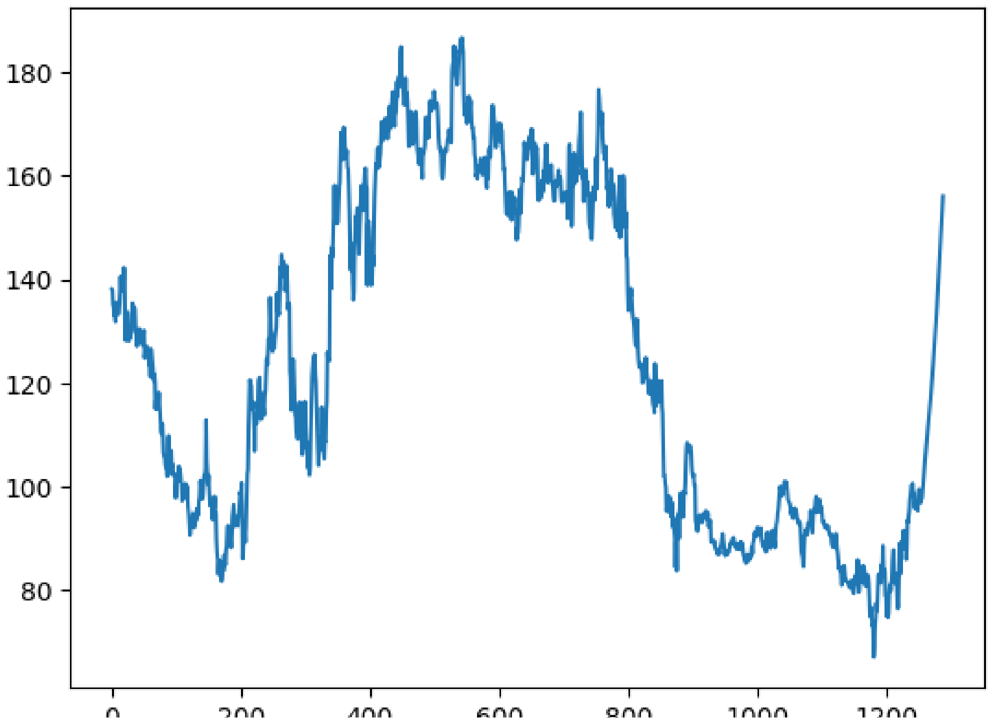
```
data2=scaler.inverse_transform(data2).tolist()
```

In [77]:

```
plt.plot(data2)
```

Out[77]:

```
[<matplotlib.lines.Line2D at 0x7a1b11690be0>]
```



0 200 400 600 800 1000 1200