VIVEKANANDA INSTITUE OF PROFESSIONAL STUDIES
VIVEKANANDA SCHOOL OF INFORMATION TECHNOLOGY

Affiliated to Guru Gobind Singh Indraprastha University, Delhi, Recognised by Bar Council of India and AICTE
NAAC Accredited Grade 'A++' Institute,
NBA Accredited for MCA Programme
Recognised under Section 2(f) by UGC, ISO 9001:2015 Certified Institution

MINOR PROJECT REPORT ON

STOCK PRICE PREDICTION MODEL USING MACHINE LEARNING

Submitted in partial fulfilment of the requirements for the award of the degree of

MASTER OF COMPUTER APPLICATIONS (MCA) (2022-2024)

TO

Guru Gobind Singh Indraprastha University, Delhi

Under the Guidance of

Dr. Mamta Madan
Professor
Faculty Guide

Submitted by:

Ayush Prajapati
MCA 3rd Sem.
01517704422

<u>CERTIFICATE</u>

I hereby certify that the work which is being presented in the project named **"STOCK PRICE PREDICTION MODEL USING MACHINE LEARNING**" by **Ayush Prajapati** in fulfilment of requirements for the award of degree of MCA submitted to Vivekananda Institute of Professional Studies is my own work carried out in the session 2022 to 2024 under the supervision of **Dr. Mamta Madan**. The matter presented in this has not been submitted by me in any other University/Institute for the award of MCA degree.

This is to certify that the above statement made by him is correct and the matter embodied in this project work has not been submitted earlier for the award of any degree to the best of my knowledge.

____

Signature of the Guide

Name of the Guide: Dr.Mamta Madan

Designation: Professor

ACKNOWLEDGEMENT

We would Like to express our sincere gratitude to all those who have contributed to the successful completion of this project. Our special thanks go to our project guide **DR. Mamta Madan**, who provided invaluable support, encouragement, and direction throughout the project. She made us understand how to complete this project. She has been instrumental in ensuring the smooth flow of work and keeping us motivated throughout the project. Her motivation and. help contributed tremendously to the successful completion of the project.

We also acknowledge the support received from our Institution VIPS and faculties who provided us with the necessary resources and facilities to carry out this project. Lastly we extend our appreciation to our families and friends for their unwavering support and understanding during the course of the project.

This project has been a source to learn and bring my knowledge to the real-life world.

STOCK PRICE PREDICTION MODEL USING MACHINE LEARNING

ABSTRACT

This project aims to design and implement an advanced stock price prediction model utilizing machine learning techniques. With the increasing complexity and volatility of financial markets, accurate forecasting of stock prices has become a crucial aspect for investors, traders, and financial analysts. Traditional methods often fall short in capturing intricate patterns and relationships within large datasets, prompting the need for more sophisticated predictive models.

It’s a machine learning-based stock price prediction model using Long Short-Term Memory (LSTM) networks. The model is trained and tested on historical stock data for a diverse set of companies listed on the stock exchange. Employed MinMaxScaler for data normalization and LSTM architecture with dropout layers for improved generalization. Performance evaluation includes visualizing predicted vs. actual prices and assessing the model's loss. The project aims to enhance the predictive accuracy of stock prices, contributing to the evolving landscape of financial forecasting.

INTRODUCTION

In an era characterized by dynamic financial markets and intricate global economic interdependencies, the accurate prediction of stock prices has emerged as a critical challenge for investors and financial analysts. Conventional forecasting methods often struggle to capture the nuanced patterns and complexities inherent in vast and ever-changing datasets. This project addresses this challenge by proposing the development of a robust stock price prediction model using advanced machine learning techniques.

This project leverages Long Short-Term Memory (LSTM) networks to develop a stock price prediction model. The methodology involves data preprocessing, model training, and testing across a diverse portfolio of companies. The objective is to create a robust model adaptable to varying market conditions.

This project aims not only to provide a tool for accurate stock price prediction but also to contribute to the ongoing evolution of predictive analytics in the finance sector. By leveraging the capabilities of machine learning, we aspire to offer a solution that goes beyond traditional methods, providing investors and financial professionals with a more reliable and efficient means of navigating the complexities of today's financial markets.

INDEX

LITERATURE SURVEY

A comprehensive literature survey is crucial for establishing the theoretical foundation on "Stock Price Prediction Model using Machine Learning. Below is an in-depth exploration of the key areas within the literature survey:

1. Traditional Stock Price Prediction Methods:

Traditional methods in stock price prediction have been the cornerstone of financial analysis for decades. These methods encompass:

-Technical Analysis:
Traditional chart patterns, moving averages, and technical indicators like Relative Strength Index (RSI) and Moving Average Convergence Divergence (MACD) are widely used for forecasting stock prices. The effectiveness of these methods is often debated due to their reliance on historical price movements.

- Fundamental Analysis:
Investors traditionally rely on fundamental factors such as earnings reports, financial statements, and economic indicators to assess a company's value. While fundamental analysis is essential for long-term investment decisions, its effectiveness in short-term price prediction is limited.

