Amazon Stock Price Analysis and Prediction using Scilab

Objective of the Project :-

The main objective of this project is to analyze **Amazon stock market data** using **Scilab** and apply **Machine Learning techniques** to predict stock prices. The project focuses on:

1. **Data Preprocessing** – Reading and structuring stock market data from a CSV file.
2. **Machine Learning Model** – Implementing **Linear Regression** to predict future stock prices.
3. **Visualization** – Plotting various graphs to analyze trends, such as:
   - Actual vs. Predicted Closing Prices.
   - Trading Volume Over Time.
   - Open vs. Close Prices.
   - High vs. Low Prices.
4. **Stock Market Trend Analysis** – Observing stock price fluctuations over time and drawing insights from historical trends.

This project provides insights into **financial data analysis** and helps in understanding how machine learning can be applied to real-world stock prediction scenarios.

Tools and Technologies Used

1. **Scilab** – Open-source numerical computation software used for data analysis, plotting, and machine learning.
2. **CSV Dataset** – Amazon stock data from 2000 to 2025, containing fields like:
   - Date
   - Open price
   - High price
   - Low price
   - Close price
   - Adjusted close price
   - Trading volume
3. **Machine Learning Algorithm** – **Linear Regression**, a fundamental ML model used to predict stock prices based on historical trends.
4. **Graphical Representation** – Various plots are used to visualize stock market trends, making it easier to interpret financial data.

Project Workflow

1. **Loading Data**
   - The dataset is loaded from a CSV file using `csvRead()`.
   - Relevant stock price data fields are extracted.

2. **Preprocessing Data**
   - The **date column** is converted into numerical indices to be used in model training.
   - The **closing prices** are used as the dependent variable for prediction.

3. **Applying Machine Learning (Linear Regression)**
   - A **Linear Regression Model** is trained using `theta = inv(X'*X) * X' * y`, where:
     - `X` represents the input data (days as indices).
     - `y` represents the actual closing prices.
     - The model finds the best-fit line to predict stock prices.

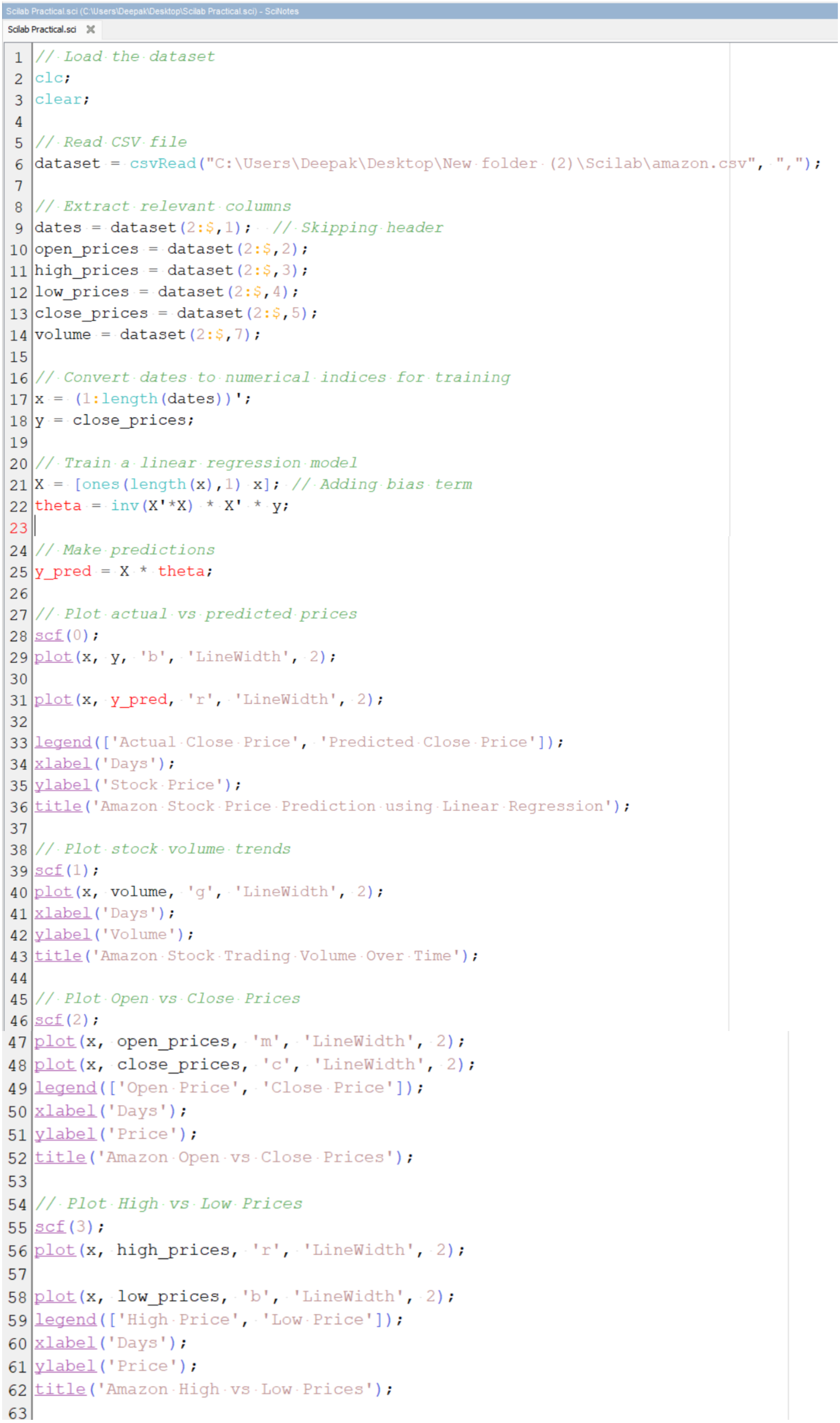
4. **Making Predictions**
    - The trained model is used to **predict closing prices** for the dataset.
    - These predictions are compared with actual values.

