Comparison of price forecasting with and without differencing

TATA STEEL LIMITED

Name of Intern: Shreyansh Khaitan
Duration: 4 weeks (from 3rd July)

Guide: Ms. Amrita Biswas
Sponsor: Ms. Asha Nair
Reference Number: VT20241504
Name of Institute: VIT University, Chennai
Domain: Data & Analytics

Introduction

Accurate price forecasting is crucial in the industrial sector for making informed investment decisions and developing effective procurement strategies. One pertinent example is coal, a key commodity for Tata Steel. The company actively purchases coal to meet its business needs, and procuring coal at the best price is essential for maximizing profits. Therefore, identifying the most effective methods for forecasting coal prices is of utmost importance.

My project, undertaken as part of an internship at Tata Steel, aims to compare the effectiveness of price forecasting models with and without differencing. Here, I have worked on stock prices of Tata Steel Limited.

Time series forecasting occurs when you make scientific predictions based on historical time stamped data. It involves building models through historical analysis and using them to make observations and drive future strategic decision-making. An important distinction in forecasting is that at the time of the work, the future outcome is completely unavailable and can only be estimated through careful analysis and evidence-based priors.

Stock prices often exhibit non-stationarity, making traditional forecasting methods less effective. Differencing is a technique used to transform non-stationary time series data into a stationary one by subtracting the previous observation from the current observation. This project explores how applying differencing impacts the accuracy of stock price forecasts.

Advanced machine learning models are used, such as Long Short-Term Memory (LSTM) neural networks and Multilayer Perceptron (MLP), Trigonometric, Box-Cox transform, ARMA errors, Trend and Seasonal components (TBATS) and Exponential Smoothening, which are known for their ability to capture temporal dependencies in sequential data.

Figure 1. Time series forecasting

The steps followed for forecasting are given below in brief:

1. Data Collection and Preprocessing: Gathering historical stock price data for Tata Steel and preparing it for analysis by cleaning and normalizing the data.

2. Model Training and Evaluation: Training models on both differenced and non-differenced data and evaluating their performance using metrics such as Mean Absolute Percentage Error (MAPE) and Directional Accuracy.

3. Forecasting: Using the trained models to forecast stock prices for the next 7 days and comparing the results.

4. Analysis and Comparison: Analyzing the forecast accuracy and the impact of differencing on the model performance.

Through this project, the aim is to provide insights into the benefits and drawbacks of using differencing in time series forecasting and identifying the most effective approach for predicting stock prices in a volatile market. The findings can help in refining forecasting techniques, ultimately aiding Tata Steel in better financial planning and risk management.

Scope of Project

The primary scope of this project is to evaluate and compare different time series forecasting methods for predicting the stock prices of Tata Steel, with a specific focus on the impact of differencing on forecast accuracy. The following forecasting methods were employed:

1. Exponential Smoothing: A time series forecasting method that uses weighted averages of past observations to forecast future values.

2. TBATS (Trigonometric, Box-Cox transformation, ARMA errors, Trend, and Seasonal components): A complex model designed to handle complex seasonal patterns in time series data.

3. MLP (Multi-Layer Perceptron): A class of feedforward artificial neural network models.

4. LSTM (Long Short-Term Memory): A type of recurrent neural network (RNN) capable of learning long-term dependencies in sequential data.

The project leveraged these methods to forecast stock prices over a 7-day horizon, evaluating the performance of each method using key metrics:

1. Directional Accuracy (DA): Measures the percentage of times the predicted direction of stock price movement matches the actual direction.

2. Mean Absolute Percentage Error (MAPE): A measure of prediction accuracy in a forecasting method, indicating the average absolute percentage error between forecasted and actual values.

3. Validation Accuracy: The percentage of correct predictions on the validation set, indicating the model's performance on unseen data.

To establish benchmark scores, we calculated the directional accuracy and validation accuracy for each method based on the 7-day stock price forecasts. The scope includes identifying the method that outperforms these benchmark scores, thereby providing more accurate and reliable forecasts.

Additionally, the project investigated the effectiveness of differencing in improving the forecasting models. Differencing is a preprocessing step that helps stabilize the mean of a time series by removing changes in the level, making the series more stationary. By comparing the performance of models with and without differencing, we aimed to demonstrate the benefits of this technique in enhancing the accuracy of stock price predictions.

Methodology

The methodology of this project involves several key steps, including data collection, preprocessing, model selection, model training, evaluation, and comparison. Each of these steps is crucial to ensure the accuracy and reliability of the forecasting models used for predicting Tata Steel's stock prices.

Data Collection

The stock price data for Tata Steel was collected, including key attributes such as the opening price, high price, low price, closing price, volume and change percentage. The dataset spanned a total of 10 years, providing a comprehensive basis for analysis and forecasting.

The historical data was taken from investing.com website. Below is the link.

https://www.investing.com/equities/tata-steel

Figure 2. Price graph of Tata Steel Ltd from July 2014 to July 2024

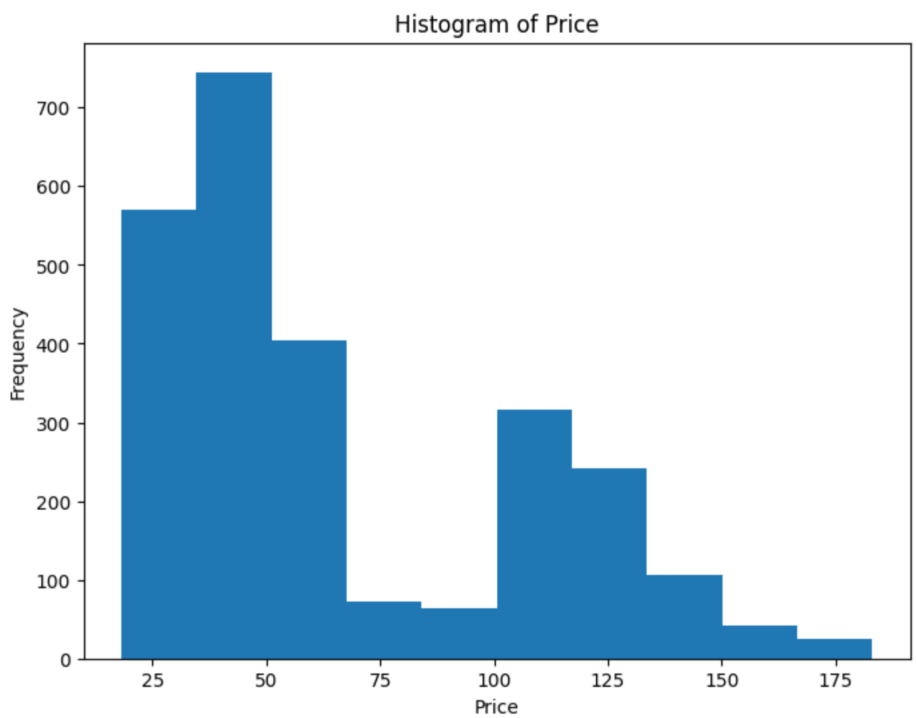

Figure 3. Histogram of price frequency

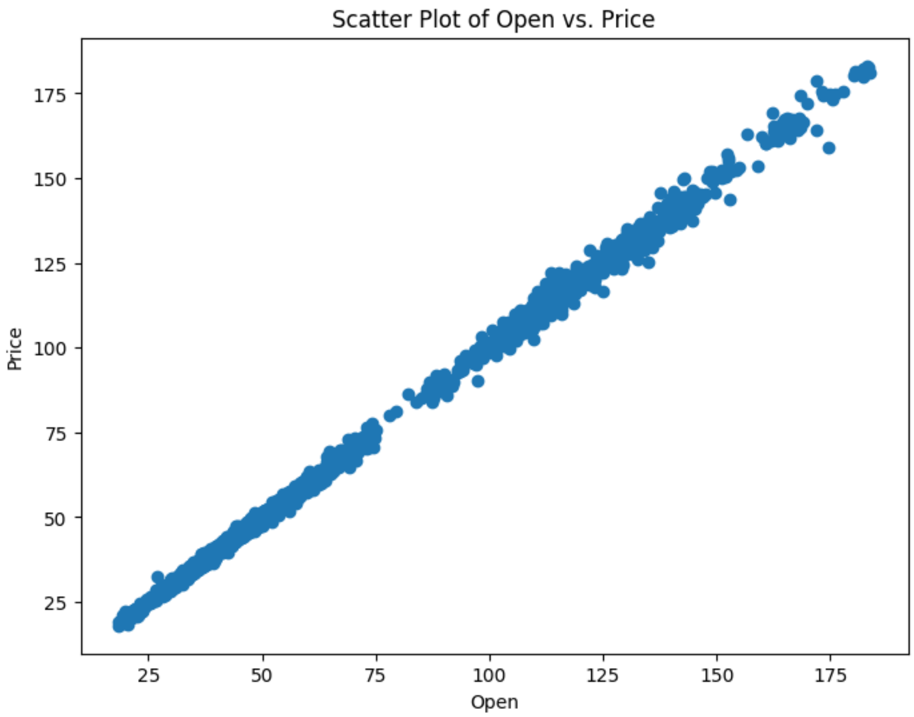

Figure 4. Scatterplot of Open Price vs Close Price

Benchmark Scores

We analyzed three years of Tata Steel data from 2021 to 2024 to calculate benchmark results for model comparison. To predict scores, we compared actual (current day) prices to prices from seven days prior. The difference between these two values was used to determine the changes and direction of the stock.