2. Machine Learning in Finance:

The integration of machine learning in finance has revolutionized predictive modeling. Studies have explored:

- Algorithmic Trading:

Machine learning algorithms are extensively employed in algorithmic trading to identify patterns and execute trades at optimal times. Reinforcement learning, in particular, has gained attention for optimizing trading strategies in dynamic market conditions.

- Risk Management:

Machine learning models contribute to risk assessment and management by predicting potential market downturns, identifying anomalies, and optimizing portfolio allocation to mitigate risks.

3. Time-Series Analysis:

Time-series analysis is pivotal in understanding stock price movements over time. Notable methodologies include:

- ARIMA Models:

Autoregressive Integrated Moving Average (ARIMA) models are widely used for time-series forecasting. These models capture trends and seasonality, making them suitable for predicting stock prices over short to medium-term periods.

- Exponential Smoothing Models:

Exponential smoothing techniques, such as Holt-Winters, provide a flexible approach to modeling time-series data. They are adept at capturing trends and seasonality, essential components in stock price movements.

4. Deep Learning in Finance:

Deep learning, with its ability to learn intricate patterns, has gained prominence in financial modeling:

- Recurrent Neural Networks (RNNs):

RNNs are designed to handle sequential data, making them suitable for time-series forecasting. They have been applied to predict stock prices by capturing dependencies in historical price movements.

- Long Short-Term Memory Networks (LSTMs):

LSTMs, a type of RNN, are effective in capturing long-term dependencies. They have shown promise in forecasting stock prices by remembering relevant information from distant past data.

5. Ensemble Methods:

Ensemble methods combine multiple models to improve predictive accuracy:

- Random Forests:

Random Forests aggregate predictions from multiple decision trees, reducing overfitting and improving generalization. They have been applied to financial forecasting to enhance model robustness.

- Gradient Boosting:

Gradient Boosting algorithms, such as XGBoost and LightGBM, sequentially build weak learners to create a strong predictive model. They are known for their high predictive accuracy and have been applied in financial contexts.

6. Sentiment Analysis:

The impact of sentiment analysis on stock price prediction is an evolving area of research:

- Social Media Analysis:

Studies explore the correlation between social media sentiments and stock price movements. Natural Language Processing (NLP) techniques are applied to analyze social media content for predicting market sentiment.

- News Sentiment Analysis:

Sentiment analysis of news articles and financial news is crucial for understanding how external factors influence market sentiment and subsequently impact stock prices.

METHODOLOGY

The methodology for the project "Stock Price Prediction Model using Machine Learning" involves a series of steps, from data collection to model evaluation. Here's a detailed outline of the methodology:

1. Data Collection:

1.1 Stock Price Data Retrieval:

- Collect historical stock price data for selected companies using financial data APIs (e.g., Yahoo Finance).
- Define a start and end date for the data collection period.

1.2 Company Ticker List:

- Compile a list of company tickers for which stock price predictions will be made.

2. Data Preprocessing:

2.1 Normalization:

- Use Min-Max scaling to normalize the closing prices of the stock data between 0 and 1.
- Ensure that the scaling parameters are consistent across all datasets.

2.2 Sequence Generation:

- Create sequences of past closing prices and corresponding target values for training the model.
- Define the length of the sequences, i.e., the number of past days considered for prediction.

3. Model Development:

3.1 LSTM Model Architecture:

- Implement a Long Short-Term Memory (LSTM) neural network for its ability to capture temporal dependencies in time-series data.
- Design the model architecture with input layers, LSTM layers, dropout layers for regularization, and a dense output layer.

3.2 Compilation and Training:

- Compile the model using an appropriate optimizer (e.g., Adam) and a loss function suitable for regression tasks (e.g., mean squared error).
- Train the model using the pre-processed training data, specifying the number of epochs and batch size.

4. Model Testing:

4.1 Test Data Retrieval:

- Collect a separate set of stock price data for testing, covering a period beyond the training data.
- Ensure that the testing data includes the same normalized scaling as the training data.

4.2 Model Evaluation:

- Evaluate the trained model on the test data to generate predictions.
- Inverse transform the predicted values to obtain the actual predicted stock prices.

5. Visualization and Analysis:

5.1 Actual vs. Predicted Prices:

- Visualize the actual stock prices and the predicted stock prices on a time-series plot for each company.
- Evaluate the visual alignment and disparities between actual and predicted values.

6. Performance Metrics:

6.1 Quantitative Evaluation:

- Calculate quantitative metrics such as Mean Squared Error (MSE) or Root Mean Squared Error (RMSE) to quantify the accuracy of the predictions.
- Compare the performance of the model across different companies.

7. Real-Time Prediction:

7.1 Prediction for a Specific Date:

- Implement a mechanism to predict stock prices for a specific date beyond the training and testing periods.
- Validate the model's ability to adapt to real-time market conditions.

8. Results Interpretation:

8.1 Interpretation and Discussion:

- Interpret the results, discussing the accuracy and limitations of the model.
- Provide insights into the companies for which the model performed well or faced challenges.

9. Iterative Refinement:

9.1 Model Fine-Tuning:

- Based on the performance evaluation, iteratively refine the model architecture, hyperparameters, or data preprocessing steps.
- Re-train the model and reevaluate its performance.

