Project Report: Stock Price Prediction Using LSTM

Intern Name: Pyata Nandini
Internship Title: Tech Internship at Cloudcredits
Internship Duration: April 4, 2025 – June 4, 2025
Project Title: Stock Price Prediction Using LSTM
Submission Date: [June 4,2025]

Acknowledgment

I would like to express my sincere gratitude to Cloudcredits for providing this internship opportunity and supporting my learning experience. I am also thankful to my mentors and peers for their continuous guidance throughout the internship period.

Table of Contents

---

1. Introduction

Stock price prediction is a significant challenge in the finance domain. Traditional statistical methods often fail to capture the complex patterns in stock market data. In this project, we implement a Long Short-Term Memory (LSTM) neural network to predict stock prices using historical data. The model is trained and tested on real stock data fetched using the `yfinance` library.

---

2. Objective

To build and evaluate an LSTM-based deep learning model that can predict the future closing price of a stock using historical closing prices.

---

3. Technology Stack

- **Programming Language:** Python
- **Environment:** Jupyter Notebook / Google Colab
- **Libraries Used:**
  - `numpy`
  - `pandas`
  - `matplotlib`
  - `scikit-learn`
  - `keras`
  - `tensorflow`
  - `yfinance`

---

4. Methodology

1. **Data Collection:**
   - Data was retrieved from Yahoo Finance using `yfinance`.
   - The stock symbol was used to fetch historical closing prices.
2. **Data Preprocessing:**
   - Normalization using `MinMaxScaler`
   - Creating training sequences of past 60 time steps
   - Splitting data into training and testing sets
3. **Model Building:**
   - Constructed an LSTM model using Keras
   - Added Dense layers for output
   - Used `mean_squared_error` as the loss function
4. **Training and Evaluation:**
   - Trained the model on training data
   - Plotted predictions against actual values for evaluation

---

5. Implementation Details

- LSTM model architecture:
  - Input layer with 60 time steps
  - LSTM layer with 50 units
  - Dense output layer
- Optimizer: Adam
- Loss: Mean Squared Error

- Visualization using `matplotlib` to compare predicted vs actual prices

Code Snippet:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
from keras.models import Sequential
from keras.layers import LSTM, Dense
import yfinance as yf

# Load stock data
data = yf.download('AAPL', start='2010-01-01', end='2020-01-01')
closing_prices = data['Close'].values.reshape(-1, 1)

# Normalize the data
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(closing_prices)

# Prepare training data
x_train, y_train = [], []
for i in range(60, len(scaled_data)):
    x_train.append(scaled_data[i-60:i, 0])
    y_train.append(scaled_data[i, 0])
x_train, y_train = np.array(x_train), np.array(y_train)
x_train = np.reshape(x_train, (x_train.shape[0], x_train.shape[1], 1))

# Build LSTM model
model = Sequential()
model.add(LSTM(units=50, return_sequences=False,
input_shape=(x_train.shape[1], 1)))
model.add(Dense(units=1))