For example, the price of stock on Day 1 is ₹100, Day 2 it is ₹105 and so on until Day 7 where it is ₹130. Now on Day 8 the price will be ₹135. This price of Day 8 will become the forecasted price of Day 1, i.e., Actual price of Day is ₹100 and forecasted price of Day 1 is ₹135 (which is equal to Day 8 price).

	Date	Price	Open	High	Low	Vol.	Change %
0	2021-06-28	118.40	117.40	118.90	116.30	116600000.0	1.61%
1	2021-06-29	117.25	118.45	119.20	116.56	78040000.0	-0.97%
2	2021-06-30	116.66	118.11	119.10	116.28	80430000.0	-0.51%
3	2021-07-01	116.36	117.19	117.68	115.64	56150000.0	-0.26%
4	2021-07-02	113.60	116.60	116.68	113.02	75730000.0	-2.37%
...	...	...	...	...	...	...	...
737	2024-06-18	181.12	183.80	184.60	180.60	35030000.0	-1.11%
738	2024-06-19	180.02	182.45	182.49	179.31	27480000.0	-0.61%
739	2024-06-20	182.28	181.60	182.95	179.37	38840000.0	1.26%
740	2024-06-21	179.94	179.40	180.90	178.18	65500000.0	-1.28%
741	2024-06-24	177.96	176.16	178.90	175.13	47590000.0	-1.10%

Figure 5. Screenshot of same data from 2021 to 2024

Using this method, we obtained the following values:

- MAPE: 4.02%
- Validation Accuracy: 96%
- Directional Accuracy: 50.94%

Vol.	Change %	Forecasted_Price	Daily_Change(₹)	F(t) – X(t–1)	Actual_Direction	Forecasted_Direction	Directional_Accuracy
35030000.0	-1.11%	172.05	-2.03	-11.10	-1.0	-1.0	1
27480000.0	-0.61%	178.90	-1.10	-2.22	-1.0	-1.0	1
38840000.0	1.26%	180.29	2.26	0.27	1.0	1.0	1
65500000.0	-1.28%	181.33	-2.34	-0.95	-1.0	-1.0	1
47590000.0	-1.10%	182.23	-1.98	2.29	-1.0	1.0	0

Figure 6. Dataset after calculations

Data Preprocessing

Data Cleaning: This step involved handling missing values, outliers, and any anomalies in the dataset. Ensuring the data is clean and free from errors is critical for accurate modeling. Some of the data had to be converted to integer datatypes, for example the volume column was 'string' type data which had to be changed into floating point data.

Normalization: The stock price data was normalized using MinMaxScaler to scale the features within a specific range, typically between 0 and 1. This step is essential for neural network models to ensure efficient training.

Differencing: Differencing was applied to the data to stabilize the mean of the time series and remove trends, making the data more stationary. This step is particularly beneficial for models sensitive to non-stationary data.

Train-Test Split: The dataset was split into training and testing sets, with 70% of the data used for training the models and 30% used for testing and validation. This split ensures that the models are evaluated on unseen data to assess their performance accurately.

Model Selection

Four different forecasting methods were selected for comparison:

I. TBATS (Trigonometric, Box-Cox transformation, ARMA errors, Trend, and Seasonal components)

TBATS is an advanced forecasting method designed to model time series data with complex seasonal patterns. This technique is particularly useful when dealing with data that exhibit multiple seasonal cycles, such as daily data with weekly and annual patterns, or hourly data with daily, weekly, and annual patterns. TBATS extends the capabilities of exponential smoothing by incorporating additional components to capture the intricacies of the data.

Components of TBATS:

1. Box-Cox Transformation: A power transformation that stabilizes variance and makes the data more normally distributed.

2. Trigonometric Seasonal Components: Uses Fourier series to model multiple seasonal periods simultaneously.

3. ARMA Errors: Accounts for autocorrelations in the residuals through ARMA (AutoRegressive Moving Average) components.

4. Trend: Includes linear or exponential trend components to capture long-term movements in the data.

5. Seasonal Components: Models seasonal patterns that may occur at different frequencies, using trigonometric functions.

Procedure:

- Transformation: The time series data undergoes a Box-Cox transformation to stabilize variance and make the series more stationary.
- Decomposition: The transformed series is decomposed into trend, seasonal, and error components.
- Modeling Seasonal Patterns: Seasonal components are modeled using trigonometric functions via Fourier series to capture multiple seasonal periods.
- Combining Components: The model combines exponentially smoothed trend and seasonal components along with ARMA components to account for any remaining autocorrelation in the residuals.
- Hyper-parameter Tuning: TBATS conducts hyper-parameter tuning using the Akaike Information Criterion (AIC) to determine the optimal combination of components.

For this project, TBATS was employed to forecast Tata Steel's stock prices. The model was trained on historical data and validated using recent observations. The results were evaluated using key performance metrics such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Directional Accuracy.

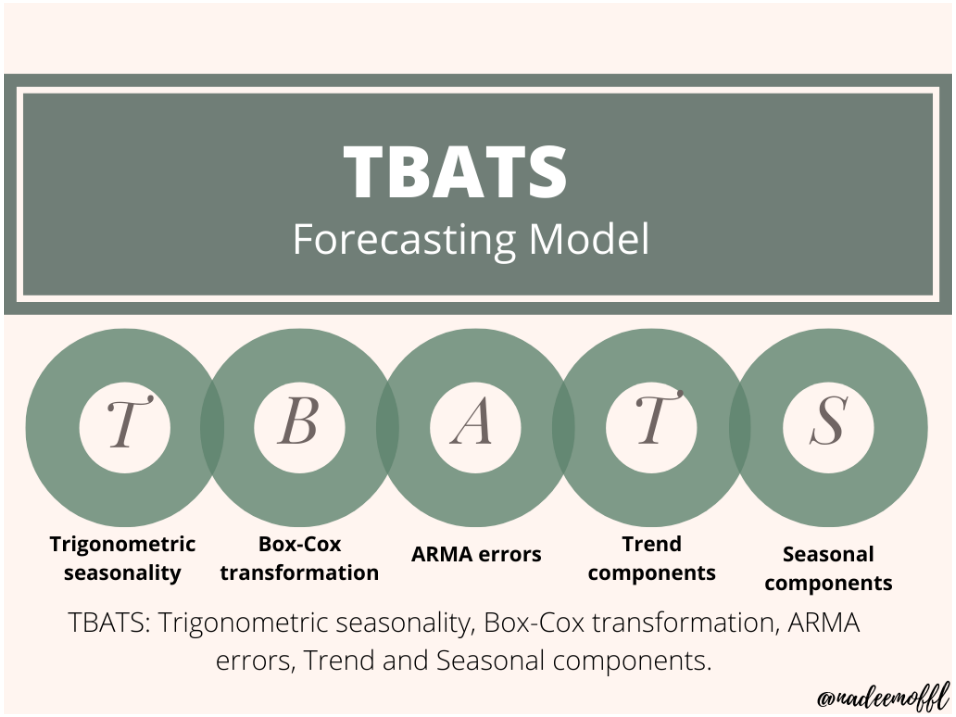

Figure 7. Overview of TBATS model

II. Exponential Smoothing

It is a time series forecasting method that is particularly useful for short-term predictions. It applies weighted averages to past observations, where the weights decrease exponentially over time. This means that more recent observations have a higher impact on the forecast than older ones. The technique is simple yet effective, especially for data with no clear trend or seasonal pattern.

<u>Components of Exponential Smoothing</u>

1. Level (α): The smoothed value of the series at the current time step. It captures the level of the series.

2. Trend (β): The smoothed trend at the current time step. It reflects the increasing or decreasing pattern in the data.

3. Seasonality (γ): The smoothed seasonal component at the current time step. It captures the repeating short-term cycle in the data.

Formula: $s_t = \alpha x_t + (1 - \alpha)s_{t-1} = s_{t-1} + \alpha(x_t - s_{t-1})$

Here,

s_t = smoothed statistic, it is the simple weighted average of current observation x_t

s_{t-1} = previous smoothed statistic

α = smoothing factor of data; $0 < \alpha < 1$

t = time period

Types of Exponential Smoothing:

1. Simple Exponential Smoothing (SES): Used for data without trend or seasonality. It focuses solely on the level component.

2. Holt's Linear Trend Model: Extends SES to capture linear trends in the data by including both level and trend components.

3. Holt-Winters Seasonal Model: Extends Holt's model to account for seasonality by including level, trend, and seasonal components.

