Stock Price Prediction Using LSTM Neural Networks

Comprehensive Project Report

Project Resources

- **Project Implementation**: Google Colab Notebook
- **LSTM Background**: Detailed LSTM Notes
- **Technical Indicators Guide**: Technical Analysis Notes

Executive Summary

This project implements a Long Short-Term Memory (LSTM) neural network for stock price prediction using historical market data. The model demonstrated exceptional performance with a testing R^2 score of 0.93 and a Mean Absolute Percentage Error (MAPE) of 2.82% on the test dataset. A basic trading strategy based on the model's predictions showed a significant return of 210.58% on the initial investment, highlighting the potential practical applications of the system.

1. Introduction

1.1 Problem Statement

Stock price prediction remains one of the most challenging problems in financial analysis due to:

- High market volatility
- Complex dependencies on multiple factors
- Non-linear relationships between variables
- Influence of external events and market sentiment

1.2 Why LSTM?

LSTM networks were chosen for this project because they:

- Can capture long-term dependencies in time-series data
- Mitigate the vanishing gradient problem common in traditional RNNs
- Excel at learning sequential patterns
- Can maintain memory of relevant historical information

2. Methodology

2.1 Data Collection

- Source: Yahoo Finance (yfinance library)
- Stock: Apple Inc. (AAPL)
- Period: 2015-03-04 to 2022-03-31
- Data Points: 1,783 trading days
- Features: Open, High, Low, Close prices, and Volume

2.2 Data Preprocessing

1. Data Cleaning:
   - Checked for missing values (none found)
   - Verified data consistency
   - Ensured proper datetime indexing
2. Feature Engineering:
   - Simple Moving Average (SMA)
   - Exponential Moving Average (EMA)
   - Relative Strength Index (RSI)
   - Moving Average Convergence Divergence (MACD)
   - Bollinger Bands
   - Volatility measures
3. Feature Selection: Final features chosen based on correlation analysis:
   - Trading Volume
   - RSI
   - MACD
   - Bollinger Band Middle
   - Volatility

2.3 Model Architecture

```
Model Structure:
1. Input Layer: (time_steps=5, features=5)
2. LSTM Layer 1: 64 units, tanh activation
3. Dropout Layer: 0.2 rate
4. LSTM Layer 2: 32 units, tanh activation
5. Dropout Layer: 0.2 rate
6. Dense Layer: 1 unit (output)
```

3. Implementation Details

3.1 Data Preparation

- MinMaxScaler for feature normalization
- Sequence creation (5 time steps)
- 80-20 train-test split
- Maintained time series order in split

3.2 Training Configuration

- Optimizer: Adam
- Loss Function: Mean Squared Error
- Batch Size: 32
- Early Stopping: Patience=10
- Epochs: Up to 100 (with early stopping)

4. Results and Analysis

4.1 Model Performance Metrics

Training Performance:

- MSE: 15.15
- MAPE: 8.20%
- RMSE: 3.89
- R² Score: 0.97

Testing Performance:

- MSE: 23.52
- MAPE: 2.82%
- RMSE: 4.85
- R² Score: 0.93

5. Key Findings

1. Model Accuracy:
    - High R² scores indicate strong predictive capability
    - Low MAPE suggests reliable percentage accuracy
    - Consistent performance across training and testing sets
2. Trading Performance:
    - Significant returns on basic strategy implementation
    - Strategy showed good market timing ability
    - Results suggest practical applicability
3. Feature Importance:
    - Technical indicators improved model performance
    - Volume and price-based indicators provided complementary signals
    - Feature selection reduced noise and improved efficiency

6. Conclusions

The LSTM-based stock prediction model successfully demonstrated:

1. Strong predictive accuracy for stock price movements
2. Practical applicability through trading strategy results
3. Effective capture of market trends and patterns
4. Robust performance across different market conditions

7. Future Enhancements

7.1 Model Improvements

1. Architecture Enhancements:

   - Implement attention mechanisms
   - Explore hybrid architectures (CNN-LSTM, Transformer-LSTM)
   - Add bidirectional LSTM layers
   - Experiment with deeper networks

2. Feature Engineering:

   - Add sentiment analysis
   - Include macroeconomic indicators
   - Incorporate sector-specific metrics
   - Develop custom technical indicators

7.2 Data Enhancements

1. Additional Data Sources:

   - Order book data
   - Options market information
   - News and social media feeds
   - Alternative data sources

2. Multiple Timeframes:

   - Intraday data
   - Weekly patterns
   - Monthly trends
   - Market regime analysis

8. Getting Started

1. Access the Google Colab notebook
2. Review LSTM and Technical Analysis notes
3. Follow implementation steps
4. Experiment with parameters
5. Test suggested enhancements

9. Acknowledgments

Special thanks to the open-source community and the providers of educational resources that made this project possible.

This report serves as a comprehensive guide for understanding the implementation, results, and future directions of the stock price prediction project using LSTM neural networks.