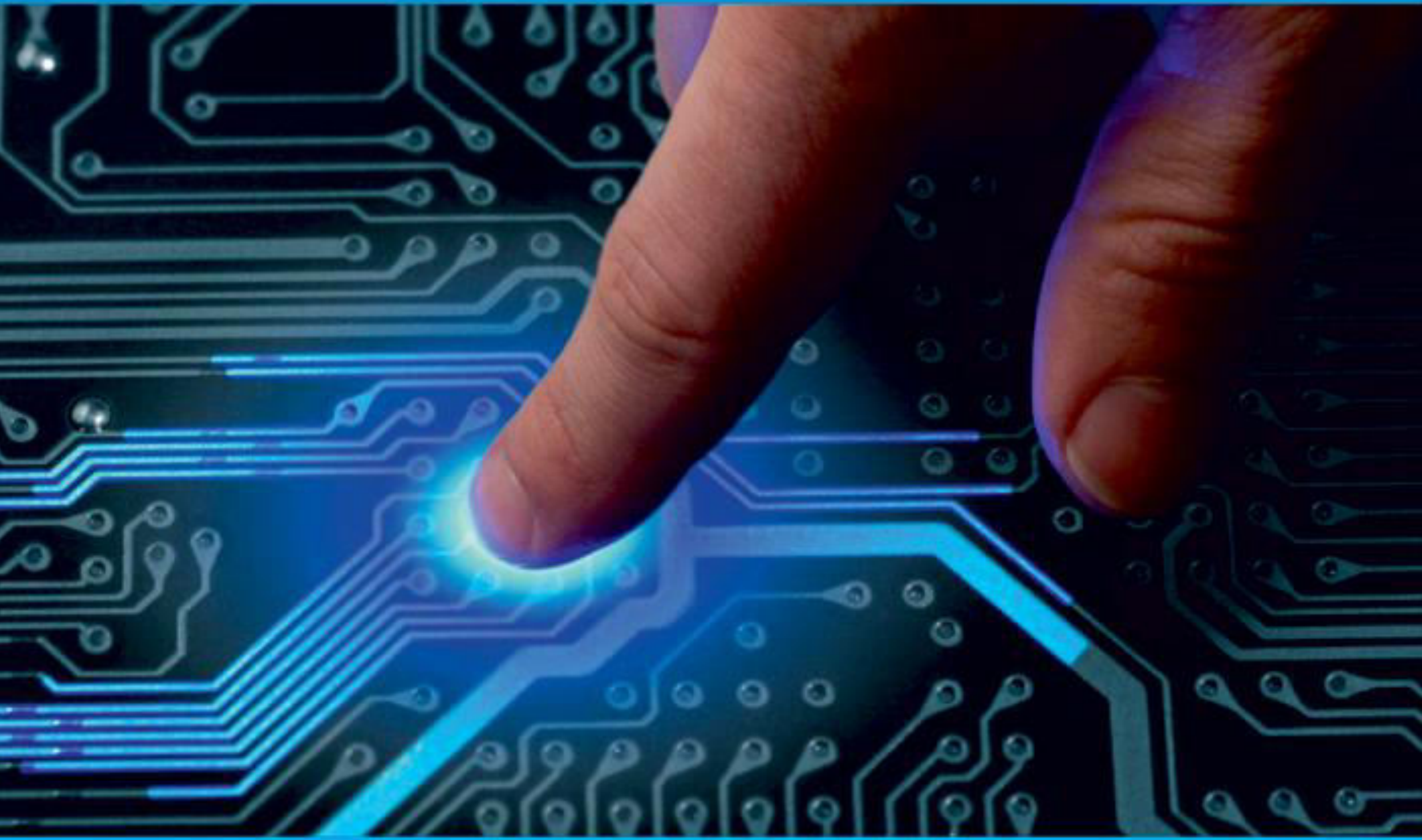
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 12, Issue 11, November 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.625