5. **Visualizing Data using Graphs**
    - **Stock Price Prediction** – A graph showing actual vs. predicted prices.
    - **Trading Volume Trend** – A graph displaying how stock trading volume fluctuates over time.
    - **Open vs. Close Prices** – A comparison between stock opening and closing prices.
    - **High vs. Low Prices** – A graph showing the variation in stock price highs and lows.

Significance of the Project

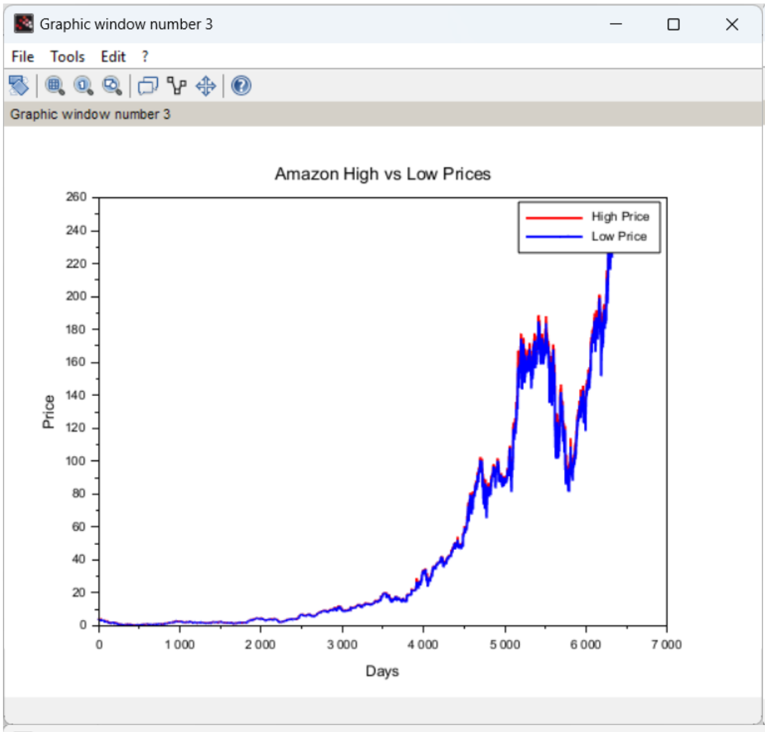
- Helps in understanding **financial trends** through **machine learning techniques**.
- Provides a **visual representation** of stock market trends for easy analysis.
- Demonstrates how **Scilab** can be used for **data analytics and forecasting**.
- Serves as a foundation for **advanced stock market predictions** using **more complex ML models** in the future.