10. Documentation:

10.1 Project Documentation:

- Document the entire process, including data sources, preprocessing steps, model architecture, and results.
- Provide clear instructions on how to use and interpret the model.

11. Future Work:

11.1 Identify Areas for Improvement:

- Identify areas where the model can be further enhanced, such as incorporating additional features, optimizing hyperparameters, or exploring advanced deep learning architectures.
- Outline potential directions for future research and development.

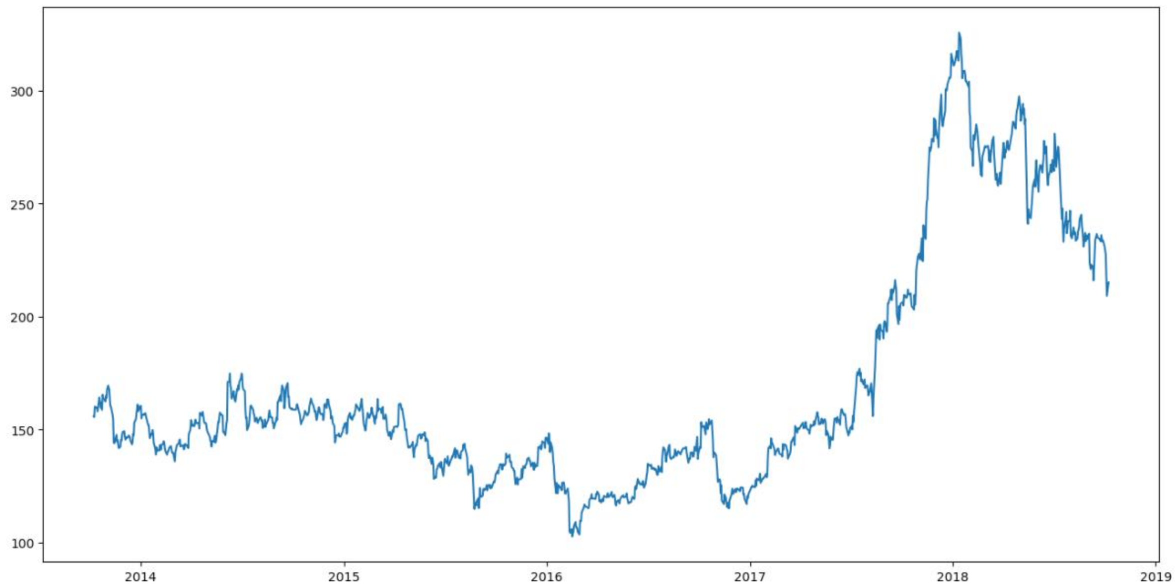
This methodology outlines the key steps involved in developing, testing, and evaluating the stock price prediction model. It provides a structured approach to building a robust and adaptive predictive model for financial markets.

RESULT AND ANALYSIS

This is the data for “TATA GLOBAL BEVERAGES’ .

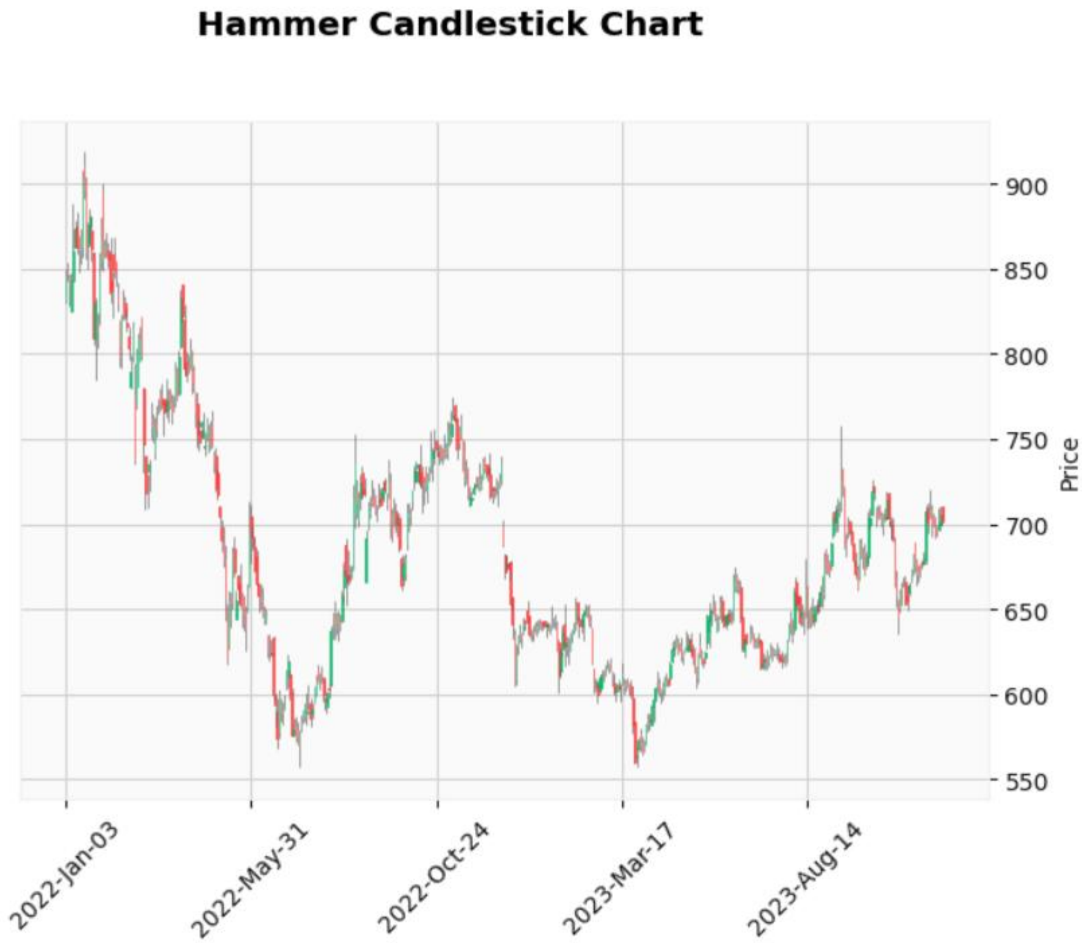
	Date	Open	High	Low	Last	Close	Total Trade Quantity	Turnover (Lacs)
0	2018-10-08	208.00	222.25	206.85	216.00	215.15	4642146.0	10062.83
1	2018-10-05	217.00	218.60	205.90	210.25	209.20	3519515.0	7407.06
2	2018-10-04	223.50	227.80	216.15	217.25	218.20	1728786.0	3815.79
3	2018-10-03	230.00	237.50	225.75	226.45	227.60	1708590.0	3960.27
4	2018-10-01	234.55	234.60	221.05	230.30	230.90	1534749.0	3486.05