For this project, we applied simple exponential smoothing and Holt's linear trend model to forecast Tata Steel's stock prices. The models were trained using historical data and validated using recent observations. The results were evaluated based on key metrics such as Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Directional Accuracy.

III. MLP (Multi-Layer Perceptron)

The Multi-Layer Perceptron (MLP) is a type of artificial neural network that is particularly effective for complex regression tasks, including time series forecasting. Unlike traditional regression models, MLP can capture nonlinear relationships between inputs and outputs, making it a powerful tool for predicting stock prices.

Components of MLP:

1. Input Layer: This layer takes in multiple features from the dataset, such as 'Open', 'High', 'Low', and 'Change %'.

2. Hidden Layers: MLP consists of one or more hidden layers of neurons that process the input features. In our project, we used a single hidden layer with 50 neurons.

3. Output Layer: The final layer of the network produces the forecasted stock price.

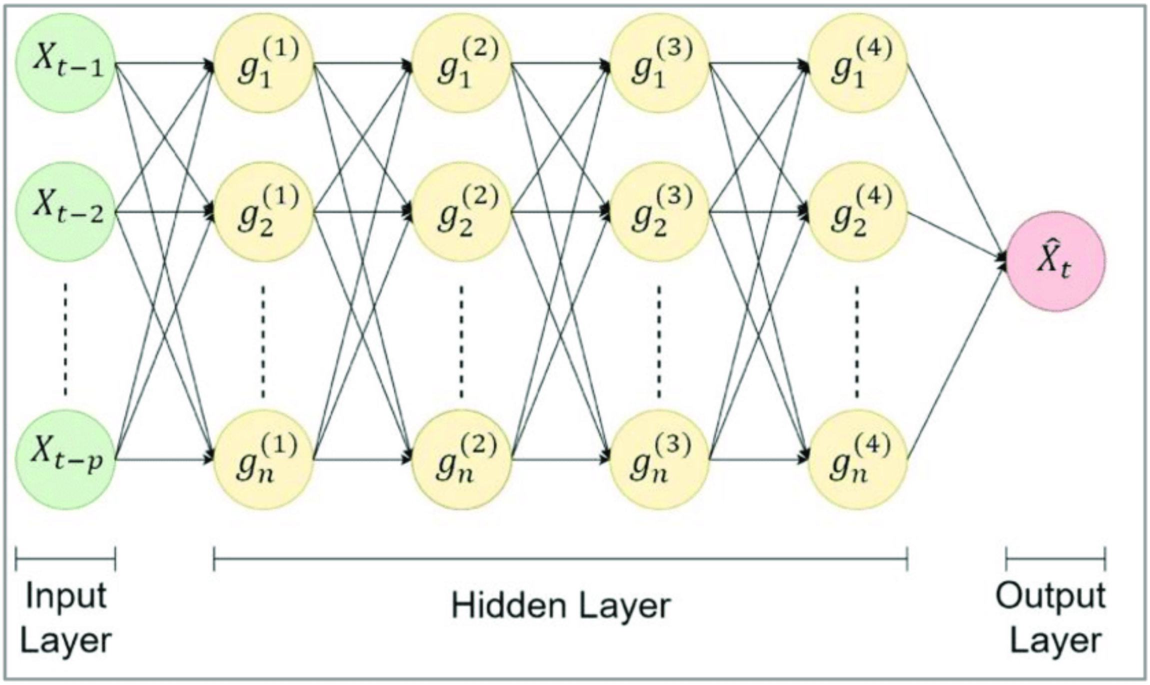

Figure 8. Architecture of MLP

<u>Methodology</u>:

- Data Preparation: The training dataset included features like 'Open', 'High', 'Low', and 'Change %' to predict the 'Price'.
- Model Initialization: The MLP regressor was set up with a hidden layer containing 50 neurons and a maximum of 1000 iterations for training.
- Training: The model was trained on the prepared dataset, learning from historical data to predict future stock prices.

In summary, the MLP model was a key component of our project, allowing us to forecast stock prices by leveraging multiple input features. By comparing the MLP's performance with other models like TBATS, LSTM, and exponential smoothing, we aimed to identify the most accurate and reliable forecasting method.

IV. LSTM (Long Short-Term Memory):

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) specifically designed to handle sequential data and capture long-term dependencies. LSTM networks are particularly well-suited for time series forecasting, where patterns and trends evolve over time. In this project, we employed LSTM to predict the stock prices of Tata Steel.

Model Configuration:

- LSTM Units: 128 units, which define the dimensionality of the output space.
- Dropout Layers: A dropout rate of 0.5 to prevent overfitting by randomly setting half of the input units to zero during training.
- Dense Units: 1 unit in the dense layer to output the predicted stock price.
- Optimizer: Adam optimizer for efficient training and faster convergence.
- Epochs: 10 or 20 epochs to train the model over multiple iterations.
- Loss Metric: Mean squared error (MSE) to measure the difference between predicted and actual values.

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_4 (LSTM)               (None, 60, 128)           68096

 dropout_4 (Dropout)         (None, 60, 128)           0

 lstm_5 (LSTM)               (None, 60, 128)           131584

 dropout_5 (Dropout)         (None, 60, 128)           0

 lstm_6 (LSTM)               (None, 60, 128)           131584

 dropout_6 (Dropout)         (None, 60, 128)           0

 lstm_7 (LSTM)               (None, 128)               131584

 dropout_7 (Dropout)         (None, 128)               0

 dense_1 (Dense)             (None, 1)                 129

=================================================================
Total params: 462977 (1.77 MB)
Trainable params: 462977 (1.77 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 9. The summary of LSTM model

Methodology:

- Single Feature (Closing Price): Initially, we trained the LSTM model using only the closing prices of the stock. This approach aimed to understand the model's ability to predict future prices based solely on historical closing prices.
- Multiple Features (Open, High, Low, Close Prices): Subsequently, we expanded the input features to include the opening, high, low, and closing prices. This provided the model with a more comprehensive view of the stock's daily price movements.

Using LSTM for stock price prediction provided valuable insights into the stock's future trends. By leveraging both single and multiple features, the model could achieve high accuracy and robust performance, making it a powerful tool for financial forecasting.

Model Training

The model training process involved preparing the data, selecting appropriate models, tuning hyperparameters, and evaluating performance metrics. Here's a detailed overview of the steps taken to train the various models used in the project:

- Data Collection: We collected three years of Tata Steel stock price data, including open, high, low, close prices, and volume.
- Data Preprocessing: The data was cleaned and preprocessed, including handling missing values, normalizing the features, and creating appropriate input-output pairs for training.
- Differencing: To remove trends and seasonality, differencing was applied to the time series data, where the stock prices were compared with the prices from seven days earlier.

1. Exponential Smoothing:

- This method uses weighted averages of past observations to forecast future values.
- It is simple yet effective for time series data with consistent patterns.
- The model was trained using historical data to predict future stock prices.

2. TBATS (Trigonometric, Box-Cox, ARMA, Trend, and Seasonal Components):

- TBATS is designed to handle complex seasonal patterns in time series data.

- It involves a Box-Cox transformation, trigonometric modeling for seasonality, and ARMA components for autocorrelation.

- The model was trained with hyperparameter tuning to optimize AIC.

3. MLP (Multilayer Perceptron):

- MLP is a feedforward neural network used for regression tasks.

- Features used for training included open, high, low prices, and change percentage.

- The model was trained with a hidden layer size of 50 units, a maximum of 1000 iterations, and a random state for reproducibility.

4. LSTM (Long Short-Term Memory):

- LSTM networks are well-suited for sequential data and capturing long-term dependencies.

- The model configuration included 128 LSTM units, dropout layers of 0.5, and 1 dense unit.

- The model was trained using the Adam optimizer, 20 epochs, and mean squared error as the loss metric.

- Two approaches were used: training with only closing prices and training with multiple features (open, high, low, close prices).

Hyperparameter Tuning

- Hyperparameters for each model were carefully selected and tuned to achieve optimal performance.

- Cross-validation and grid search techniques were employed to identify the best combination of hyperparameters.

Model Evaluation

The performance of each model was evaluated using the following metrics:

1. Directional Accuracy (DA): The percentage of times the predicted direction of stock price movement matched the actual direction. This metric assesses the model's ability to predict the trend correctly.
2. Mean Absolute Percentage Error (MAPE): The average absolute percentage error between forecasted and actual values. MAPE provides a measure of the prediction accuracy of the models.
3. Validation Accuracy: The percentage of correct predictions on the validation set. This metric indicates the model's performance on unseen data.

Comparison and Analysis

After training and evaluating the models, their performances were compared based on the calculated metrics. The impact of differencing on model accuracy was also analyzed by comparing the performance of models with and without differencing.

Forecasting Future Prices

Finally, the best-performing model was used to forecast stock prices for the next 7 days. The forecasted values were plotted to visualize the predicted trend and provide insights into the expected stock price movements.