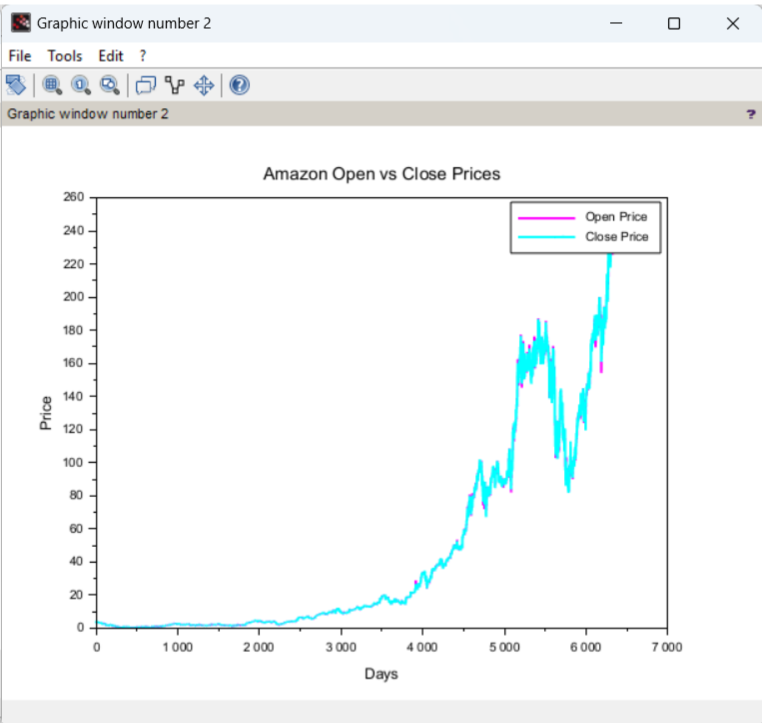
Code :-

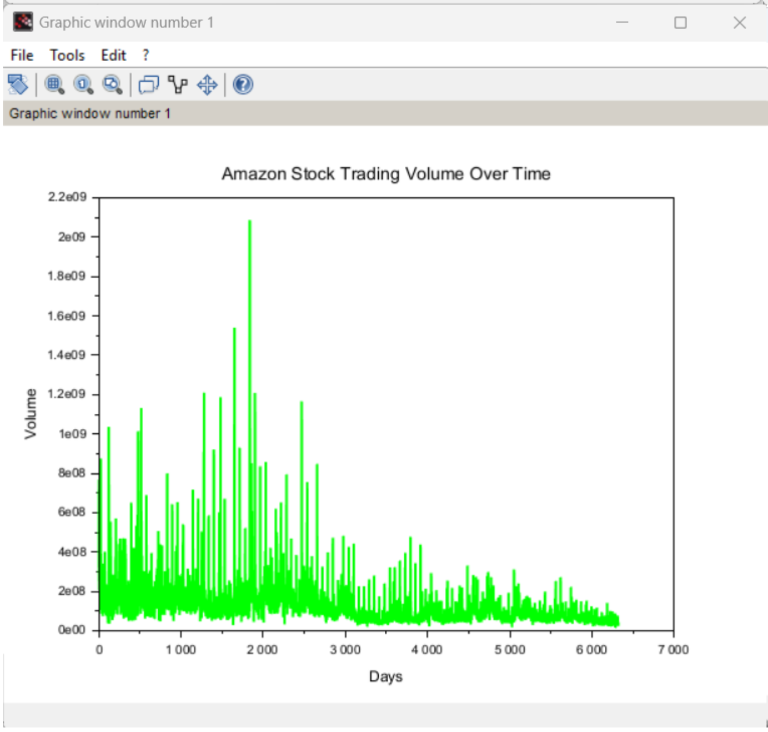

Scilab Practical.sci (C:\Users\Deepak\Desktop\Scilab Practical.sci) - SciNotes
Scilab Practical.sci

