NTT Stock Price Prediction
(Time Series Forecasting Model)

Submitted By
Pragya Gupta

Background

Stock price prediction is crucial for investment decisions and risk management.

Accurate forecasting helps investors, traders, and financial institutions.

Influenced by global events, investor sentiment, and regulatory changes, make predictions complex and uncertain.

It requires analyzing vast amounts of data, including historical trends, financial ratios, macroeconomic indicators, and social media sentiment.

As a tech and telecommunications giant, NTT operates in a dynamic industry, making stock price forecasting both challenging and insightful.

Data Preprocessing

- Data was checked for null values and duplicate values.
- Volume and Percentage change data type was converted in compatible format.
- Volume was normalized for further processing.
- Basic Data statistics and EDA was carried out.

Data Overview

Time Period: 1987-02-12 to 2024-08-01

Privatization drove up stock prices as expectations for greater profitability and market efficiency grew.

The crash of the dot-com bubble (2000 to 2002) and economic challenges led to a sharp decline.

Increased investments in new technologies and economic improvements in Japan contributed to NTT's stock price increase during the period from 2015 to 2020.

The inflated valuations of tech companies during the dot-com bubble in 1992-.

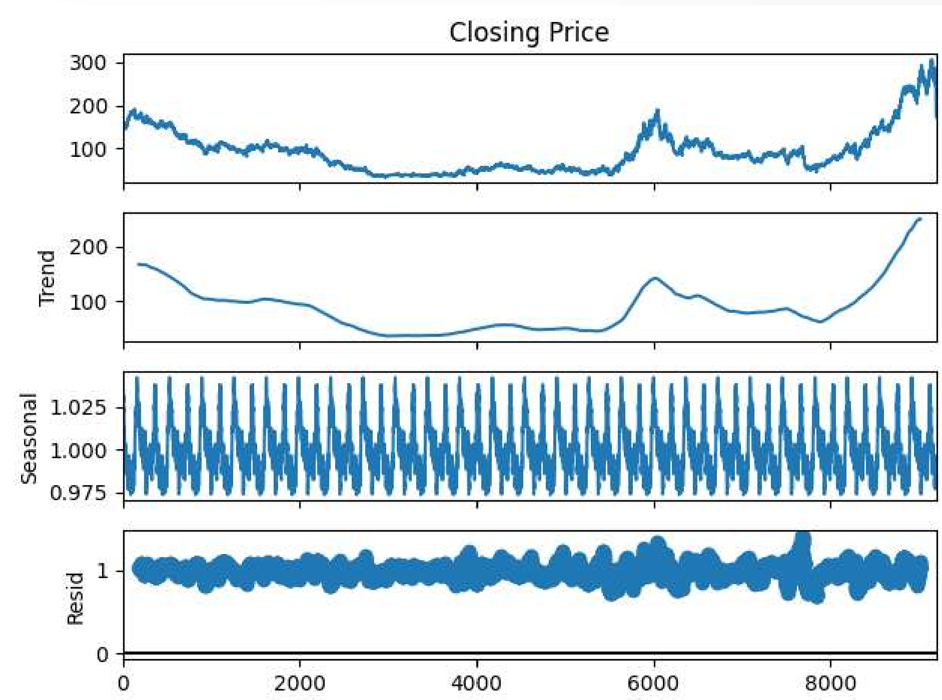
Stable but modest growth during the telecom industry's maturation phase during the period from 2002 to 2015

The effects of the pandemic, coupled with large investments in 5G technology stabilized NTT's stock price at a higher level than before.

Technical Data Overview:

Based on Correlation analysis,

- Strong correlation among Closing price (CP), high price (HP), Low price (LP) and opening price (OP). Thus some features are redundant.
- Volume has negative correlation which is interesting as it suggests volume of trades increases, the stock price tends to fall.
- Percentage Change has a negligible correlation with Closing Price (0.0172). This variable might not add much predictive value.
- Based on ACF test, the time series is non-stationary, meaning it has trends or seasonality. This will affect our model selection, as many time-series models (like ARIMA) require stationary data.



Feature Engineering

A. Lagged Features (Lag Variables): Since stock prices show autocorrelation, lag-1 and lag-2 are included for their direct impact on future prices. Lag-7 and lag-30 are added for experimentation.