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Stock Market Prediction Using Machine Learning Algorithms

Chaitanya Ajay Katare, Pawan Vilas Shinde, Pallavi Subhash Kothare, Prof. (Mrs) H. A. Navare

U.G. Student, Department of Electronics and Telecommunication Engineering, Sinhgad College of Engineering, Wadgaon, Pune, Maharashtra, India

U.G. Student, Department of Electronics and Telecommunication Engineering, Sinhgad College of Engineering, Wadgaon, Pune, Maharashtra, India

U.G. Student, Department of Electronics and Telecommunication Engineering, Sinhgad College of Engineering, Wadgaon, Pune, Maharashtra, India

Associate Professor, Department of Electronics and Telecommunication, Sinhgad College of Engineering, Wadgaon, Pune, Maharashtra, India

ABSTRACT:

This project aims to leverage machine learning algorithms, specifically Random Forest and Long Short-Term Memory (LSTM) networks, to predict stock market trends, particularly focusing on the Nifty 50 index.
Data Acquisition and Preprocessing:
Real-time stock data is extracted from Yahoo Finance using the yfinance library
The data is preprocessed to ensure data quality and suitability for model training.
Model Development:
Random Forest: A supervised learning algorithm that creates multiple decision trees and combines their predictions.
LSTM: A type of recurrent neural network well-suited for time series data, capable of capturing long-term dependencies.
Model Training and Evaluation:
The models are trained on historical stock data to learn patterns and trends.
The performance of the models is evaluated using metrics like accuracy and mean squared error.
As of now , We have more then 50% accuracy in our random forest model and we are yet to test out with Long short term memory model .
I hope you do like our project since it took great effort for us to learn these model in depth and to learn , search machine learning libraries.
On "date- 5th-11-24" the predicted price was accurate and last 5 days before the given date , with the accuracy of 50% meaning 50% times the model was correct about the prediction, later on this project after comparing lstm we will develop a model with much greater accuracy.
Trading has just started in india and there are much more updates going to come due to ease of technology.
Traders now a days use more technical tools then there were ever .
Our project is not a technical tools rather a solution to user , an ai solution to the given query of predictions.
Future Work:
Developing a user-friendly interface for real-time predictions and visualizations.
This project has the potential to provide valuable insights to investors and traders, empowering them to make informed decisions in the dynamic stock market.

KEYWORDS: scikit learn library, pandas, yfinance library

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

I. INTRODUCTION

The stock market, a complex and dynamic system, has long fascinated investors and researchers alike. Accurate prediction of stock prices could potentially yield significant financial returns. However, the inherent volatility and unpredictability of the market pose significant challenges to traditional forecasting methods.

In recent years, machine learning has emerged as a powerful tool for analyzing complex data and making accurate predictions. By leveraging advanced algorithms, researchers have explored the potential of machine learning to forecast stock prices with greater precision. This research paper investigates the effectiveness of two prominent machine learning techniques: Random Forest and Long Short-Term Memory (LSTM) networks, in predicting stock price movements, specifically focusing on the Nifty 50 index.

Data Acquisition and Preprocessing

To train and evaluate our models, we acquired real-time stock data from Yahoo Finance using the yfinance library. The collected data includes various features such as opening price, closing price, highest price, lowest price, and trading volume. Before feeding the data to the models, we performed necessary preprocessing steps, including:

Data Cleaning: outliers to ensure data quality and Handling missing values .
Feature Engineering: Creating additional features like moving averages and technical indicators to enhance model performance.
Data Normalization: Scaling the features to a common range to improve model convergence.

Model Development and Training

1) **Random Forest** Random Forest is an ensemble learning method that combines multiple decision trees to make accurate predictions. Each decision tree is trained on a random subset of the 1 data and features, reducing overfitting and improving generalization

2) **Long Short-Term Memory** (LSTM) Networks LSTM networks are a type of recurrent neural network specifically designed to capture long-term dependencies in sequential data. They are well-suited for time series analysis, such as stock price prediction, as they can learn from past patterns and trends.

