Project Title: Stock Price Trend Prediction using LSTM

ABSTRACT:

This project presents a deep learning approach to forecast stock market trends using historical data. By utilizing Long Short-Term Memory (LSTM) networks, a type of Recurrent Neural Network (RNN), the model is trained to predict future stock closing prices based on a window of past values. The aim is to help users understand how machine learning can be used in time series prediction problems, particularly in financial data analysis. The results indicate that LSTM models are capable of capturing temporal dependencies and trends in stock prices with reasonable accuracy.

INTRODUCTION:

Predicting stock prices is a well-known challenge due to the market's volatility, nonlinear behaviour, and numerous influencing factors. Traditional statistical methods often fall short when modelling the complex patterns in financial time series data. Recent advances in deep learning offer promising solutions. In this project, we focus on predicting stock price trends using LSTM neural networks, which are highly effective for sequence prediction problems. Our goal is to predict the next day's closing price of a stock (TATAMOTORS.NS) based on its past 60 days of data.

TOOLS AND TECHNOLOGIES USED:

Component	**Details**
Language	Python
Libraries	pandas, NumPy, matplotlib, scikit-learn, TensorFlow, keras, yfinance
Model	LSTM (Long Short-Term Memory)
Dataset	Stock data from Yahoo Finance via yfinance API
IDE	Jupyter Notebook
Stock Duration	Data from 2010-01-01 to 2024-01-01

STEPS INVOLVED IN BUILDING THIS PROJECT:

Step 1: Data Collection

- Fetch stock price data using the yfinance API for a given ticker symbol (e.g., TATAMOTORS.NS).
- Extract relevant columns like Close prices.

Step 2: Data Preprocessing

- Normalize data using MinMaxScaler to bring all values between 0 and 1.
- Create a sliding window of 60 days to predict the 61st day's closing price.
- Split the dataset into training and testing sets.

Step 3: Building the LSTM Model

- Create a Sequential LSTM model using TensorFlow/Keras.
- Layers include:
    - LSTM layer 1 (50 units)
    - LSTM layer 2 (50 units)
    - Dense output layer (1 unit)
- Compile the model using Adam optimizer and mean squared error loss.

Step 4: Model Training

- Train the model on the training set using an appropriate number of epochs.
- Monitor the loss to evaluate training efficiency.

Step 5: Evaluation & Visualization

- Predict on the test set.
- Inverse-transform predictions to original scale.
- Visualize actual vs predicted stock prices using matplotlib.

CONCLUSION:

This project demonstrates that LSTM networks can successfully learn from historical stock price data to make future predictions. Although the model cannot predict with perfect accuracy due to the unpredictable nature of the stock market, it can capture trends and overall direction effectively. This project provides a foundation for more advanced systems that can integrate other inputs like technical indicators or market sentiment for improved performance.