By following this detailed methodology, the project aims to identify the most effective forecasting method for Tata Steel's stock prices and demonstrate the benefits of differencing in improving model accuracy.

Model Findings

Through extensive model training, validation, and evaluation, we examined the performance of various forecasting methods on Tata Steel's stock prices, both with and without differencing. The results revealed significant differences in the performance of these models under different conditions.

Results

The performance of the different forecasting models was evaluated using multiple metrics, including Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Directional Accuracy. Below are the detailed results for each model.

1. Exponential Smoothing

Fig. 10 shows the graph of training and test values of Exponential Model whereas Fig. 12 shows the same graph but for differenced values. Fig. 11 and 13 shows the 7-day price forecast for stock prices, no differencing and with difference respectively.

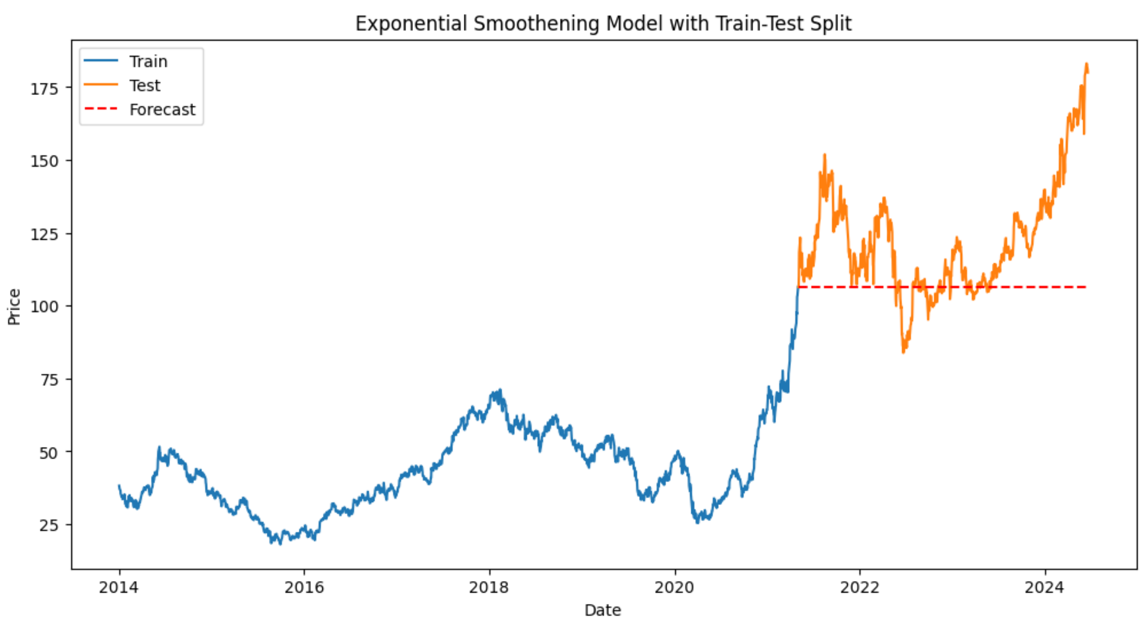

Figure 10. Exponential Smoothing model graph

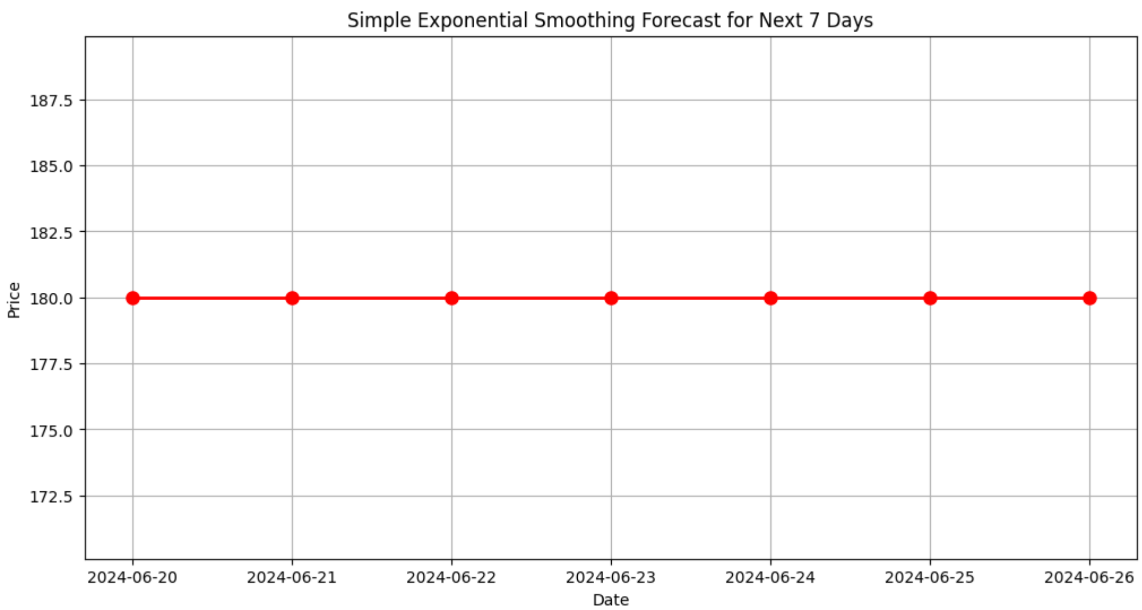

Figure 11. Price prediction for 7 days

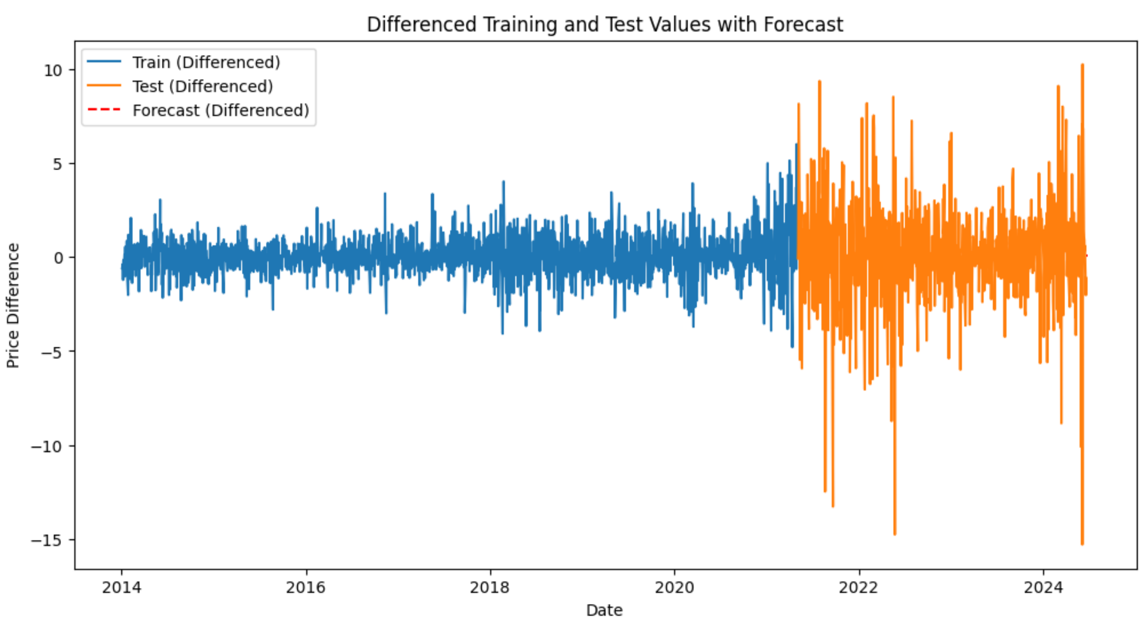

Figure 12. Exponential smoothing for differenced values

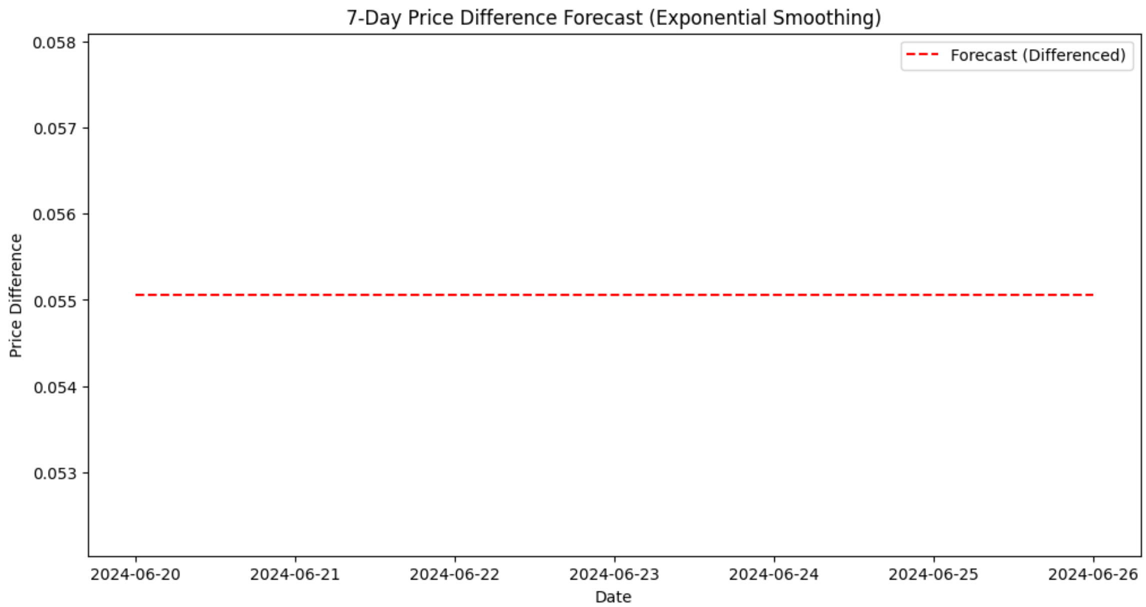

Figure 13. Seven (7) days prediction of differenced values.