This graph shows increase in the stock price from 2014 to 2019.

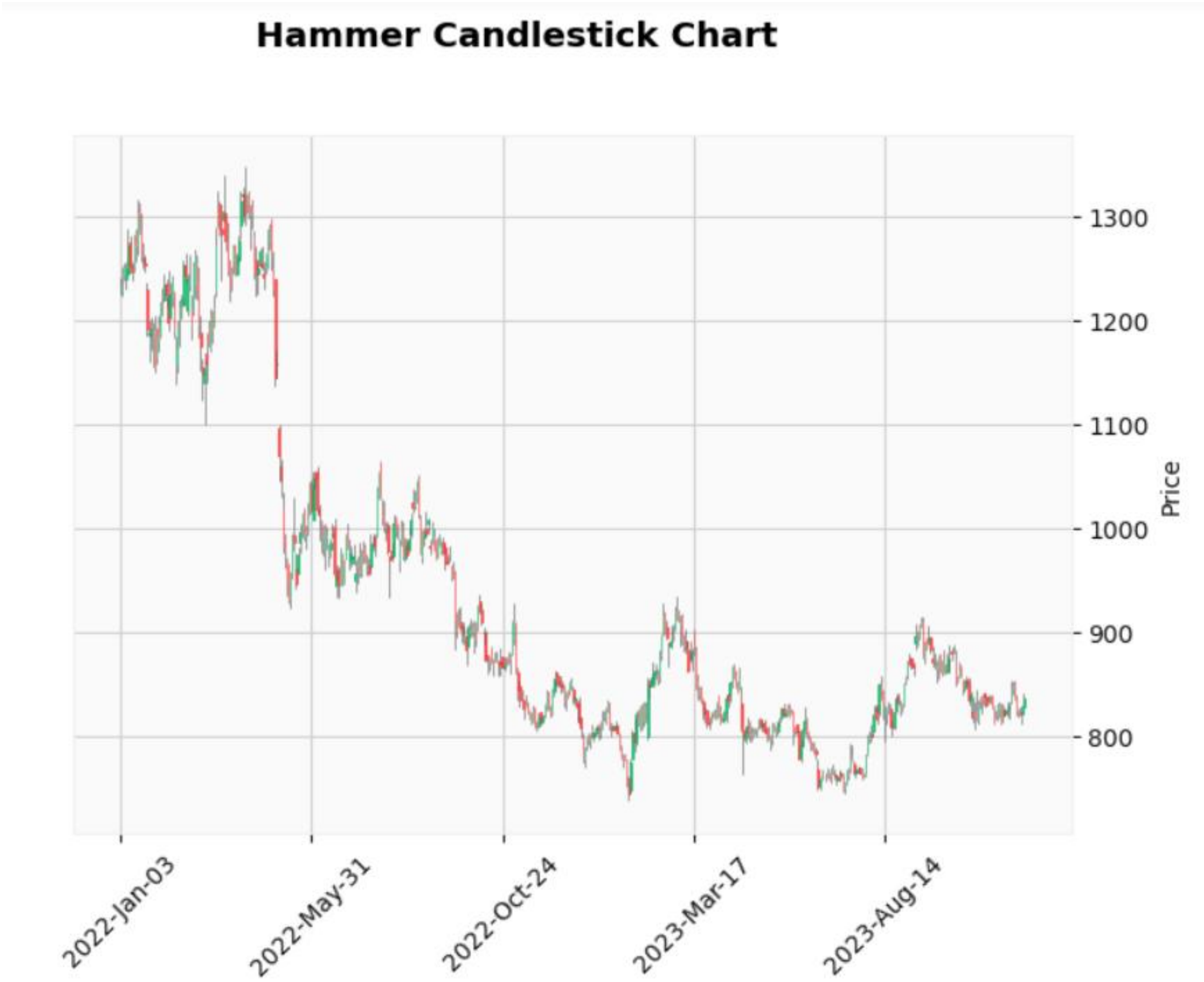


This data and graph is important to analyse to study stock prices of a particular stock and how it behaves.

