Stock Market Prediction Using LSTM and Django

Introduction

This project aims to develop a web-based Stock Market Prediction System using machine learning techniques. It helps users forecast the future closing prices of stocks based on historical data, enabling better investment decisions. The web application is built using Django for backend logic and user interface, and LSTM (Long Short-Term Memory) model is used for accurate time series prediction.

Abstract

Stock markets are highly volatile and influenced by numerous factors. Accurate prediction is challenging but valuable for investors. This system utilizes an LSTM model trained on past stock data to predict the next day's closing price. The frontend allows users to input a stock ticker, and the backend processes the data and displays predicted results. The system ensures ease of use and a clean user experience.

Tools Used

- Python
- Django Framework
- HTML, CSS, Bootstrap
- MySQL
- Pandas, NumPy, Matplotlib
- TensorFlow/Keras (for LSTM model)

Steps Involved in Building the Project

1. Data Collection: Historical stock data collected using Yahoo Finance API or CSV files.

2. Data Preprocessing: Cleaned and normalized the data to be suitable for training.

3. Model Building: Implemented LSTM-based neural network for stock prediction.

4. Backend Integration: Integrated the model into Django backend.

5. Frontend Development: Designed responsive and user-friendly UI using HTML, CSS, and Bootstrap.

6. Result Display: Displayed prediction results and comparison with previous stock prices.

Stock Market Prediction Using LSTM and Django

7. Deployment: The complete web application runs locally and can be deployed online.

Conclusion

The Stock Market Prediction System built using LSTM and Django demonstrates how machine learning can be effectively applied to financial time series forecasting. While stock prediction remains inherently uncertain, the model provides useful trends and insights. This project also showcases the integration of AI with web development to solve real-world problems.