Exponential Smoothing demonstrated a straightforward approach but struggled with capturing more complex patterns in the stock prices.

2. TBATS (Trigonometric, Box-Cox, ARMA, Trend, and Seasonal Components)

In Fig. 14, we can see the graph for 7-days of price prediction using TBATS model. The values are far from accurate.

Fig. 15 shows the differenced price prediction for next 7-days. Values appear to be constant.

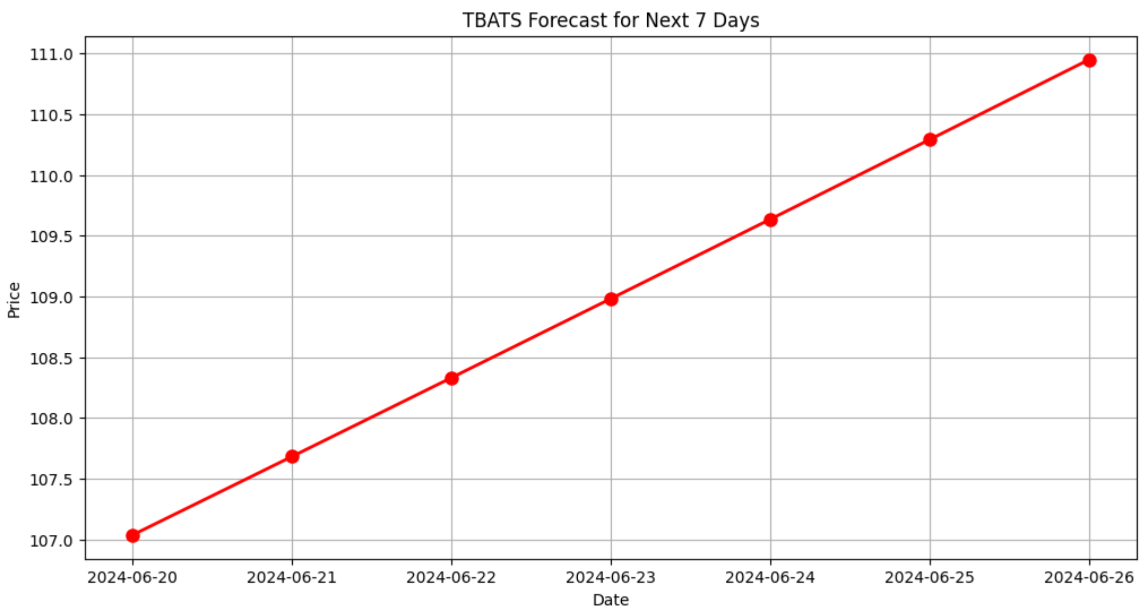

Figure 14. TBATS model predicted price for next 7 days.

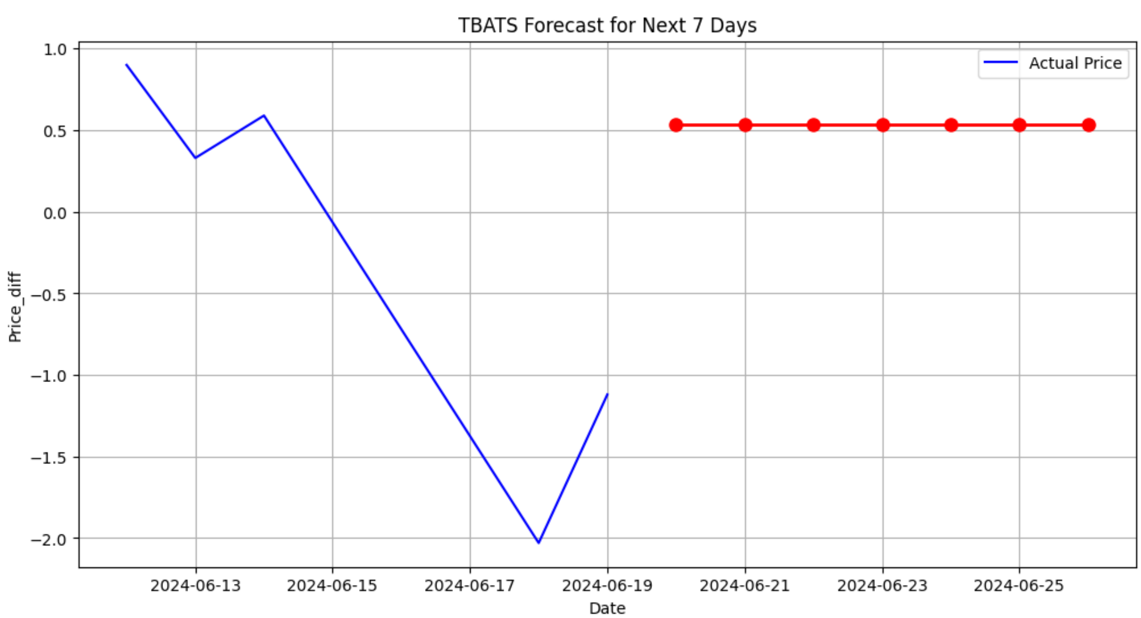

Figure 15. Forecasted differenced pricing shown by the red line.

TBATS excelled in handling multiple seasonal periods, especially after applying differencing, significantly improving its directional accuracy.

3. MLP (Multilayer Perceptron)

Fig. 16 shows the graph of the test dataset of Tata Steel stock price. Here, the actual price and predicted prices are compared. Fig. 17 shows the next 7-day predictions.