# Compile and train
model.compile(optimizer='adam', loss='mean_squared_error')
model.fit(x_train, y_train, epochs=10, batch_size=32)
```

- **LSTM Model Summary:**

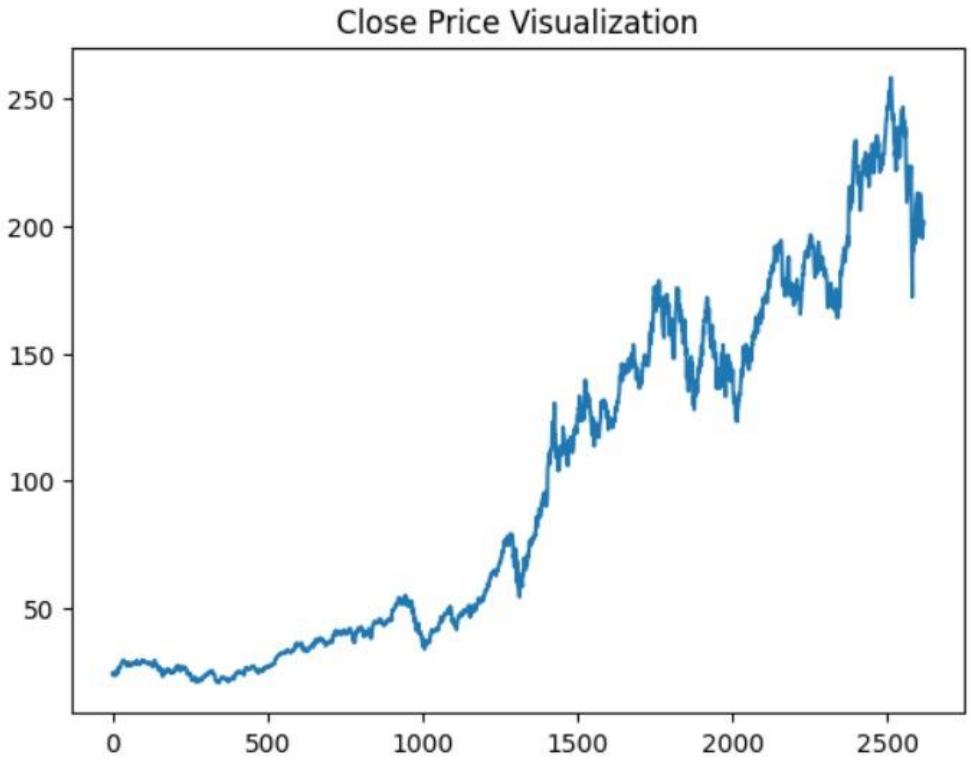
---

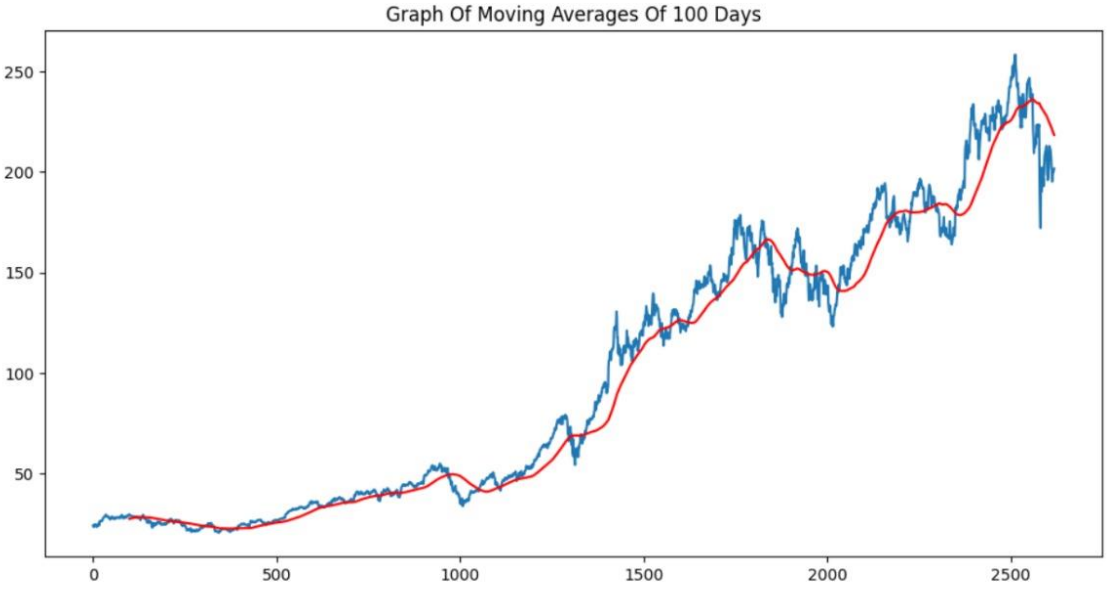
6. Results and Evaluation

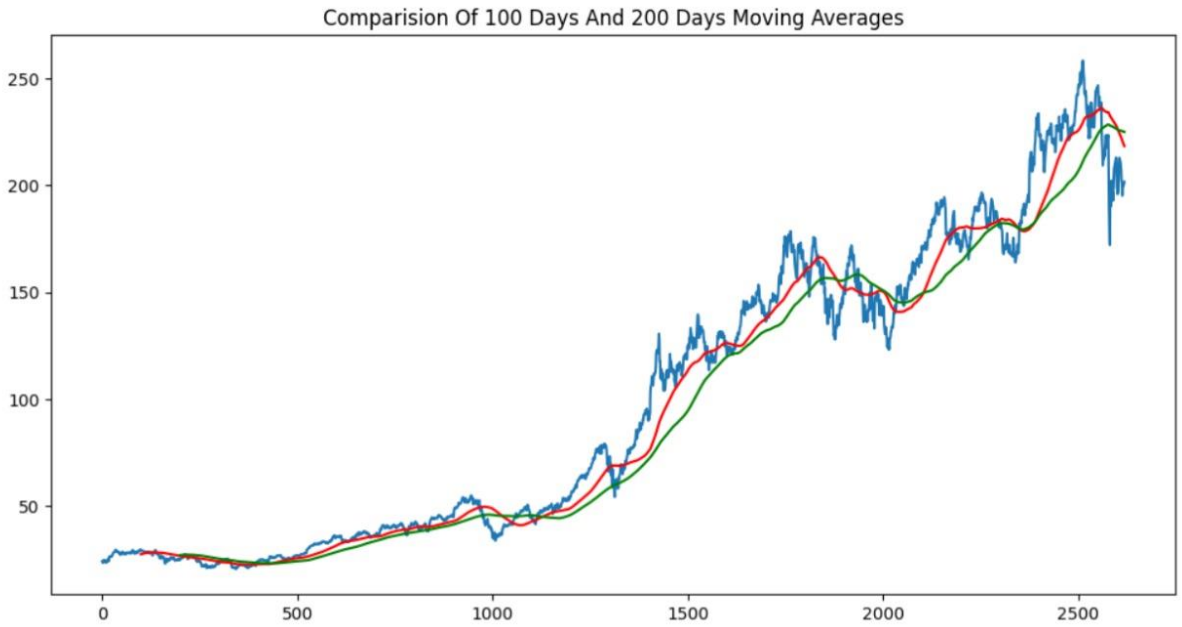
- The model was trained successfully and provided good predictions visually aligned with actual trends.
- RMSE was used to evaluate model performance.
- The results showed the model could effectively learn from past data to predict future values within a reasonable error range.

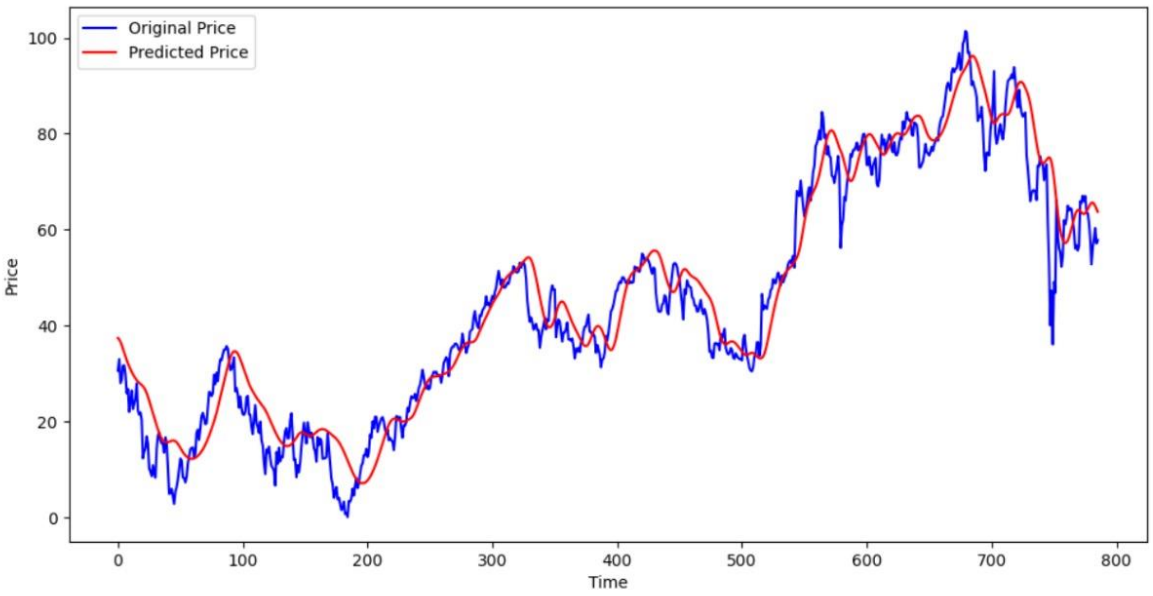
Graphs:

- **Training vs Validation Loss Curve:**









These visualizations helped in understanding the learning progress of the model and how close the predicted values were to the actual values.

---

7. Conclusion

This project demonstrated how LSTM models can be used to predict stock prices. The accuracy of the model was sufficient for short-term trend analysis. Future improvements could involve adding more features like volume, open/high/low prices, and tuning hyperparameters.

---

8. References

- https://github.com/034adarsh/Stock-Price-Prediction-Using-LSTM
- https://keras.io/
- https://www.tensorflow.org/
- https://pandas.pydata.org/
- https://finance.yahoo.com/

---

9. Appendix

Complete code for the model is provided in the attached Jupyter Notebook file named `LSTM_model.ipynb`. All steps from data loading to prediction are documented and reproducible.

GitHub Link: https://github.com/Pyatanandini/Cloudcredits.git

---

End of Report