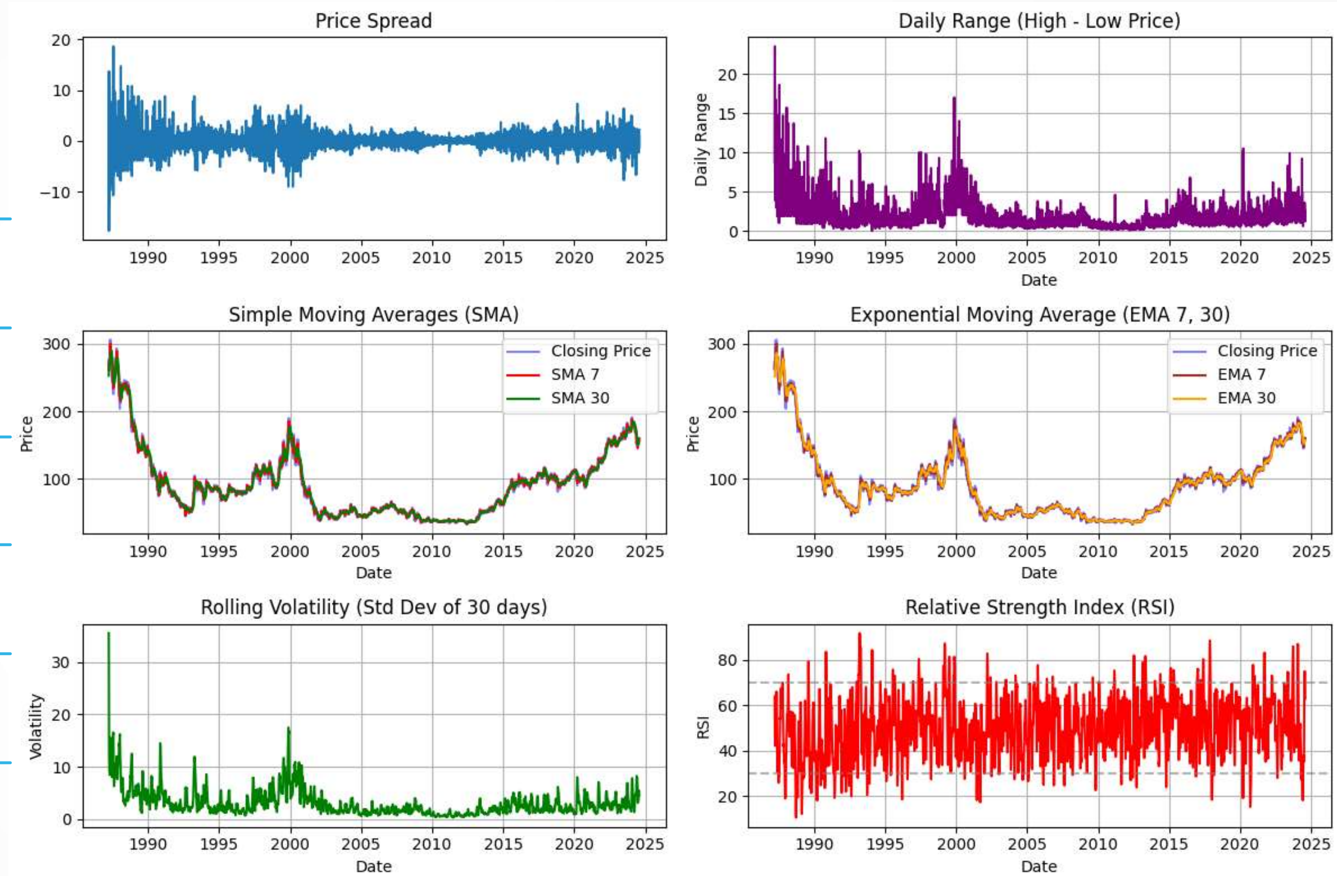
B. Moving Averages & Exponential Moving Averages (EMA): Moving averages smooth data and capture trends. EMA reacts faster to changes, making it crucial for short-term forecasting. We included 7- and 30-day windows.

C. Relative Strength Index (RSI): RSI measures momentum to identify overbought or oversold conditions. It can signal potential price reversals, aiding short-term prediction.