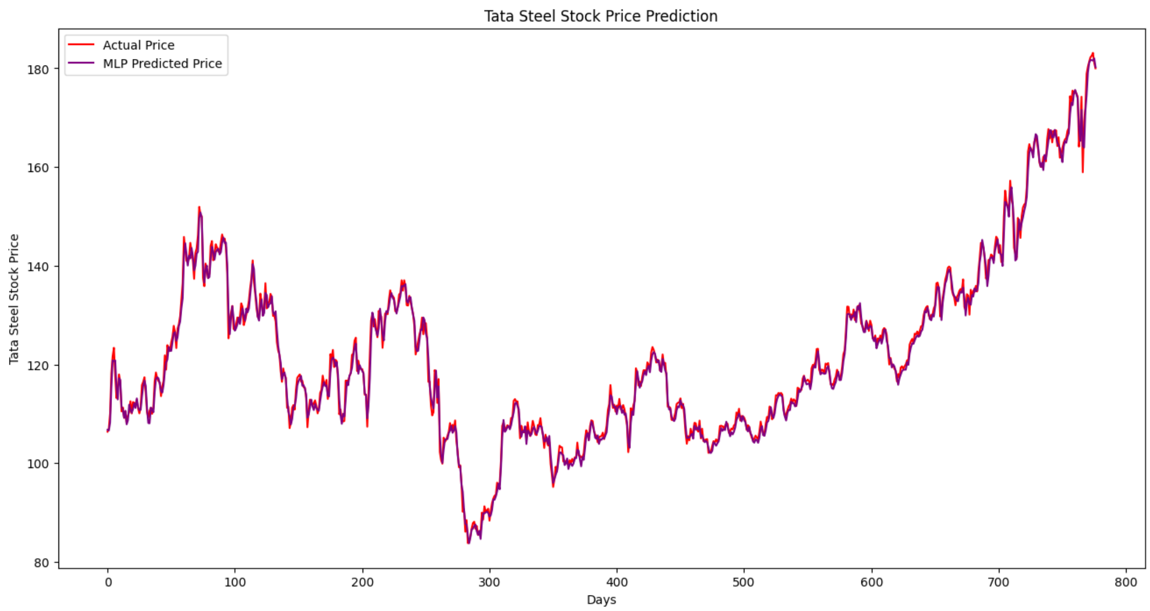

Figure 16. Test dataset graph of TSL using MLP method

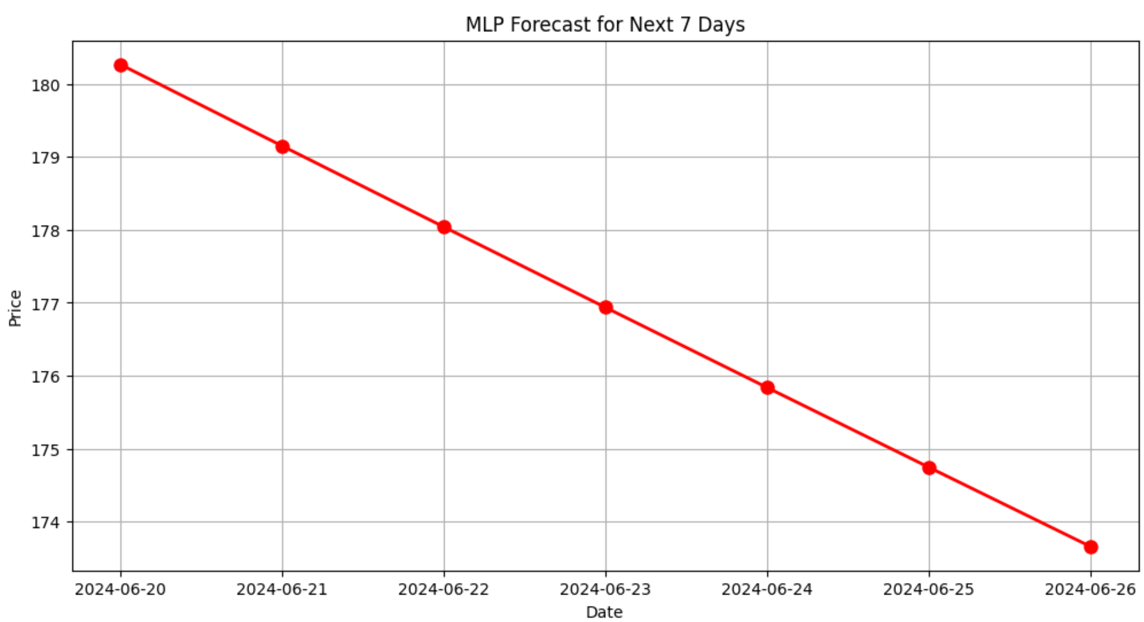

Figure 17. Forecasted values using MLP

Fig. 18 shows the graph of the test dataset of stock price with differenced values. Here, the actual price and predicted prices are compared. Fig. 19 shows the next 7-day predictions.

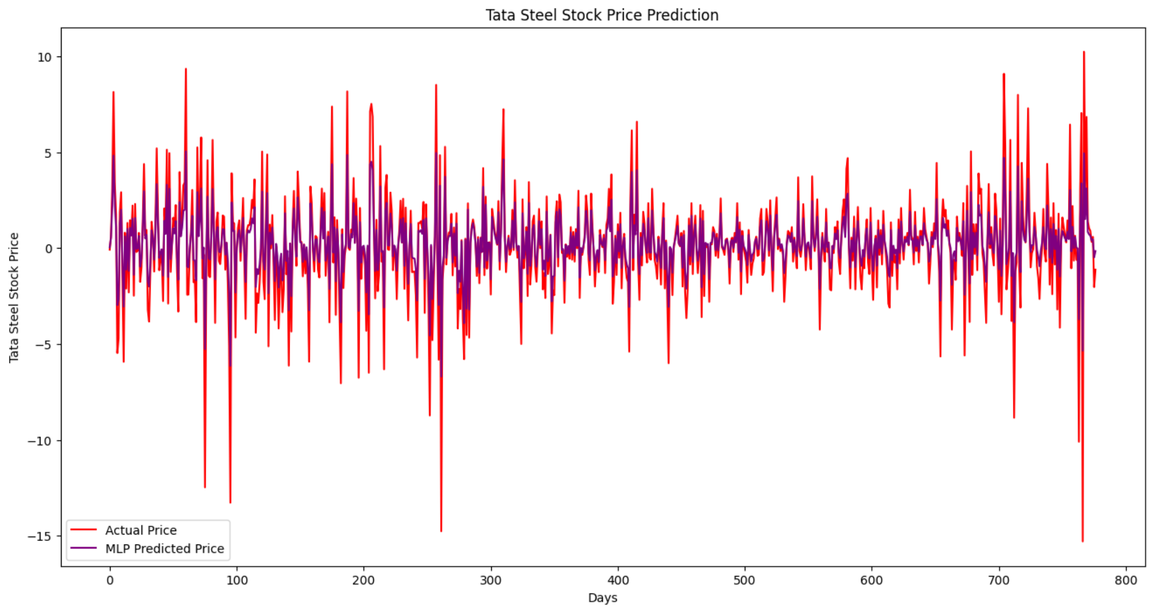

Figure 18. Test data values graph for differenced prices using MLP

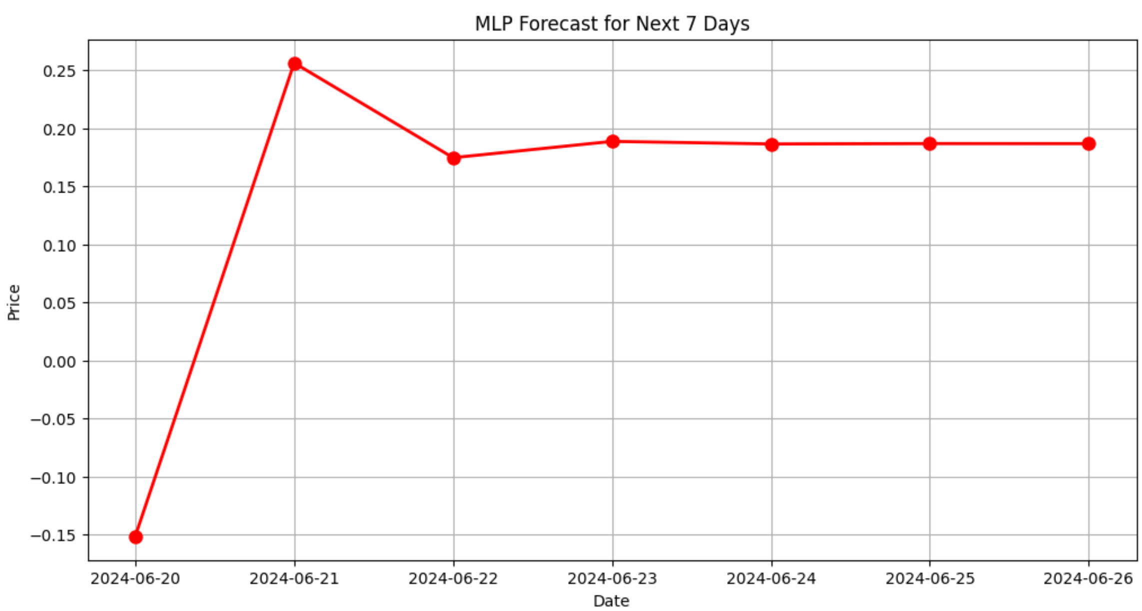

Figure 19. Predicted values for differenced prices using MLP.

MLP showed potential in forecasting stock prices with a high degree of accuracy in non-differenced data but faced issues when differencing was applied, leading to highly inflated MAPE values.

4. LSTM (Long Short-Term Memory)

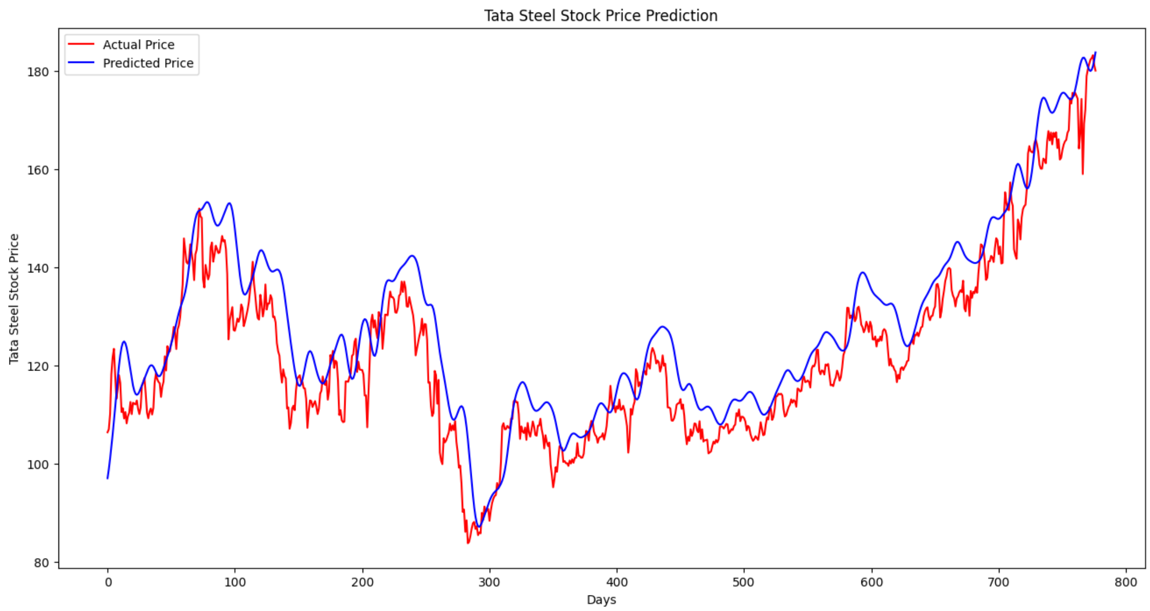

Figure 20. Graph of test set of Tata Steel stock price

Here, Fig. 20 shows the graph comparing actual test data prices to that predicted by LSTM model. Here only the 'close price' is trained upon. Fig. 21 shows the model training loss decreasing with increasing epochs.

