FINAL PROJECT

AN ALGORITHMIC APPROACH

TO

PREDICT HONDA’S STOCK MARKET TRENDS

Dr. Mahdi Aliyari-Shoorehdeli

Amir Mohammad Saffar

40006783

TABLE OF CONTENTS

Links:

Honda Motors Stock Data (1980-2024)

Colab notebook link

GitHub link

Introduction to the Data:

The dataset used in this project pertains to the stock performance of Honda Motor Corporation. The dataset is sourced from Yahoo Finance and spans the years 1980 to 2024, providing over four decades of stock data. It includes key metrics such as:

- Adjusted Close Price
- Opening Price
- Closing Price
- High Price
- Low Price
- Trading Volume

Project Goals:

The primary aim of this project is to use this dataset to predict the future stock prices of Honda using deep learning models. Given the comprehensive nature of the dataset, it is ideal for identifying trends and predicting stock prices.

Features and Goals of the Dataset:

- **Features:** The data is recorded on a daily basis and includes all necessary parameters for time-series analysis.
- **Goals:** To identify price trends, predict future prices, and analyze market behavior to aid investment decisions.

Process and Methodology:

Project Workflow:

1. **Data Preprocessing:**
    - Normalize raw data to a 0-1 range and create 60-day sequences as model input.
2. **Building the Model:**
    - Use an LSTM Neural Network for its ability to learn temporal patterns.
    - Key components: LSTM layer, Dropout layer (to reduce overfitting), and a Dense layer for output.
    - Train the model using the Adam optimizer and MSE loss function.
3. **Training and Evaluation:**
    - Split the dataset: 80% for training, 20% for testing.
    - Evaluate the model using R^2 Score (accuracy) and MAE (average error).
4. **Comparison and Analysis:**
    - Compare predicted values with actual data via graphs and analyze market trends.

Introduction:

Stock price forecasting takes a major part in financial market analysis, providing investors and traders with information for making meaningful decisions. The project aims to develop a deep learning-based approach for Long Short-Term Memory (*LSTM*) stock price prediction. LSTM is more suited for time-series forecasting because it is able to capture long-term dependencies in sequential data.

Preprocessing Steps:

- Normalization: The data is scaled between 0 and 1 using MinMaxScaler.
- Sequence Creation: A rolling window of the past 60 days is used to predict the next day's stock price.
- ***Train-Test-Validation Split:***
  - ✓ 80% Training
  - ✓ 10% Validation
  - ✓ 10% Testing

Model Architecture:

The LSTM model is designed with:

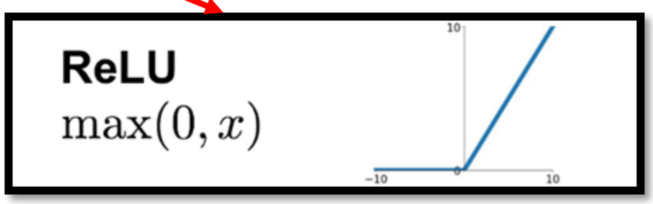
- LSTM Layer: 50 neurons with ReLU activation.
- Dropout Layer: 20% dropout (*preventing overfit*).
- Dense Output Layer: Single neuron for predicting the next day's price.
- Loss Function: Mean Squared Error (MSE).
- Optimizer: Adam optimizer for adaptive learning rate adjustment.

Training and Validation Process:

- Epochs: 20
- Batch Size: 32
- Validation Data: 10% of the dataset to monitor model performance.
- Callbacks Used:
  - ✓ Early Stopping: Stops training when validation loss stops improving.

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 50)	10,400
dropout (Dropout)	(None, 50)	0
dense (Dense)	(None, 1)	51

```
model.add(LSTM(50, activation="relu", input_shape=(X.shape[1], X.shape[2])))
```



Model Evaluation Metrics:

- R^2 Score: Measures how well predictions match actual values.
- Mean Absolute Error (*MAE*): Measures the average difference between actual and predicted values.