D. Normalized Volume: Volume, which showed a negative correlation (-0.317), is included as a scaled feature. It can act as a leading indicator of price movements, especially on high-volume days.

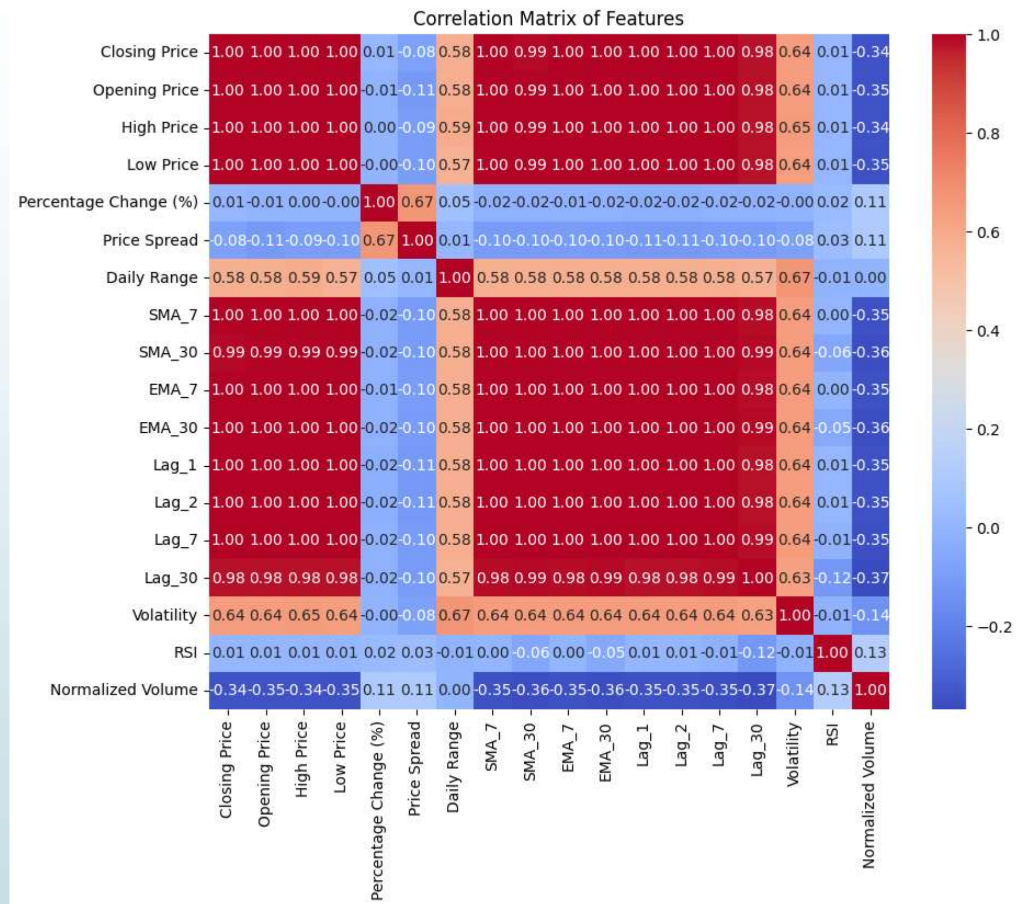
E. Percentage Change: This captures daily stock price momentum, helping the model detect periods of rapid shifts.

F. Price Spread & Price Range: Useful for identifying volatility, these features track fluctuations within the trading day.



Correlation Matrix

Relevant features for our model: Daily Range, SMA_30, EMA_7, EMA_30, Lag_1, Lag_2, Lag_7, Volatility, Normalized Volume.



Model Selection: LSTM

SARIMA as Baseline: I started with SARIMA due to its ability to handle time-series data with trends and seasonality. However, it struggles with non-linear patterns and long-term dependencies common in stock prices.

Switch to LSTM: I shifted to LSTM because it handles non-stationary data, captures complex, non-linear relationships, and models long-term dependencies. This makes LSTM better suited for forecasting stock prices in dynamic, volatile environments.

Model Architecture: LSTM

- 2 LSTM layers.
- Dropout layers for regularization with Dropout rate 0.2.
- Dense layer for final prediction.

Training Settings:

- Optimizer: Adam.
- Loss Function: Mean Squared Error (MSE).

