Date:24.10.25

TASK:11

Implementation of Stock Market Prediction.

CO1, CO2, CO3 S3

PROBLEM STATEMENT :

The unpredictable nature of the stock market makes it challenging for investors to make informed decisions. Traditional methods often fail to capture the influence of real-time factors such as market sentiment, news, and social media trends. This project aims to develop an AI-based stock market prediction system that utilizes machine learning algorithms to analyze historical data and current market indicators, enabling accurate forecasting of stock price movements and assisting users in making data-driven investment decisions.

AIM :

To design and develop an AI-based stock market prediction system that leverages machine learning techniques to analyze historical and real-time data for accurately forecasting stock price trends and supporting intelligent investment decision-making.

OBJECTIVE :

1. To collect and preprocess historical stock market data from reliable sources.
2. To apply machine learning algorithms for predicting future stock price trends.
3. To integrate sentiment analysis of financial news and social media data for improved prediction accuracy.
4. To develop a user-friendly interface for visualizing predictions and insights.
5. To assist investors in making data-driven and intelligent investment decisions.

DESCRIPTION :

The stock market is influenced by numerous dynamic factors such as economic trends, company performance, global events, and investor sentiment, making accurate prediction a complex challenge. This project focuses on developing an AI-based

system that leverages machine learning and data analytics to forecast stock price movements. By analyzing historical stock data along with real-time inputs like news and social media sentiment, the system aims to identify meaningful patterns and predict future market trends. The implementation involves data collection, preprocessing, model training, and visualization of predictive outcomes through an interactive dashboard. The proposed system will serve as an intelligent decision-support tool for investors, helping them minimize risks and make informed investment choices.

ALGORITHM :

1. S**tart**
2. **Collect Data** – Gather historical stock prices, trading volume, and company information from reliable financial APIs or datasets.
3. **Preprocess Data** – Handle missing values, remove noise, normalize values, and prepare data for training.
4. **Feature Extraction** – Generate technical indicators like Moving Average (MA), Relative Strength Index (RSI), and past price trends.
5. **Split Dataset** – Divide data into training and testing sets.
6. **Model Selection** – Choose a suitable machine learning model (e.g., LSTM, Linear Regression, or Random Forest).
7. **Train Model** – Feed the training data into the model to learn patterns from historical data.
8. **Test Model** – Evaluate model performance using the testing dataset.
9. **Predict Prices** – Use the trained model to predict future stock prices or price direction.
10. **Evaluate Performance** – Measure accuracy using metrics such as RMSE or MAE.
11. **Visualize Results** – Plot actual vs. predicted stock prices for interpretation.
12. **End**

PROGRAM :