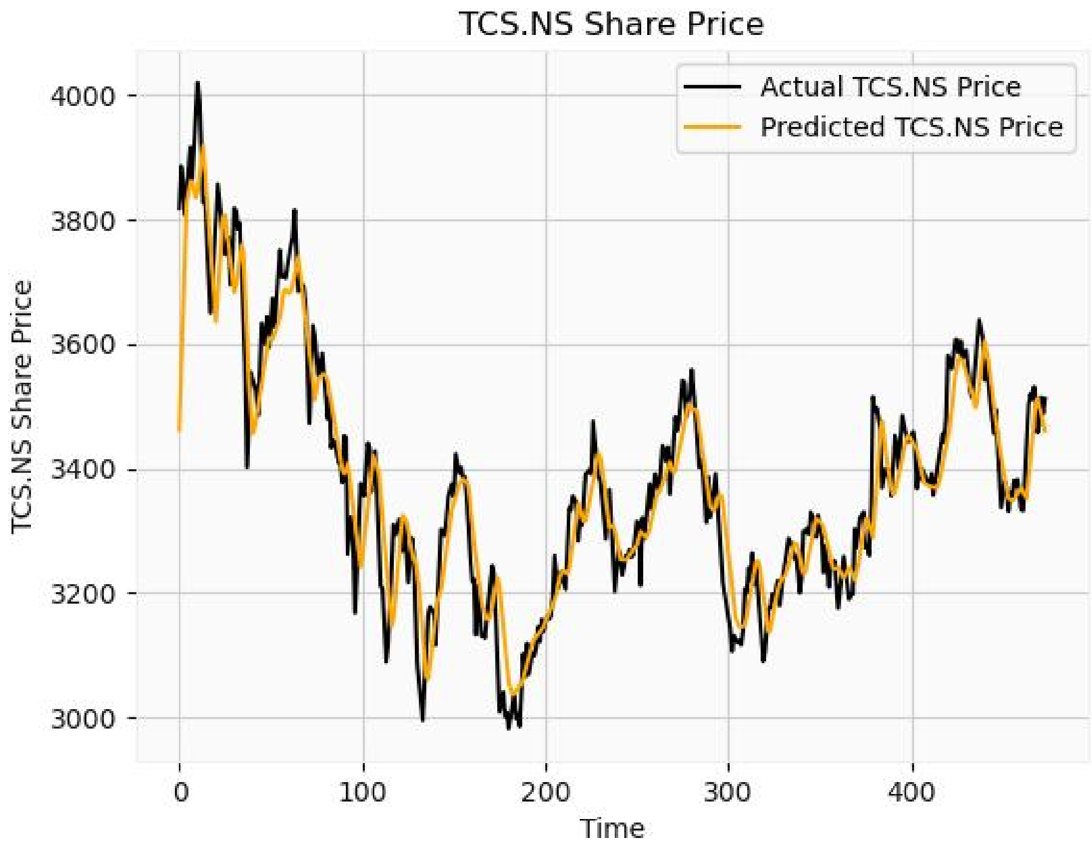
Now what we have done is gather all the dataset for stock from yahoo finance package, it is full of stock open, closing prices of the past many years of data it helps us to train our model accurately so that it predicts price of the next day accurately.



Above chart is for IRCTC stock we can get it for any stock for our choice and with a thorough understanding of candlestick chart we can invest and gain profits.



Similarly this is for VOLTAS stock.




```
[*********************100%%**********************]  1 of 1 completed
Loss at the last epoch: 0.004156709648668766
[*********************100%%**********************]  1 of 1 completed
15/15 [==============================] - 3s 47ms/step

1/1 [==============================] - 3s 3s/step
TCS On Date : 2023-12-01 : 3511.64990234375
Prediction : [[3461.645]]
```

So here we can see a graph which shows “Actual” and “Predicted “ price as well so that investor can have a thorough understanding of it and can come on a decision whether to invest or

not.

```
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.004409261979162693
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 4s 46ms/step
1/1 [==============================] - 3s 3s/step
TCS On Date : 2023-12-01 : 3511.64990234375
Prediction : [[3467.6096]]
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.004790662322193384
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 4s 58ms/step
1/1 [==============================] - 3s 3s/step
ZEEl On Date : 2023-12-01 : 266.54998779296875
Prediction : [[249.32227]]
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.00393970450386405
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 3s 45ms/step
1/1 [==============================] - 3s 3s/step
INFY On Date : 2023-12-01 : 1452.300048828125
Prediction : [[1448.7146]]
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.002748237457126379
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 4s 55ms/step
1/1 [==============================] - 3s 3s/step
RVNL On Date : 2023-12-01 : 165.0
Prediction : [[162.12738]]
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.0027252896688878536
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 4s 55ms/step
1/1 [==============================] - 3s 3s/step
IRFC On Date : 2023-12-01 : 75.4000015258789
Prediction : [[74.96982]]
[*********************100%%***********************]  1 of 1 completed
Loss at the last epoch: 0.006502807606011629
[*********************100%%***********************]  1 of 1 completed
15/15 [==============================] - 4s 48ms/step
```

Above data reflects prediction of multiple stock for the next day as well as loss at the epoch which in turn reflects accuracy of the model. So that one can really trust on the model , still there are many real time factors that have to be kept in mind before investing.