Evaluation Metrics: RMSE

- RMSE measures the average magnitude of errors between predicted and actual stock prices, giving more weight to larger errors.
- It penalizes large prediction errors, which is important in stock forecasting where big deviations can be costly.
- It provides results in the same units as the target variable (price), making it easier to interpret.

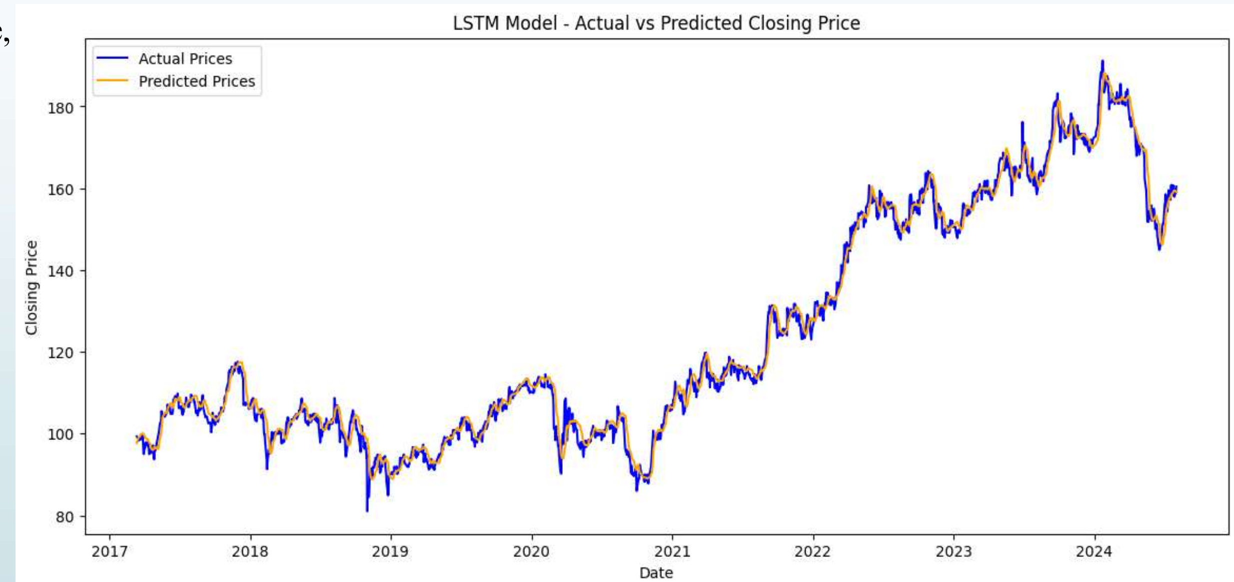
Prediction Results using basic features

Features chosen: Closing Price, Opening Price, High Price, Low Price, Normalized Volume

Model Hyperparameters: LSTM units=50, Time_steps=30, batch_size= 64, epochs=10

Training RMSE: 3.2232

Test RMSE: 2.731



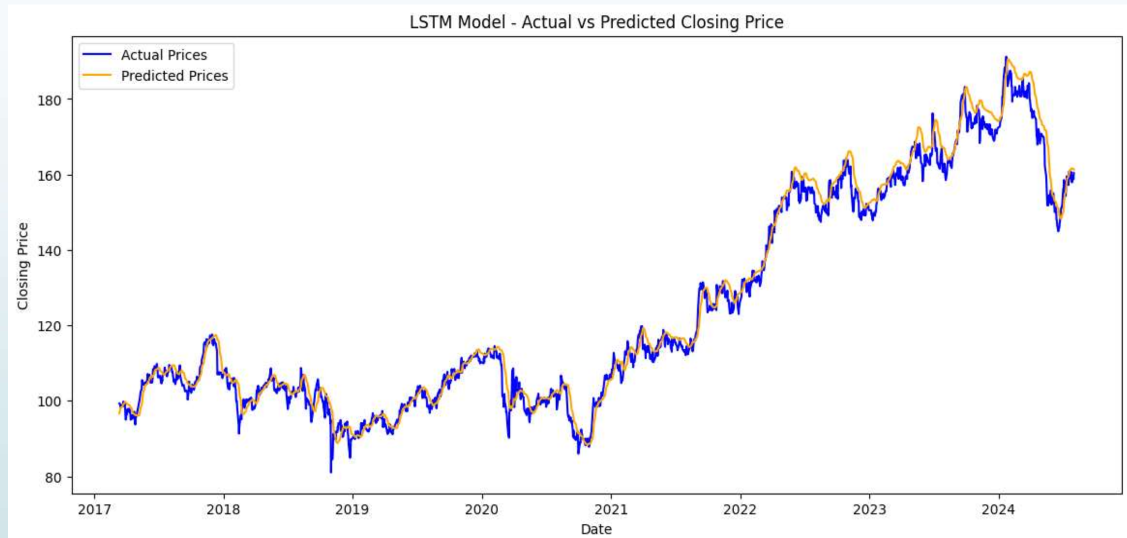
Prediction Results using Derived features

