

Stock Price Prediction Project

PROJECT OVERVIEW:

Stock price prediction involves forecasting the future value of a company's stock based on past price movements, market trends and other relevant financial and economic data. It is crucial for investors to make informed decisions on buying, selling, or holding stocks.

The purpose of our project is to develop a model using Python libraries to predict stock prices based on historical data. Our project aims to assist traders, investors, and analysts in making data-driven investment decisions.

PROBLEMS OUR PROJECT IS GOING TO SOLVE->

Stock markets are highly volatile, influenced by various factors like economic trends, political events, and company performance.

Predicting stock prices helps:

- Investors make informed buying/selling decisions.
- Traders identify profitable opportunities.
- Researchers explore financial market trends.

KEY FEATURES->

- Analyze stock price trends using historical data.
- Build predictive models using ML (Random Forest) and DL (LSTM).
- Optimize models for better accuracy using feature engineering.
- Deploy an interactive dashboard using Streamlit to visualize predictions (optional).

DEPENDENCIES:

Python library is a collection of pre-written code that you can use in your programs to perform specific tasks without having to write the code from scratch.

We used several python libraries to build our project such as ***Numpy, Matplotlib, Pandas and yfinance*** for data collection, preprocessing, model building and visualization.

NUMPY: NumPy is the fundamental library for numerical computing in Python. It's much more efficient for numerical operations than standard Python lists. NumPy includes

functions for linear algebra operations. A wide range of mathematical functions are available for arrays (trigonometric, exponential, logarithmic, etc.)

MATPLOTLIB: Matplotlib is a popular Python library used for data visualization. It helps create line charts, bar graphs, histograms, scatter plots, and more. Supports customization (colors, labels, grids, etc.) Can handle large datasets Works well with NumPy & Pandas.

PANDAS: Pandas is built on top of NumPy and provides data structures and tools for working with structured data (like tables or spreadsheets).

Series: A one-dimensional labelled array capable of holding data of any type.

Data Frame: A two-dimensional labelled data structure with columns of potentially different types.

LSTM: LSTM (Long Short-Term Memory) plays a crucial role by effectively handling sequential and time-series data. Unlike traditional machine learning models, LSTM is designed to capture long-term dependencies and patterns in stock price movements, It prevents issues like vanishing gradients, which often occur in standard neural networks when dealing with long sequences.

In our project, LSTM is used to learn from past stock prices and technical indicators, enabling the model to recognize trends and fluctuations over time. By using multiple LSTM layers along with dropout layers to prevent overfitting, the model can generate more accurate stock price forecasts. With its ability to analyze past dependencies, LSTM provides a data-driven approach to making informed investment decisions.

SETUP INSTRUCTIONS:

1. Data Processing and Analysis with NumPy:
 - Once the data is retrieved, NumPy could be helping you perform mathematical operations or manipulations, such as calculating averages, percentages, or returns.
 - It's likely playing a key role in handling large datasets efficiently.
2. Visualization with matplotlib:

- To understand and interpret trends, you're using matplotlib to visualize stock price movements.
- This library enables you to create line plots, bar charts, or candlestick charts for stock data, helping users grasp patterns.

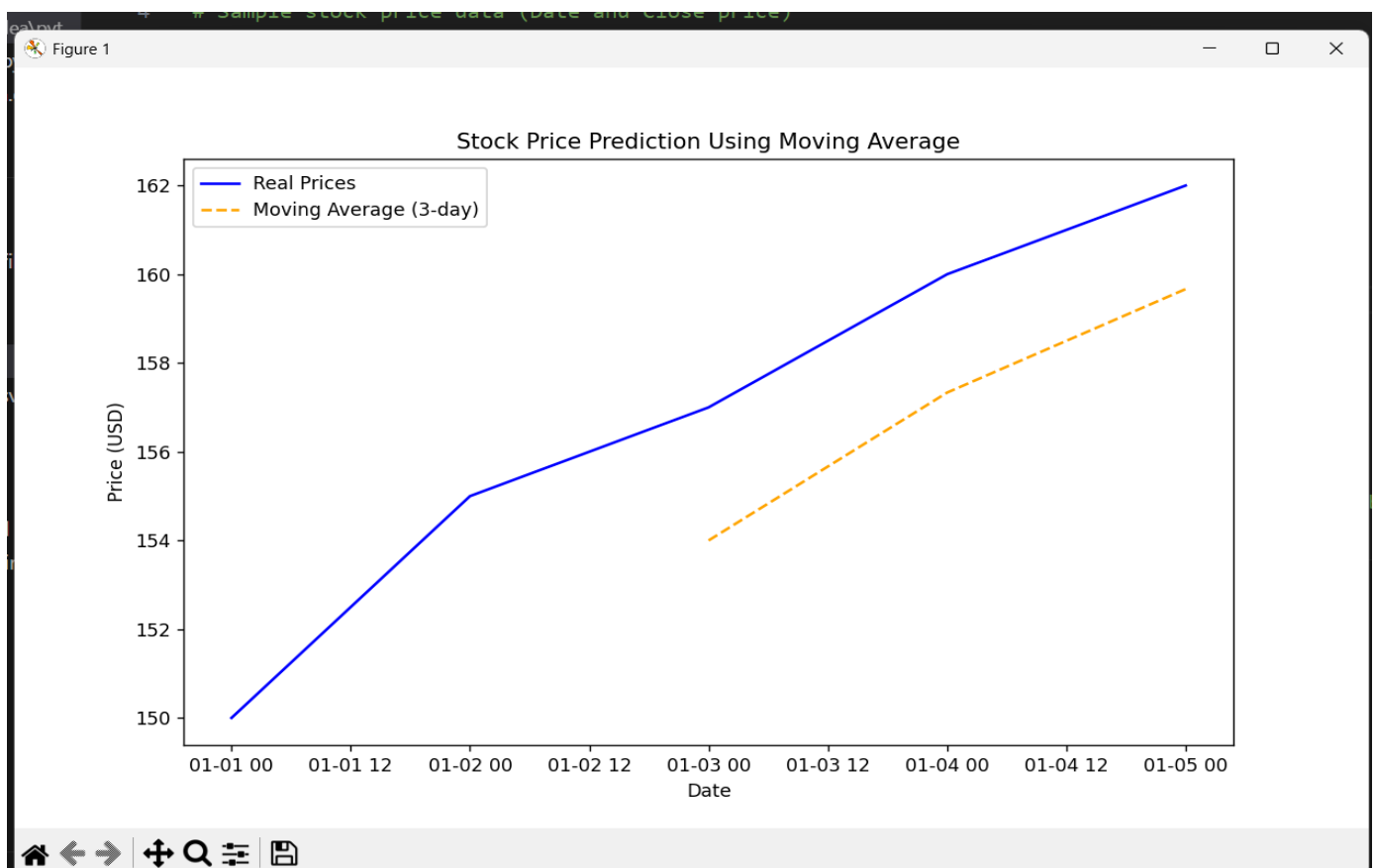
3. Stock Prediction Logic:

- Depending on the complexity, the project might include predictive modelling techniques, like regression analysis, moving averages, or even machine learning algorithms. If that's the case, additional libraries like Scikit-learn or TensorFlow might be used.
- Based on historical data trends, your logic provides predictions about future stock prices.

4. User Interface in Visual Studio Code:

Visual Studio Code acts as the development environment where you write, debug, and run your Python scripts seamlessly.

OUTPUT:



TEAM MEMBERS:

1. Tanya Verma
2. Shreya Lakhmani
3. Saksham Sharma
4. Arati Madhav