We trained both models on historical stock data, using a portion of the data for training and the remaining portion for validation and testing. The models were evaluated using metrics like accuracy and mean squared error to assess their predictive performance.

II.LITERATURE SURVEY

The prediction of stock prices has become a prominent and captivating area of research within both financial and academic fields to understand the dynamics of economies. There has yet to be a comprehensive set of guidelines for estimating and forecasting stock values in the stock market. Numerous advanced technologies, including technical, fundamental, time series, statistical, and series analysis, exist to assist in the prediction process; however, none have been validated as trustworthy and precise tools for society in estimating stock market or share market trends. In this paper, we endeavor to introduce innovative work utilizing a machine-learning approach to forecast or analyze the behavioral patterns of the stock market's Sensex. We have effectively implemented Machine Learning models such as Linear Regression, Support Vector Regression, Decision Tree, Random Forest Regressor, and Extra Tree Regressor to predict stock prices and elucidate the interactions between buyers and sellers in the exchanges. Our stock price predictions are based on the closing values and stock prices. We conduct a comparative analysis of each model's accuracy by employing a highly accurate algorithm, ultimately identifying the most effective algorithm for stock price prediction. Given that the share market is an inherently uncertain domain,

www.ijircce.com | e-ISSN: 2320-9801, p-ISSN: 2320-9798| Impact Factor: 8.625| ESTD Year: 2013|

International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)

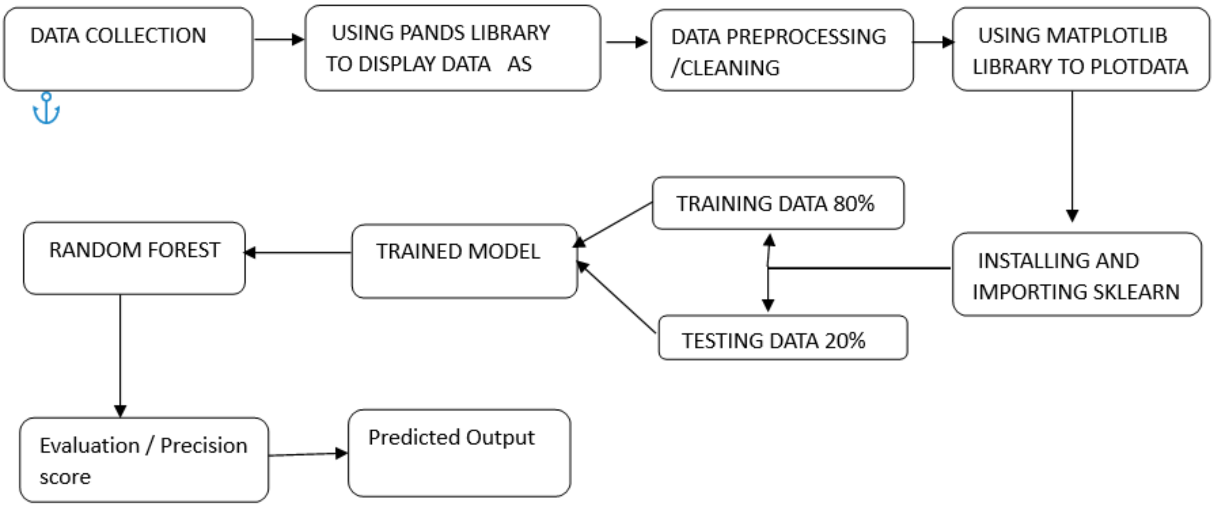
(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

accurately predicting its conditions remains a challenge; nonetheless, this task can be approached systematically and technically through our research, with the primary objective of this paper being the application of Machine Learning algorithms to forecast stock prices.