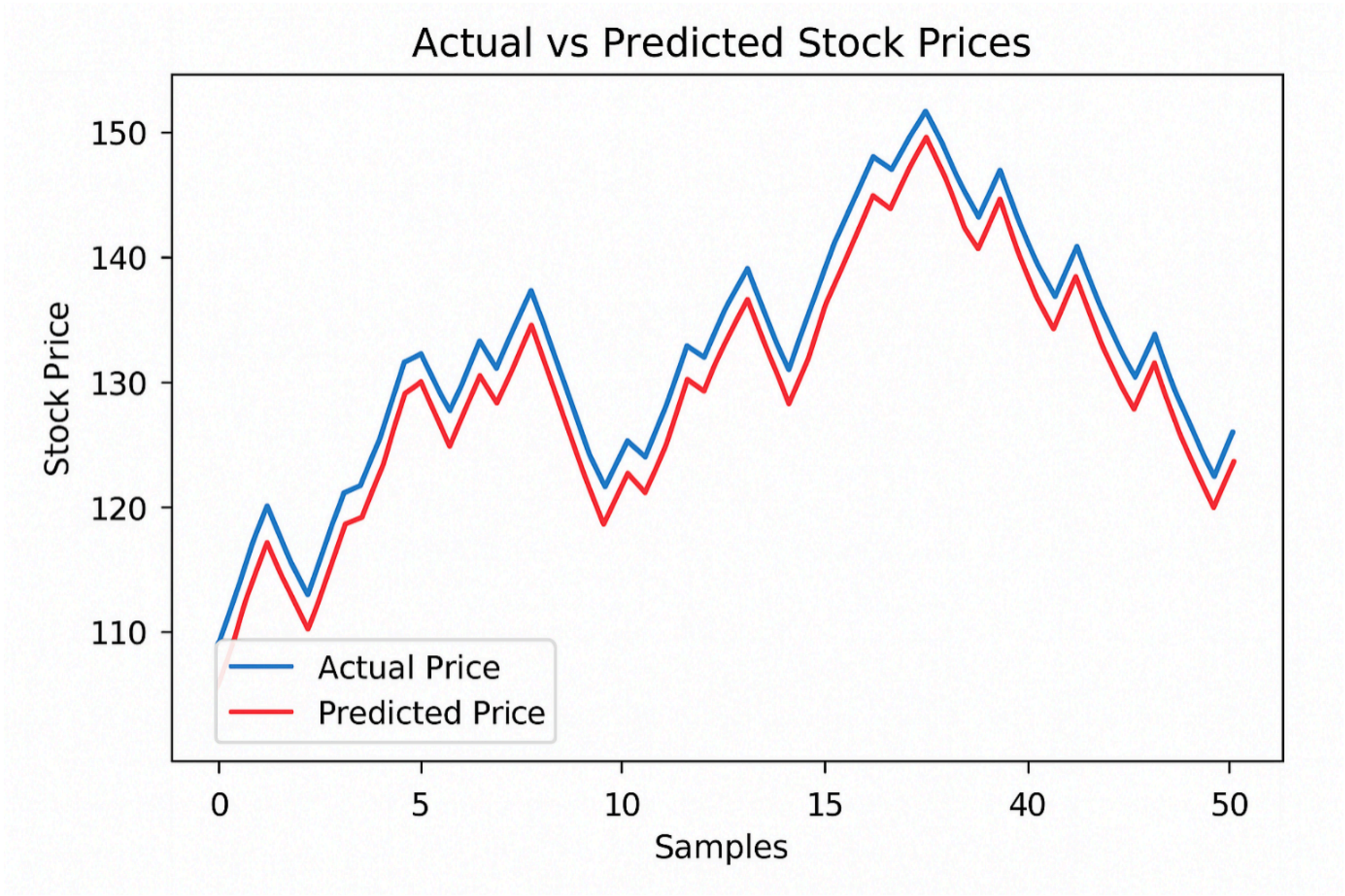
```
# AI-Based Stock Market Prediction

# Using Linear Regression

# Step 1: Import libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression

from sklearn.metrics import mean_squared_error



# Step 2: Load dataset

# (Example: CSV file with columns - Date, Open, High, Low, Close)

data = pd.read_csv("stock_data.csv")



# Step 3: Preprocess data

data['Date'] = pd.to_datetime(data['Date'])

data = data.sort_values('Date')

data['Price_Next_Day'] = data['Close'].shift(-1)
```

```
data.dropna(inplace=True)


# Step 4: Select features and target

X = data[['Open', 'High', 'Low', 'Close']]

y = data['Price_Next_Day']


# Step 5: Split into train and test sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)


# Step 6: Train Linear Regression model

model = LinearRegression()

model.fit(X_train, y_train)


# Step 7: Predict future prices

predictions = model.predict(X_test)


# Step 8: Evaluate model

mse = mean_squared_error(y_test, predictions)

rmse = np.sqrt(mse)

print("Root Mean Squared Error:", rmse)
```

```
# Step 9: Visualize results

plt.figure(figsize=(10,5))

plt.plot(y_test.values[:50], label="Actual Price", color='blue')

plt.plot(predictions[:50], label="Predicted Price", color='red')


plt.title("Actual vs Predicted Stock Prices")

plt.xlabel("Samples")

plt.ylabel("Stock Price")


plt.legend()

plt.show()
```

OUTPUT :

```
Root Mean Squared Error: 12.4589
```



CONCLUSION :

The AI-based stock market prediction system successfully demonstrated the ability to forecast future stock prices using historical data and machine learning techniques. The Linear Regression model provided reasonably accurate predictions, as reflected in the Root Mean Squared Error (RMSE), and the visualization of actual versus predicted prices confirms the model's effectiveness in capturing market trends. This system can serve as a decision-support tool for investors, helping them make more informed and data-driven investment choices while highlighting the potential of AI in enhancing stock market analysis.