

Project Report: Stock Price Trend Prediction with LSTM

1. Project Overview

This project aims to predict stock price trends using a Long Short-Term Memory (LSTM) neural network. The model is trained on historical stock data, enhanced with technical indicators such as Moving Average (MA) and Relative Strength Index (RSI), and is deployed with a user-friendly Streamlit web application for interactive predictions and visualization.

2. Project Structure & Files

- **AAPL_5years.csv**: Historical stock data for Apple Inc. (AAPL) used for training and testing.
- **main.ipynb**: Jupyter notebook containing the full workflow for data preprocessing, model training, evaluation, and initial visualization.
- **lstm_stock_model.h5**: Saved LSTM model after training.
- **app.py**: Streamlit application for interactive predictions, technical indicator calculation, and visualization.
- **LSTM/pyvenv.cfg**: Virtual environment configuration.

3. Data Preparation

- The dataset consists of 5 years of daily stock data for AAPL, including columns like Close, High, Low, Open, and Volume.
- Data cleaning ensures all values in the 'Close' column are numeric; missing or invalid values are handled.
- Technical indicators are computed:
 - **MA_20**: 20-day moving average of the closing price.
 - **RSI_14**: 14-day Relative Strength Index, indicating momentum.

4. Feature Engineering

- The LSTM model is primarily trained on the normalized 'Close' price.
- For enhanced analysis, MA_20 and RSI_14 are computed and visualized in the app.
- Sequences of length 60 are created from the normalized data to serve as input for the LSTM.

5. Model Development

- **Architecture**: The model is a Sequential Keras model with one LSTM layer (50 units) and one Dense output layer.
- **Training**: The model is trained for 20 epochs using the Adam optimizer and mean squared error loss.
- **Validation**: 20% of the data is reserved for validation.

6. Evaluation & Visualization

- **Metrics**: Model performance is evaluated using Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).
- **Visualization**:
 - The app plots actual vs. predicted prices.
 - The 20-day moving average is shown alongside prices.
 - RSI is plotted in a separate subplot, with overbought/oversold levels (70/30) highlighted.

7. Streamlit Application

- **Features**:
 - Upload any stock CSV with a 'Close' column.
 - Automatic calculation and display of MA_20 and RSI_14.
 - Interactive visualization of actual, predicted prices, MA_20, and RSI_14.
 - Performance metrics (RMSE, MAE) displayed.
- **Usage**:
 - Run the app with `streamlit run app.py`.
 - Access via browser at `http://localhost:8501`.

8. Key Learnings & Improvements

- LSTM models can capture stock price trends, but prediction remains challenging due to market noise.
- Including technical indicators like MA and RSI aids in analysis and may improve model performance if used as input features.
- The Streamlit app provides an accessible interface for both technical and non-technical users.

9. Future Work

- Integrate more technical indicators and use them as features in the LSTM model (multivariate input).
- Experiment with more advanced architectures (e.g., stacked LSTM, attention mechanisms).
- Deploy the app for real-time predictions with live data.
- Add support for other stocks and timeframes.

10. Tools & Technologies

- **Python** (pandas, numpy, scikit-learn, matplotlib)
- **Keras** (TensorFlow backend)
- **Streamlit** (for the web UI)
- **Jupyter Notebook** (for development and experimentation)