III.METHODOLOGY

The random forest refers to an estimator that analyzes data about data. It constructs multiple decision trees using various subsets of the provided dataset. It effectively manages the issue of overfitting. It enhances the accuracy of predictions. Algorithm: Step 1: Select N random records from the dataset. Step 2: Create a decision tree based on the N records. Step 3a: From your algorithm, determine the number of trees and repeat steps 1 and 2. Step 3b: In the case of a regression problem, each tree in the forest generates a prediction for Y (output) for a new record.

IV.DESIGN AND DEVELOPMENT


DATA COLLECTION
USING PANDS LIBRARY TO DISPLAY DATA AS
DATA PREPROCESSING /CLEANING
USING MATPLOTLIB LIBRARY TO PLOTDATA
TRAINING DATA 80%
RANDOM FOREST
TRAINED MODEL
INSTALLING AND IMPORTING SKLEARN
TESTING DATA 20%
Evaluation / Precision score
Predicted Output

BLOCK DIAGRAM: Random Forest

1.Data Collection
Description: Gather stock market data, including historical prices and other indicators, from reliable sources such as Yahoo Finance or other financial APIs.
Purpose: To provide a comprehensive dataset of past stock performance, which the model will analyze to predict future trends.
2. Using Pandas Library to Display Data
Description: Use the Pandas library to load and view the collected data in a structured, tabular format.
Purpose: Allows for an initial inspection of the dataset, making it easier to identify any immediate issues or inconsistencies in the data.
3. Data Preprocessing / Cleaning
Description: Prepare the data by cleaning it—removing unnecessary columns and adding extra coulumns such as “Tomorrow column ” and “ Target column ”handling missing values, normalizing, and scaling it for consistency.
Purpose: Ensures high data quality, which is essential for accurate model predictions, and makes the data suitable for training.
4. Using Matplotlib Library to Plot Data
Description: Visualize the data to detect patterns and trends over time using line charts, or other visual aids.
Purpose: Helps understand the data better, providing insights that can guide the modeling process.

5. Installing and Importing Sklearn

Description: Sklearn is a machine learning library that provides tools for building, training, and evaluating models. Such as it provides us with Random forest Classifier .

Purpose: Supplies the essential algorithms and utilities needed to develop the project

6. Training Data (80%) and Testing Data (20%)

Description: Split the dataset into training (80%) and testing (20%) portions.

Purpose: Allows the model to learn from one subset and be evaluated on another, ensuring it generalizes well to new data.

7. Trained Model

Description: The result of applying the Random Forest algorithm to the training data, where the model learns patterns in historical stock data.

Purpose: To develop a model that can make accurate predictions based on the trends it has learned.

8. Random Forest

Description: Random Forest is an ensemble method that creates multiple decision trees and combines their outputs to improve prediction accuracy.

Purpose: Its strength in handling complex data makes it suitable for analyzing volatile stock market patterns.

9. Evaluation / Precision Score

Description: Assess the model's accuracy using metrics like mean absolute error, root mean squared error, and R² score.

Purpose: Evaluation ensures the model's predictions are reliable and informs any adjustments needed for better performance.