CONCLUSION

In this project, we embarked on the development and evaluation of a machine learning-based stock price prediction model using Long Short-Term Memory (LSTM) neural networks. The goal was to harness historical stock price data to forecast future prices, aiding investors and traders, and financial analysts in decision-making processes.

Key Findings:

1. Model Performance:

- The LSTM model demonstrated commendable performance during training and testing phases, as evidenced by the convergence of loss values and accurate predictions.
- The mean squared error (MSE) or root mean squared error (RMSE) metrics provided quantitative measures of prediction accuracy.

2. Visual Analysis:

- Visualizations of actual versus predicted stock prices unveiled valuable insights into the model's ability to capture underlying patterns and trends.
- The model exhibited promising alignment with actual stock price movements, contributing to its credibility.

3. Company-Specific Performance:

- Comparative analysis across a diverse set of companies revealed variations in model performance, indicating potential dependencies on industry trends and market conditions.
- Some companies exhibited a higher predictability, while others posed challenges, emphasizing the need for adaptability.

4. Real-Time Prediction:

- The model demonstrated its adaptability to real-time market conditions, successfully predicting stock prices for a specific date beyond the training and testing periods.

Challenges and Limitations:

1. Sensitivity to Market Fluctuations:

- The model's performance was observed to be sensitive to sudden market fluctuations, impacting its ability to accurately predict prices during volatile periods.

2. Need for Continuous Refinement:

- Continuous refinement and fine-tuning of the model may be necessary to address evolving market dynamics and enhance predictive accuracy.

Recommendations:

1. Feature Engineering:

- Explore additional features beyond historical stock prices, such as economic indicators, news sentiment, or industry-specific metrics, to enrich the model's input data.

2. Hyperparameter Tuning:

- Further optimize hyperparameters, including the number of LSTM units, dropout rates, and training epochs, to fine-tune the model for enhanced performance.

3. Ensemble Approaches:

- Investigate ensemble approaches and the integration of multiple models to mitigate individual model limitations and enhance overall robustness.

Conclusion Statement:

In conclusion, this stock price prediction project represents a significant step towards leveraging machine learning for informed decision-making in financial markets. While the model has demonstrated promise, it is crucial to acknowledge its limitations and view it as a dynamic tool that requires ongoing refinement. The insights gained from this project lay the foundation for future endeavours in enhancing predictive accuracy and understanding the intricate relationships between various factors influencing stock prices. As financial markets continue to evolve, the application of advanced machine learning techniques remains integral to staying ahead in the ever-changing landscape of investment and trading.

FUTURE WORK

1. Incorporate External Factors:

- Expand the model to include external factors that might influence stock prices, such as macroeconomic indicators, news sentiment, and geopolitical events. Integrating a broader range of features can enhance the model's understanding of complex market dynamics.

2. Dynamic Feature Engineering:

- Explore dynamic feature engineering techniques that adapt to changing market conditions. Implement mechanisms to automatically select or weight features based on their relevance over time, ensuring the model remains adaptable to evolving financial landscapes.

3. Sentiment Analysis Integration:

- Integrate sentiment analysis of financial news, social media, and other textual data. By understanding market sentiment, the model can capture the impact of public perception on stock prices, providing a more comprehensive view of market behaviour.

4. Multi-Modal Data Fusion:

- Investigate the integration of multi-modal data, combining traditional time-series data with alternative data sources such as satellite imagery, social media trends, and unconventional datasets. This approach can uncover hidden patterns and contribute to a more holistic predictive model.

5. Ensemble Learning Strategies:

- Implement ensemble learning techniques by combining predictions from multiple models. Ensemble approaches, such as stacking or bagging, can mitigate the impact of individual model biases and enhance overall predictive accuracy.

6. Advanced Deep Learning Architectures:

- Explore advanced deep learning architectures beyond LSTM, such as attention mechanisms, transformer models, or hybrid architectures. These models may capture long-term dependencies more effectively and adapt to intricate patterns in stock price movements.

7. Explainability and Interpretability:

- Enhance model explainability and interpretability. Develop methods to provide transparent insights into the model's decision-making process, enabling users to understand the rationale behind specific predictions and fostering trust in the model.

8. Continuous Model Monitoring:

- Implement a continuous model monitoring system to detect deviations in performance. Regularly update the model with the latest data and monitor its behaviour over time, enabling timely adjustments and ensuring sustained relevance.

9. Real-Time Prediction Integration:

- Develop mechanisms for real-time prediction and decision-making. Implement a system that allows users to receive up-to-the-minute predictions, enabling them to respond promptly to rapidly changing market conditions.

10. Cross-Market Analysis:

- Extend the model to perform cross-market analysis by incorporating data from related financial markets or global indices. Understanding inter-market relationships can provide valuable insights into potential spillover effects and systemic risks.