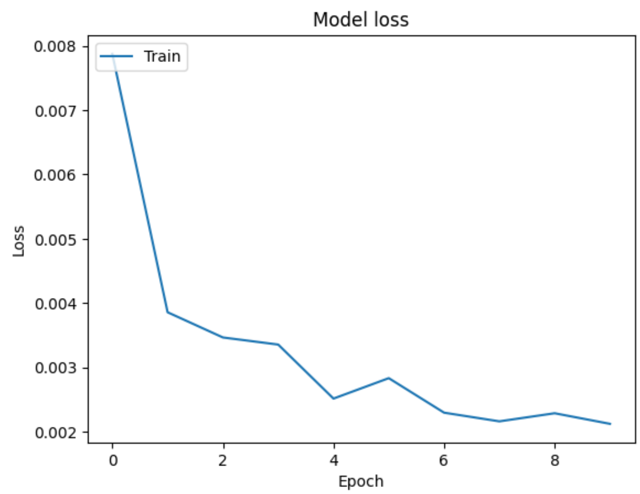

Figure 21. Training loss of LSTM

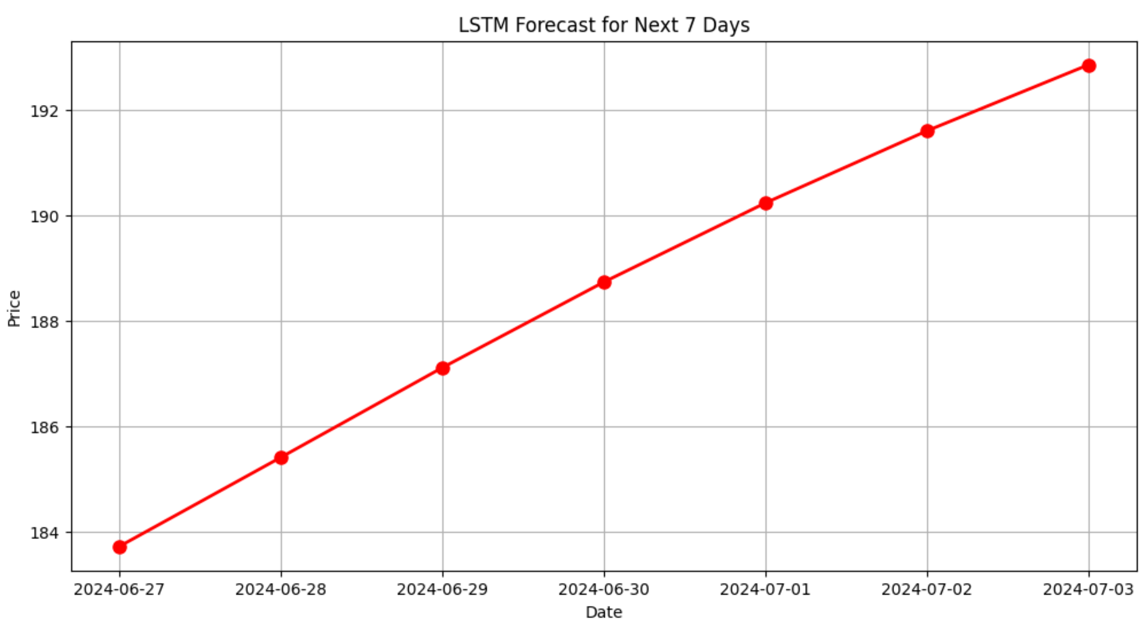

Figure 22. Forecasted stock prices

Fig. 22 depicts an almost straight line of increasing order. This is the forecasted price of stock for the next 7-days.

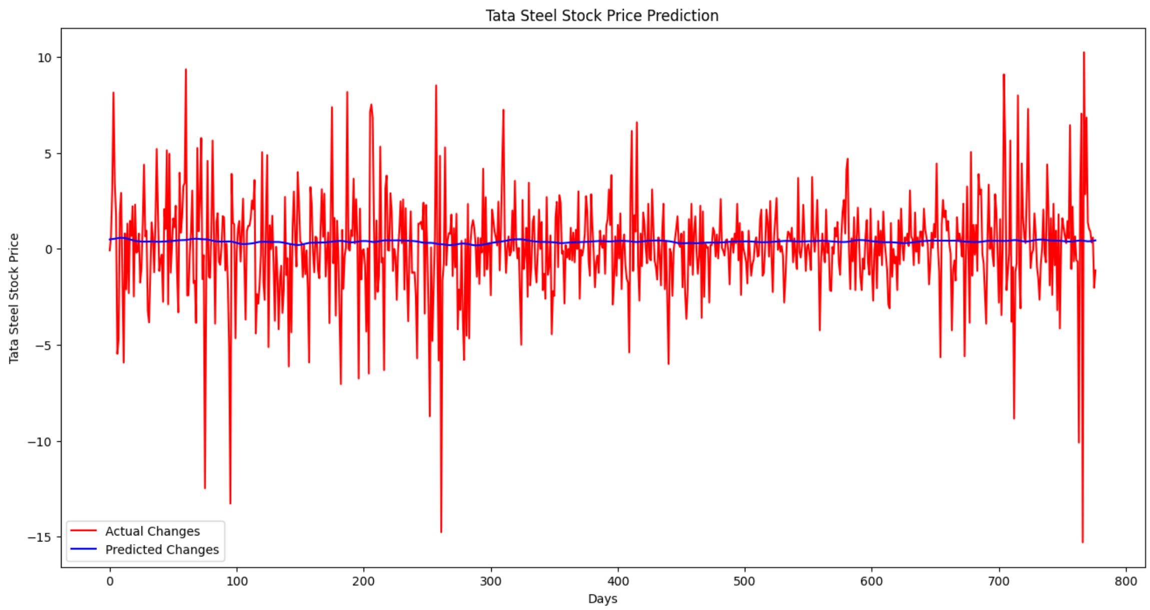

Figure 23. LSTM testing data graph for differenced values.

When differencing was applied, the test data and predicted values were compared. The graph plotted is shown in Fig. 23

Fig. 24 depicts the next 7 days of price forecasted for differenced values.