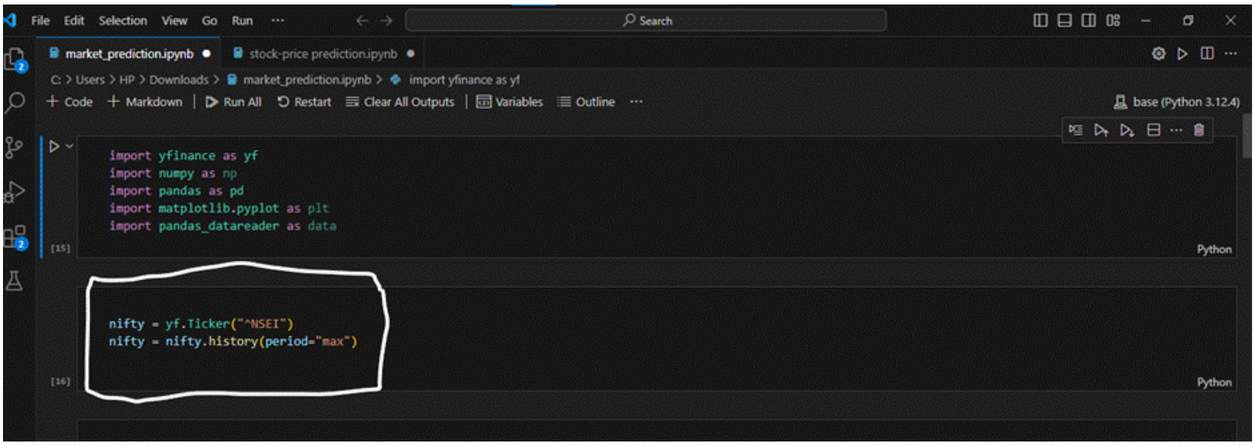
10. Predicted Output

Description: The model's final prediction, indicating the expected stock price or trend for a future time frame.

Purpose: Provides actionable insights for traders or investors, guiding their buying or selling decisions based on forecasted prices.

IV. EXPERIMENTAL RESULTS

F As mentioned before Real-time stock data is extracted from Yahoo Finance using the yfinance library nifty here is variable that holds our ticker class value of yfinance library .



File Edit Selection View Go Run ···
Search
market_prediction.ipynb
stock-price prediction.ipynb
C: > Users > HP > Downloads > market_prediction.ipynb > import yfinance as yf
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ···
base (Python 3.12.4)

```
import yfinance as yf
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader as data
```

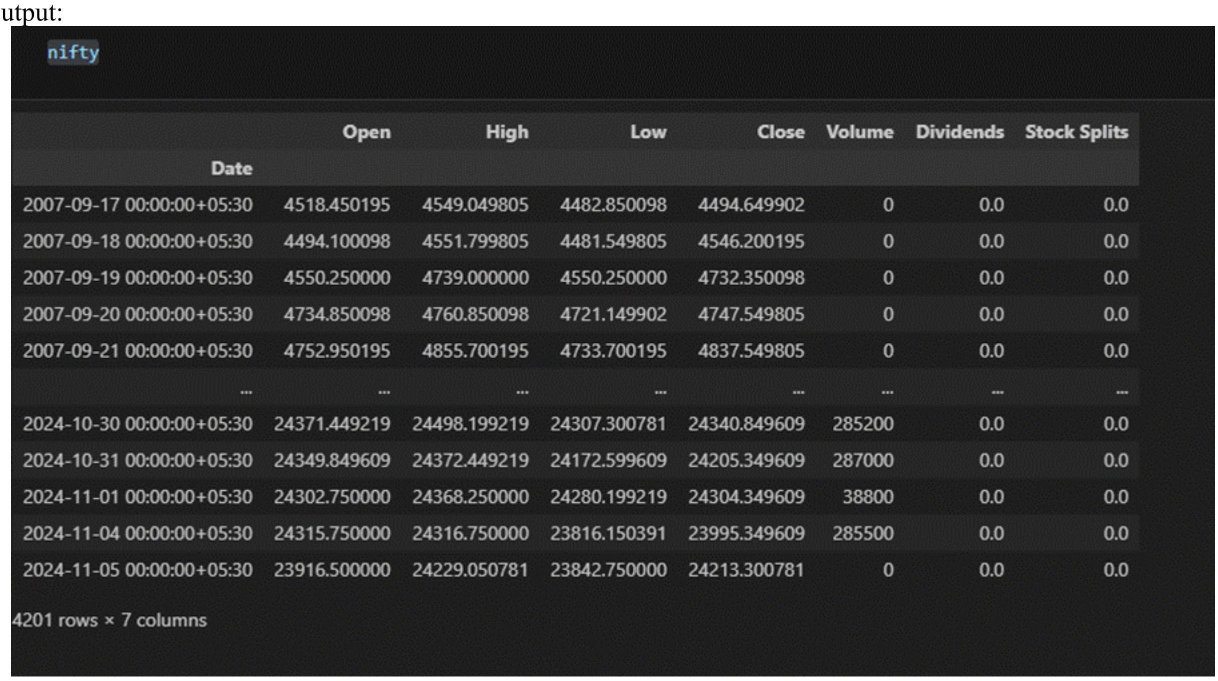
```
nifty = yf.Ticker("^NSEI")
nifty = nifty.history(period="max")
```