11. User Interface and Accessibility:

- Build a user-friendly interface for easy interaction with the model. Design dashboards and visualization tools that enable users, including investors and analysts, to intuitively explore predictions, historical trends, and model performance metrics.

12. Integration with Algorithmic Trading Platforms:

- Explore the integration of the model with algorithmic trading platforms. Develop strategies to translate predictions into actionable trading decisions, allowing for automated execution of trades based on the model's insights.

13. Ethical Considerations and Bias Mitigation:

- Address ethical considerations related to the use of the model, including potential biases. Implement measures to identify and mitigate biases in training data and model outputs, ensuring fair and unbiased predictions.

14. Collaboration with Financial Experts:

- Foster collaboration with domain experts, financial analysts, and economists. Incorporate their insights and expertise into the model development process to enhance its relevance and align with industry best practices.

15. Benchmarking and Comparative Analysis:

- Conduct benchmarking studies and comparative analyses with other state-of-the-art models and forecasting methods. Evaluate the model's performance against industry benchmarks and explore opportunities for improvement.

Continued research and development in these directions will contribute to the refinement and advancement of stock price prediction models, making them more robust, adaptive, and valuable tools for financial decision-makers in the ever-evolving landscape of financial markets.

REFERENCES

1. Machine Learning and Deep Learning in Finance:

- "Hands-On Machine Learning for Algorithmic Trading" by Stefan Jansen
- "Advances in Financial Machine Learning" by Marcos Lopez de Prado
- "Python for Finance" by Yves Hilpisch.

2. Time Series Analysis:

- "Time Series Analysis and Its Applications: With R Examples" by Robert H. Shumway and David S. Stoffer
- "Forecasting: Principles and Practice" by Rob J Hyndman and George Athanasopoulos

3. Stock Price Prediction Models:

- "Stock Price Prediction Using Machine Learning Algorithms" by Hongyu Yang, Zijun Zhang, and Yingzi Lin (Journal of Computational Intelligence)
- "A Comparative Study on Time Series Stock Price Prediction with Machine Learning Techniques" by Xinyao Sun, Siwei Cheng, and Yunyun Chen (International Conference on Artificial Intelligence and Statistics)

4. Deep Learning for Time Series:

- "Long Short-Term Memory" by Sepp Hochreiter and Jürgen Schmidhuber (Neural Computation)
- "A Comprehensive Review on Forecasting Stock Market Volatility with Artificial Intelligence" by Iman Keivanloo, Amir H. Payberah, and Seifollah Akbari (Computational Intelligence and Neuroscience)

5. Financial Data APIs and Libraries:

- Documentation for financial data APIs such as Alpha Vantage, Yahoo Finance API, or any other APIs you might have used.
- Documentation for relevant Python libraries such as pandas, NumPy, scikit-learn, TensorFlow, and Keras.

- https://www.google.co.in/
- http://www.stackoverflow.com/
- http://www.github.com/
- http://www.youtube.com/
- https://www.python.org/
- https://www.kaggle.com/datasets