Features chosen: Closing Price, Daily Range, SMA_30, EMA_7, Volatility, Lag_1, Normalized Volume

Model Hyperparameters: LSTM units=50, Time_steps=30, batch_size= 64, epochs=10

Training RMSE: 3.9424

Test RMSE: 3.5256



Validation Strategy: Optimizing Hyperparameters

- LSTM Units: The model with 100 LSTM units generally performed better than models with higher LSTM units like 50 and 128.
- Batch Size: A batch size of 12 appears to be the sweet spot for performance. Larger batch sizes like 32 and 64 performed worse.
- Epochs: Higher epochs (20) seem to provide better results than 10 epochs, which indicates that training longer might help and could prevent from overfitting

Validation Results

Best Model Hyperparameters:

LSTM Units 100

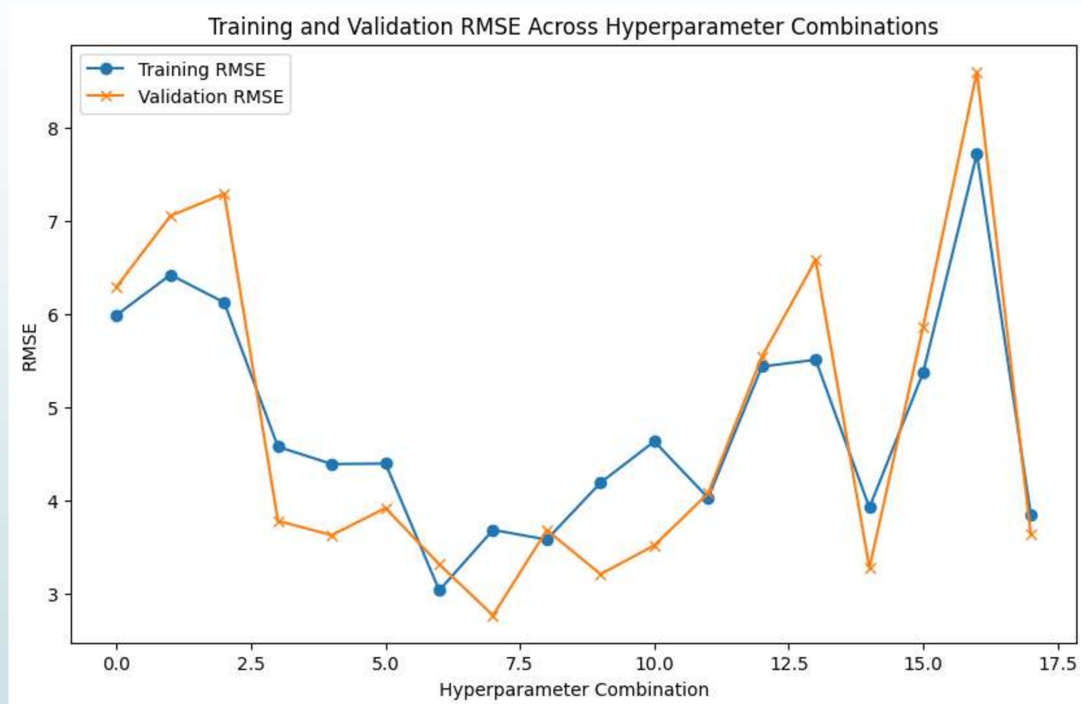
Batch Size 12

Time Steps 30

Epochs 20

Training RMSE 3.689220

Validation RMSE 2.772545



Summary

Model is exported in Github repository for future predictions.

Thank you