USING PANDAS:

Using pandas library we are displaying the nifty table as nifty

```
import pandas as pd
nifty = nifty.history(period="max")
nifty
```

output:



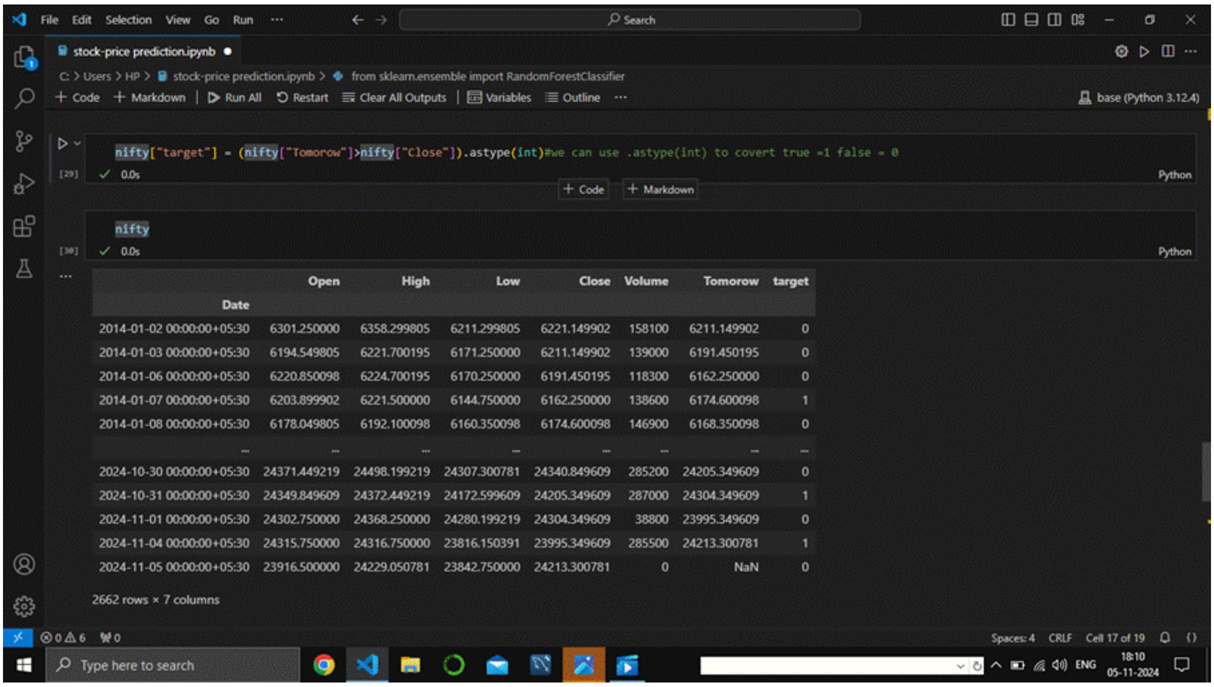
```python
nifty
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2007-09-17 00:00:00+05:30	4518.450195	4549.049805	4482.850098	4494.649902	0	0.0	0.0
2007-09-18 00:00:00+05:30	4494.100098	4551.799805	4481.549805	4546.200195	0	0.0	0.0
2007-09-19 00:00:00+05:30	4550.250000	4739.000000	4550.250000	4732.350098	0	0.0	0.0
2007-09-20 00:00:00+05:30	4734.850098	4760.850098	4721.149902	4747.549805	0	0.0	0.0
2007-09-21 00:00:00+05:30	4752.950195	4855.700195	4733.700195	4837.549805	0	0.0	0.0
...	...	...	...	...	...	...	...
2024-10-30 00:00:00+05:30	24371.449219	24498.199219	24307.300781	24340.849609	285200	0.0	0.0
2024-10-31 00:00:00+05:30	24349.849609	24372.449219	24172.599609	24205.349609	287000	0.0	0.0
2024-11-01 00:00:00+05:30	24302.750000	24368.250000	24280.199219	24304.349609	38800	0.0	0.0
2024-11-04 00:00:00+05:30	24315.750000	24316.750000	23816.150391	23995.349609	285500	0.0	0.0
2024-11-05 00:00:00+05:30	23916.500000	24229.050781	23842.750000	24213.300781	0	0.0	0.0

4201 rows × 7 columns

Next step is Data cleaning and using proper data set:

New coulumns such as tomorrow is created which is yesterdays closing price and another column called target is created which is 1 if tomorrows price > yesterdays closing price



File Edit Selection View Go Run ...
Search
stock-price prediction.ipynb
C: > Users > HP > stock-price prediction.ipynb > from sklearn.ensemble import RandomForestClassifier
+ Code + Markdown | Run All Restart Clear All Outputs | Variables Outline ...
base (Python 3.12.4)
Python
Type here to search
Spaces: 4 CRLF Cell 17 of 19
ENG
18:10
05-11-2024

