

Title: Stock Price Trend Prediction with LSTM

Introduction: This project builds an LSTM-based model to forecast the next-day closing price of a stock using historical price data and technical indicators. LSTM networks are well-suited to time-series problems due to their ability to retain long-term dependencies.

Abstract: We fetch historical stock data from Yahoo Finance, compute technical indicators (MA20, MA50, RSI14) and train a two-layer LSTM to predict closing prices. The dataset was split 80/20 for training and testing. To evaluate, we calculate RMSE and visualize predicted vs actual prices.

Tools Used: - Python, yfinance, pandas, numpy - TensorFlow/Keras (LSTM) - scikit-learn, matplotlib - Streamlit

(for optional deployment)

Steps Involved:

1. Data Collection: Download historical OHLC data for the chosen ticker using yfinance.

2.Preprocessing: Calculate MA20, MA50, RSI14; drop missing values; normalize features via MinMaxScaler.

3.Sequence Creation: Create sliding windows with a 60-day lookback to predict the next-day close.

4. Model Architecture: Two LSTM layers (64 & 32 units) with Dropout and Dense output, trained with Adam optimizer and MSE loss.

5. Training Validation: Train for 25 epochs with early checkpointing to save best model weights.

6.Evaluation: Compute RMSE and show a plot comparing actual vs predicted close prices. Save predictions to CSV.

7. Deployment (Optional): A Streamlit app lets users enter a ticker and see a one-step-ahead prediction.

8.Results & Conclusion: The model achieved an RMSE of XXX (replace after run). Visual comparison shows how well the LSTM follows the trend but cannot capture sudden market shocks. Adding indicators improved stability versus using raw close price alone. This project demonstrates the complete pipeline from data collection to deployment.

Files Submitted: - Stock_Price_Prediction.ipynb (notebook) - model_weights.h5 (best weights) - lstm_full_model/ (saved model) - predictions.csv (actual vs predicted) - app.py (Streamlit) - requirements.txt - report.pdf (this 2-page report)

Future Work: - Add multi-step forecasting and ensemble models (e.g., XGBoost stacked with LSTM). – Incorporate news sentiment, macroeconomic indicators, and volume features. - Backtest a simple trading strategy to measure financial performance (Sharpe ratio, CAGR).