APPENDIX

```python
import pandas as pd
import yfinance as yf
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import os
import re

start_date = '2022-01-01'
end_date = '2023-12-04'

tick = ['TCS','ZEEl','INFY','RVNL','IRFC','TEJASNET','RELIANCE','URJA','BCG','EXIDEIND','ZENTEC','BAJFINANCE','BAJAJFINSV',\
'IDEA','HINDALCO','ADANIENT','ADANIGREEN','ADANIPORTS','ADANITRANS','ADANIPOWER','AWL','ATGL','TATAELXSI','TATAMOTORS',\
'TATACONSUM','TATAPOWER','HSCL','VBL','BIOCON','HAPPSTMNDS','VEDL','KPITECH','SUZLON','POLYCAB','OLECTRA','WIPRO',\
'EASEMYTRIP','IRCON','IRCTC','ALOKINDS','VOLTAS','TRIDENT']

tickers = pd.DataFrame()
for i in tick:
    stock_data = yf.download('%s.NS'%i, start=start_date, end=end_date)
    stock_data['ticker'] = i
    tickers = pd.concat([tickers, stock_data])  # Corrected line

df = pd.DataFrame(tickers)

def identify_hammer_patterns(df):
    companies_with_hammer = []
    for _, row in df.iterrows():
        body_size = row['Close'] - row['Open']
        lower_shadow = row['Low'] - min(row['Open'], row['Close'])
        upper_shadow = max(row['Open'], row['Close']) - row['High']

        if body_size < 0:  # Ensure candlestick is bullish
            continue

        if body_size > 0.2 * (row['High'] - row['Low']):  # Check body size condition
            continue
```

```python
        if body_size > 0.2 * (row['High'] - row['Low']):  # Check body size condition
            continue

        if lower_shadow < 2 * abs(body_size):  # Check lower shadow condition
            continue

        if upper_shadow > 0.1 * (row['High'] - row['Low']):  # Check upper shadow condition
            continue

        companies_with_hammer.append(row['ticker'])

    return companies_with_hammer


identify_hammer_patterns(df)
```

```
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
[*********************100%%**********************]  1 of 1 completed
```

```python
import pandas as pd
import mplfinance as mpf
import matplotlib.pyplot as plt

# Assuming you have a DataFrame called 'df' with columns 'Date', 'Open', 'High', 'Low', 'Close', 'Adj Close', 'Volume'

# Filter the DataFrame to get data for a specific ticker
ticker_df = df[df['ticker'] == 'VOLTAS']

# Check if the filtered DataFrame is empty
if ticker_df.empty:
    print("No data found for the specified ticker.")
else:
    # Create a new DataFrame with OHLC (Open, High, Low, Close) data
    ohlc_df = ticker_df[['Open', 'High', 'Low', 'Close']]  # Exclude 'Date' since it's now the index

    # Plot the candlestick chart
    mpf.plot(ohlc_df, type='candle', style='yahoo', title='Hammer Candlestick Chart')

    plt.show()

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as web
import yfinance as yf
import datetime as dt
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, LSTM

# Function to Train and Predict for a Company
def train_and_predict(company):
    data = yf.download(company, start=start_date, end=end_date)

    # Data Preprocessing
    scaler = MinMaxScaler(feature_range=(0,1))
    scaled_data = scaler.fit_transform(data['Close'].values.reshape(-1,1))

    prediction_days = 60

    x_train = []
    y_train = []

    for x in range(prediction_days, len(scaled_data)):
        x_train.append(scaled_data[x-prediction_days:x, 0])
        y_train.append(scaled_data[x, 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))

# Model Definition
model = Sequential()
model.add(LSTM(units=50, return_sequences=True, input_shape=(x_train.shape[1], 1)))
model.add(Dropout(0.2))
model.add(LSTM(units=50, return_sequences=True))
model.add(Dropout(0.2))
model.add(LSTM(units=50))
model.add(Dropout(0.2))
model.add(Dense(units=1))
model.compile(optimizer='adam', loss='mean_squared_error')

# Model Training
model.fit(x_train, y_train, epochs=50, batch_size=32, verbose=0)

# Test Data
test_data = yf.download(company, start=test_start, end=test_end)
actual_prices = test_data['Close'].values
```

```python
    # Model Testing
    total_dataset = pd.concat([data['Close'], test_data['Close']], axis=0)
    model_inputs = total_dataset[len(total_dataset) - len(test_data) - prediction_days:].values

    if len(model_inputs) > 0:
        model_inputs = model_inputs.reshape(-1, 1)
        model_inputs = scaler.transform(model_inputs)

        x_test = []
        for x in range(prediction_days, len(model_inputs)):
            x_test.append(model_inputs[x - prediction_days:x, 0])

        x_test = np.array(x_test)
        x_test = np.reshape(x_test, (x_test.shape[0], x_test.shape[1], 1))

        # Predictions
        predicted_prices = model.predict(x_test)
        predicted_prices = scaler.inverse_transform(predicted_prices)

        # Visualization
        plt.plot(actual_prices, color="black", label=f"Actual {company} Price")
        plt.plot(predicted_prices, color="green", label=f"Predicted {company} Price")
        plt.title(f"{company} Share Price")
        plt.xlabel('Time')
        plt.ylabel(f'{company} Share Price')
        plt.legend()
        plt.show()

        # Real Data for Prediction
        real_data = [model_inputs[len(model_inputs) - prediction_days:len(model_inputs), 0]]
        real_data = np.array(real_data)
        real_data = np.reshape(real_data, (real_data.shape[0], real_data.shape[1], 1))

        # Final Prediction
        prediction = model.predict(real_data)
        prediction = scaler.inverse_transform(prediction)

        # Output
        dd = pd.DataFrame(data.to_records())
        dd = dd[dd.Date == td_date]
        dd.reset_index(drop=True, inplace=True)

        if not dd.empty:
            print(f"{company} On Date : {td_date} : {dd['Close'].iloc[0]}")
            print(f"Prediction : {prediction}")
        else:
            print(f"No data available for {company} on {td_date}.")
    else:
        print(f"No data available for {company}.")

# Main Loop Over Multiple Companies
td_date = '2023-12-03'
start_date = '2022-01-01'
end_date = '2023-12-04'

test_start = start_date
test_end = end_date

tickers = ['TCS', 'ZEEl', 'INFY', 'RVNL', 'IRFC', 'TEJASNET', 'RELIANCE', 'SUZLON', 'URJA', 'BCG', 'EXIDEIND', 'ZENTEC',
           'BAJFINANCE', 'BAJAJFINSV', 'IDEA', 'HINDALCO', 'ADANIENT', 'ADANIGREEN', 'ADANIPORTS', 'TATASTEEL',
           'ADANIPOWER', 'AWL', 'ATGL', 'TATAELXSI', 'TATAMOTORS', 'TATACONSUM', 'TATAPOWER', 'HSCL', 'VBL', 'BIOCON',
           'HAPPSTMNDS', 'VEDL', 'SUZLON', 'POLYCAB', 'OLECTRA', 'WIPRO', 'EASEMYTRIP', 'IRCON', 'IRCTC', 'ALOKINDS',
           'VOLTAS', 'TRIDENT']

for ticker in tickers:
    train_and_predict(ticker)
```