```python
nifty["target"] = (nifty["Tomorow"]>nifty["Close"]).astype(int)#we can use .astype(int) to covert true =1 false = 0
```

[29] ✓ 0.0s

```python
nifty
```

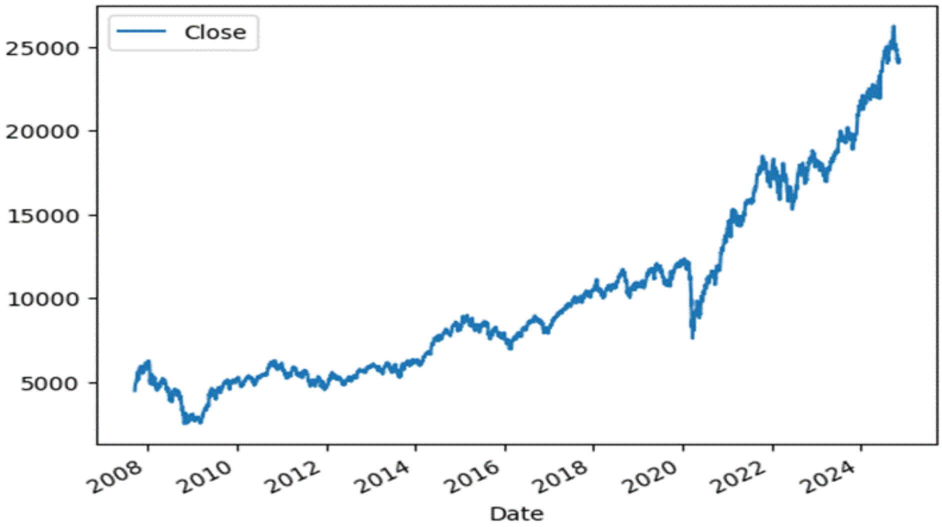
[30] ✓ 0.0s

	Open	High	Low	Close	Volume	Tomorow	target
Date							
2014-01-02 00:00:00+05:30	6301.250000	6358.299805	6211.299805	6221.149902	158100	6211.149902	0
2014-01-03 00:00:00+05:30	6194.549805	6221.700195	6171.250000	6211.149902	139000	6191.450195	0
2014-01-06 00:00:00+05:30	6220.850098	6224.700195	6170.250000	6191.450195	118300	6162.250000	0
2014-01-07 00:00:00+05:30	6203.899902	6221.500000	6144.750000	6162.250000	138600	6174.600098	1
2014-01-08 00:00:00+05:30	6178.049805	6192.100098	6160.350098	6174.600098	146900	6168.350098	0
...	...	...	...	...	...	...	...
2024-10-30 00:00:00+05:30	24371.449219	24498.199219	24307.300781	24340.849609	285200	24205.349609	0
2024-10-31 00:00:00+05:30	24349.849609	24372.449219	24172.599609	24205.349609	287000	24304.349609	1
2024-11-01 00:00:00+05:30	24302.750000	24368.250000	24280.199219	24304.349609	38800	23995.349609	0
2024-11-04 00:00:00+05:30	24315.750000	24316.750000	23816.150391	23995.349609	285500	24213.300781	1