Performance Metrics:

Sets	R^2 Score	MAE
Training	*0.9970*	*0.4646*
Testing	*0.9459*	*0.6122*
Validation	*0.9375*	*0.5745*

Results & Visualization:

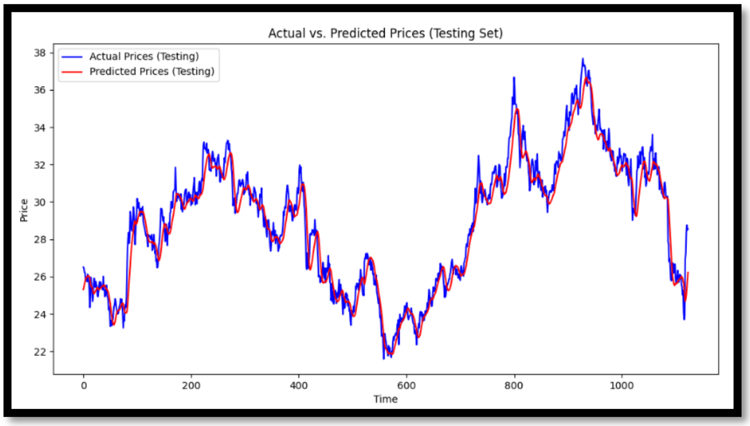


Figure 1 Actual vs. predicted prices for testing

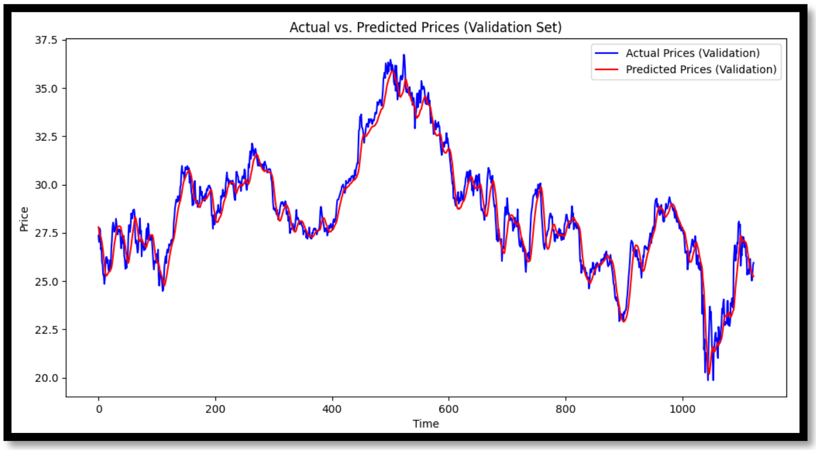


Figure 2 Actual vs. predicted prices for validation

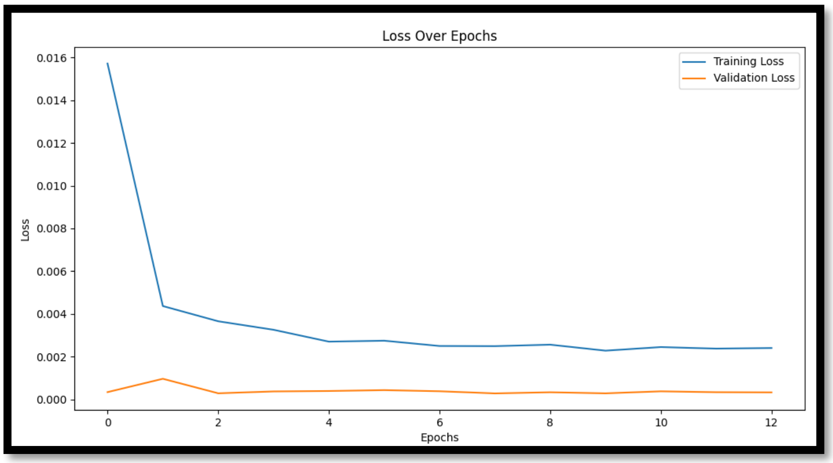


Figure 3 loss over Epochs

Conclusion:

This project demonstrates the capability of LSTM networks to predict stock prices with strong performance in capturing trends. The use of both validation& test sets ensures that the model generalizes well. On the other hand, the model shows room for improvement, especially in handling volatile price movements. By integrating more advanced architectures and additional features, future iterations could achieve even greater accuracy and robustness.

Potential Enhancements:

- Using Bidirectional LSTM: By processing data in both forward and backward directions, Bidirectional LSTMs could enhance feature learning and prediction accuracy. But also, the runtime is going to increase significantly.
- Experimenting with Different Sequence Lengths: Testing shorter or longer input sequences (30 or 90 days) might improve predictions depending on the dataset characteristics.
- Adding More Features: Incorporating additional features like Open Price, Moving Averages.

Future Work

Transformers which are known for their great performance in time-series forecasting, could illustrate better long-term predictions compared to LSTM.