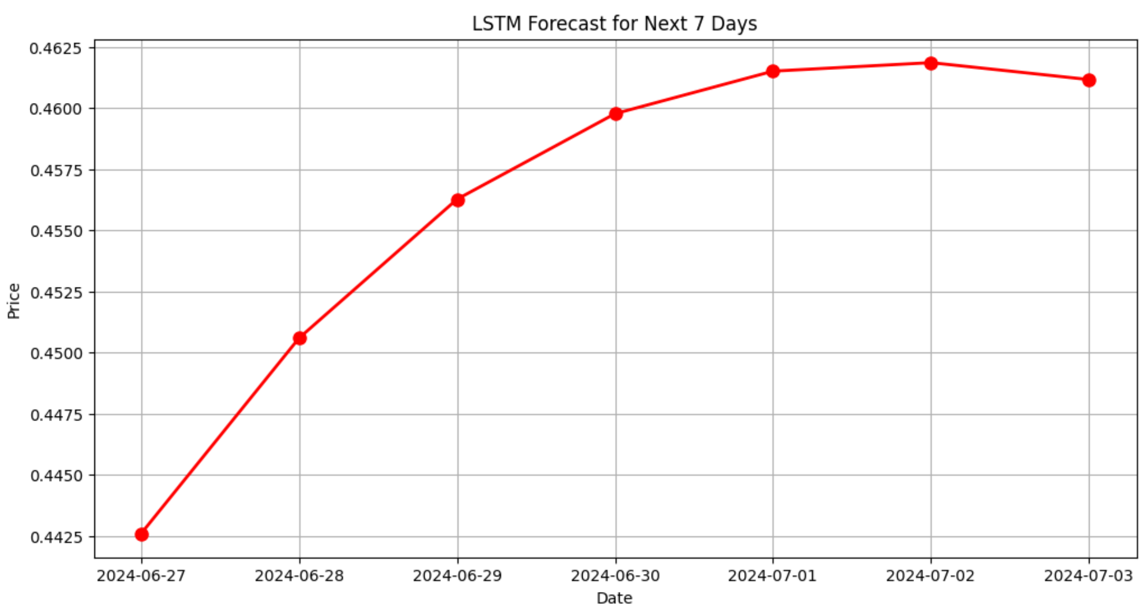

Figure 24. Differenced price prediction by LSTM

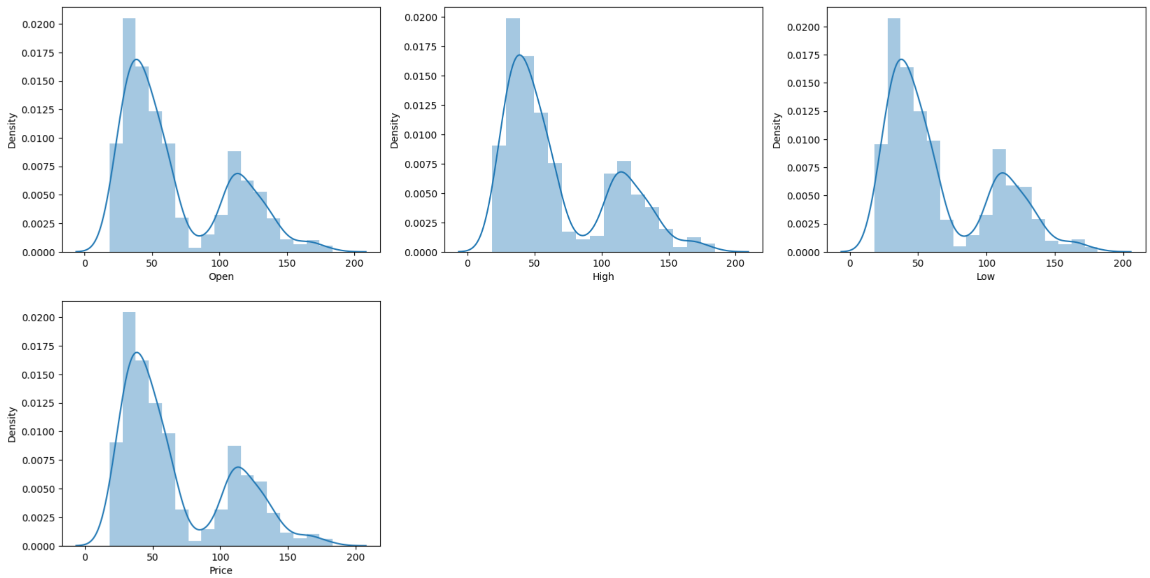

Figure 25. Graphs of Open, High, Low, Price of stock data.

Fig. 25 comprises of some graphs for each of Open price, High price, Low price and Close price respectively, starting from top left to right and then bottom. Fig. 26 shows the graph of actual test data compared to the predicted price when all these four labels were used to train the data.

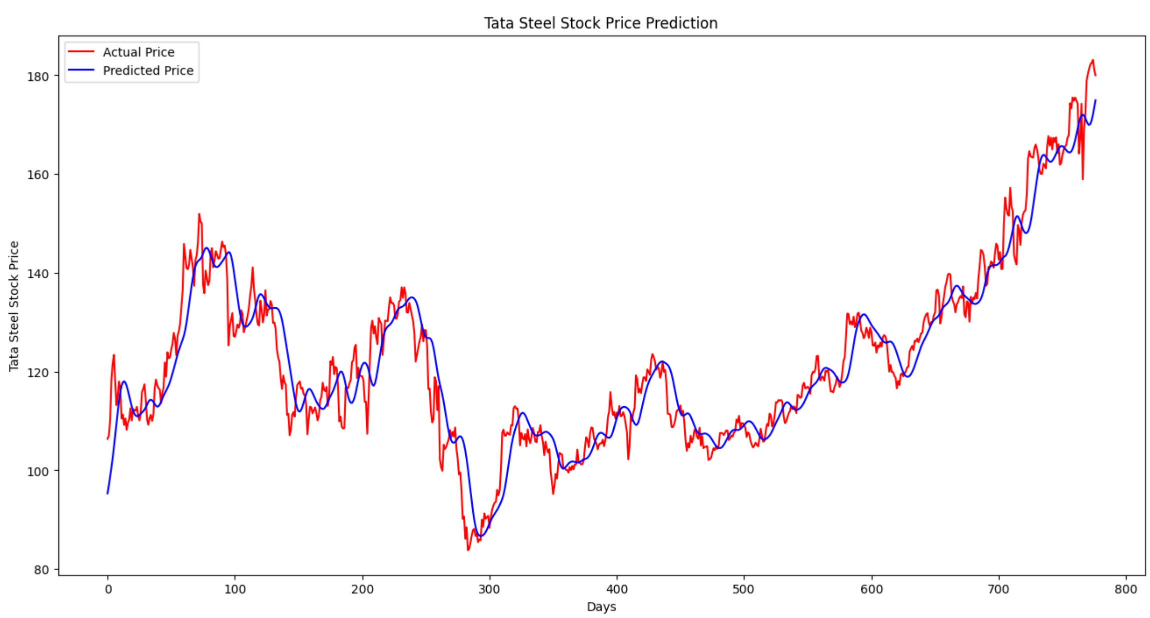

Figure 26. Test set graph of TSL stock data with 4 inputs.

The LSTM model with four inputs of close price, high, low and open price, gave the following results.

- Directional Accuracy: 51.3260242852588
- Mean Absolute Percentage Error (MAPE): 16.07%
- RMSE: 5.6073075642782255

LSTM demonstrated robust performance, especially in capturing long-term dependencies and trends in stock prices. However, the application of differencing led to extreme values in MAPE, indicating potential issues in handling differenced data.

The table below summarizes the performance metrics of the forecasting models:

	Metrics	TBATS(7 Days)	MLP	LSTM	Exponential Smoothening
without difference	MAPE	3.9%	0.84%	5.53%	13.27%
	Mean Absolute Error (MAE)	72.5	1.03	6.59	17.88
	Mean Squared Error (MSE)	5264.93	1.85	68.23	-
	RMSE	72.55	1.363	8.26	24.49
	Directional Accuracy	33.33%	51.22%	55.15%	50.45%
	Median Absolute Error	73.5	0.85	3.49	-
with difference	MAPE	70%	855×10^{10}%	227×10^{10}%	95.18%
	Mean Absolute Error (MAE)	0.886	0.812	1.84	1.03
	Mean Squared Error (MSE)	1.49	1.59	7.12	-
	RMSE	1.22	1.26	2.66	1.17
	Directional Accuracy	100%	68.49%	73.06%	71.43%
	Median Absolute Error	0.5	0.537	3.111	-

Table 1. Comparison of metrics of different models.

Analysis

Without Differencing

MAPE: MLP demonstrated the lowest MAPE, indicating the highest prediction accuracy, followed by LSTM. TBATS and Exponential Smoothing had significantly higher MAPE values.

Mean Absolute Error and Median Absolute Error: MLP again showed superior performance with the lowest errors. LSTM performed moderately well, while TBATS and Exponential Smoothing showed higher errors.

RMSE: MLP had the lowest RMSE, suggesting better overall accuracy, followed by LSTM. TBATS and Exponential Smoothing lags behind.

Directional Accuracy: LSTM achieved the highest directional accuracy, closely followed by MLP. TBATS and Exponential Smoothing showed relatively lower directional accuracy.

With Differencing

MAPE: All models, except TBATS, showed significantly higher MAPE values, indicating poor performance. This could be attributed to the inappropriate scaling or data handling during differencing.

Mean Absolute Error and Median Absolute Error: TBATS performed exceptionally well with the lowest errors, indicating that differencing greatly benefited this model. MLP and LSTM also showed improvements but to a lesser extent.

RMSE: TBATS had the lowest RMSE, confirming its superior performance with differencing. MLP and LSTM showed moderate improvements.

Directional Accuracy: TBATS achieved perfect directional accuracy, followed by MLP and LSTM, which also showed significant improvements over their non-differenced counterparts.

Conclusion

The comparison of forecasting models with and without differencing revealed several key insights:

Model Performance: Without differencing, MLP and LSTM performed significantly better than TBATS and Exponential Smoothing, with lower MAPE, errors, and higher directional accuracy. Differencing, however, markedly improved TBATS, making it the best-performing model in terms of error metrics and directional accuracy.

Impact of Differencing: Differencing had a substantial positive impact on TBATS, improving its performance metrics considerably. It also benefited MLP and LSTM but to a lesser extent. Exponential Smoothing did not show significant benefits from differencing.

Recommendation: For forecasting Tata Steel's stock prices, MLP and LSTM are recommended for use without differencing due to their superior performance. However, TBATS with differencing is highly effective, achieving perfect directional accuracy and low error metrics.

Overall, this project highlights the importance of selecting appropriate forecasting methods and preprocessing techniques, such as differencing, to enhance the accuracy and reliability of stock price predictions.

All the codes are saved and can be found in this repository link given below: https://github.com/shrey141102/tata-steel-internship