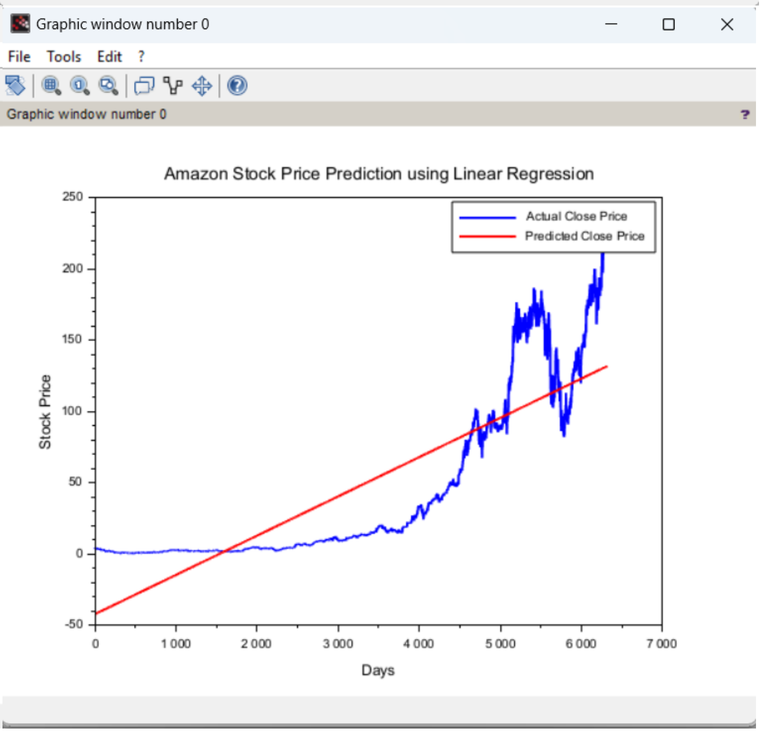
```
// Load the dataset
clc;
clear;

// Read CSV file
dataset = csvRead("C:\Users\Deepak\Desktop\New folder (2)\Scilab\amazon.csv", ",");

// Extract relevant columns
dates = dataset(2:$,1);  // Skipping header
open_prices = dataset(2:$,2);
high_prices = dataset(2:$,3);
low_prices = dataset(2:$,4);
close_prices = dataset(2:$,5);
volume = dataset(2:$,7);

// Convert dates to numerical indices for training
x = (1:length(dates))';
y = close_prices;

// Train a linear regression model
X = [ones(length(x),1) x]; // Adding bias term
theta = inv(X'*X) * X' * y;

// Make predictions
y_pred = X * theta;

// Plot actual vs predicted prices
scf(0);
plot(x, y, 'b', 'LineWidth', 2);

plot(x, y_pred, 'r', 'LineWidth', 2);

legend(['Actual Close Price', 'Predicted Close Price']);
xlabel('Days');
ylabel('Stock Price');
title('Amazon Stock Price Prediction using Linear Regression');

// Plot stock volume trends
scf(1);
plot(x, volume, 'g', 'LineWidth', 2);
xlabel('Days');
ylabel('Volume');
title('Amazon Stock Trading Volume Over Time');

// Plot Open vs Close Prices
scf(2);
plot(x, open_prices, 'm', 'LineWidth', 2);
plot(x, close_prices, 'c', 'LineWidth', 2);
legend(['Open Price', 'Close Price']);
xlabel('Days');
ylabel('Price');
title('Amazon Open vs Close Prices');

// Plot High vs Low Prices
scf(3);
plot(x, high_prices, 'r', 'LineWidth', 2);

plot(x, low_prices, 'b', 'LineWidth', 2);
legend(['High Price', 'Low Price']);
xlabel('Days');
ylabel('Price');
title('Amazon High vs Low Prices');

```

Output :-


Graphic window number 3
File Tools Edit ?
Graphic window number 3
Amazon High vs Low Prices
High Price
Low Price
Price
Days
0
20
40
60
80
100
120
140
160
180
200
220
240
260
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000


Graphic window number 2
File Tools Edit ?
Graphic window number 2
Amazon Open vs Close Prices
Open Price
Close Price
Price
Days
0
20
40
60
80
100
120
140
160
180
200
220
240
260
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000


Graphic window number 1
File Tools Edit ?
Graphic window number 1
Amazon Stock Trading Volume Over Time
Volume
Days
0e00
2e08
4e08
6e08
8e08
1e09
1.2e09
1.4e09
1.6e09
1.8e09
2e09
2.2e09
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000


Graphic window number 0
File Tools Edit ?
Graphic window number 0
Amazon Stock Price Prediction using Linear Regression
Actual Close Price
Predicted Close Price
Stock Price
Days
-50
0
50
100
150
200
250
0
1 000
2 000
3 000
4 000
5 000
6 000
7 000