2024-11-05 00:00:00+05:30	23916.500000	24229.050781	23842.750000	24213.300781	0	NaN	0

2662 rows × 7 columns

USING MATPLOTLIB:

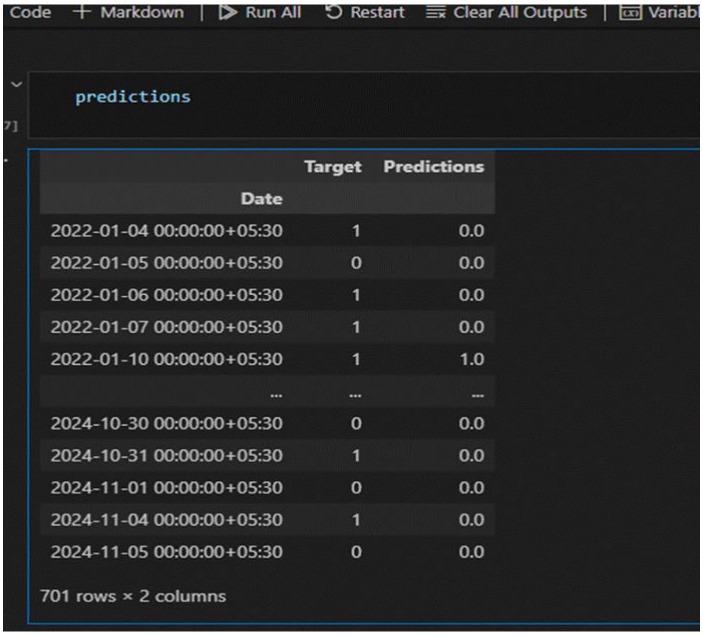
We plot the nifty graph where X AXIS label = index that is the date column and Y axis is a closing price :
Output:



USING RANDOM FOREST CLASSIFIER TO TRAIN AND TEST DATASET:

SKLEARN is the library that holds randomforest classifier hence you must install it first
Here we train the data using train and test variables which are values except last 100 rows to train on old data first For Training the model on historical data then the test data set is trained from last 100 rows

Final output :



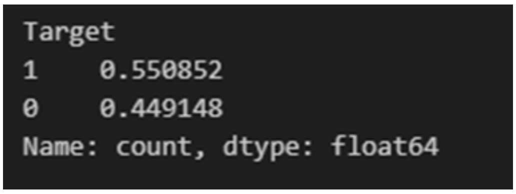
```
predictions
```

Date	Target	Predictions
2022-01-04 00:00:00+05:30	1	0.0
2022-01-05 00:00:00+05:30	0	0.0
2022-01-06 00:00:00+05:30	1	0.0
2022-01-07 00:00:00+05:30	1	0.0
2022-01-10 00:00:00+05:30	1	1.0
...	...	...
2024-10-30 00:00:00+05:30	0	0.0
2024-10-31 00:00:00+05:30	1	0.0
2024-11-01 00:00:00+05:30	0	0.0
2024-11-04 00:00:00+05:30	1	0.0
2024-11-05 00:00:00+05:30	0	0.0

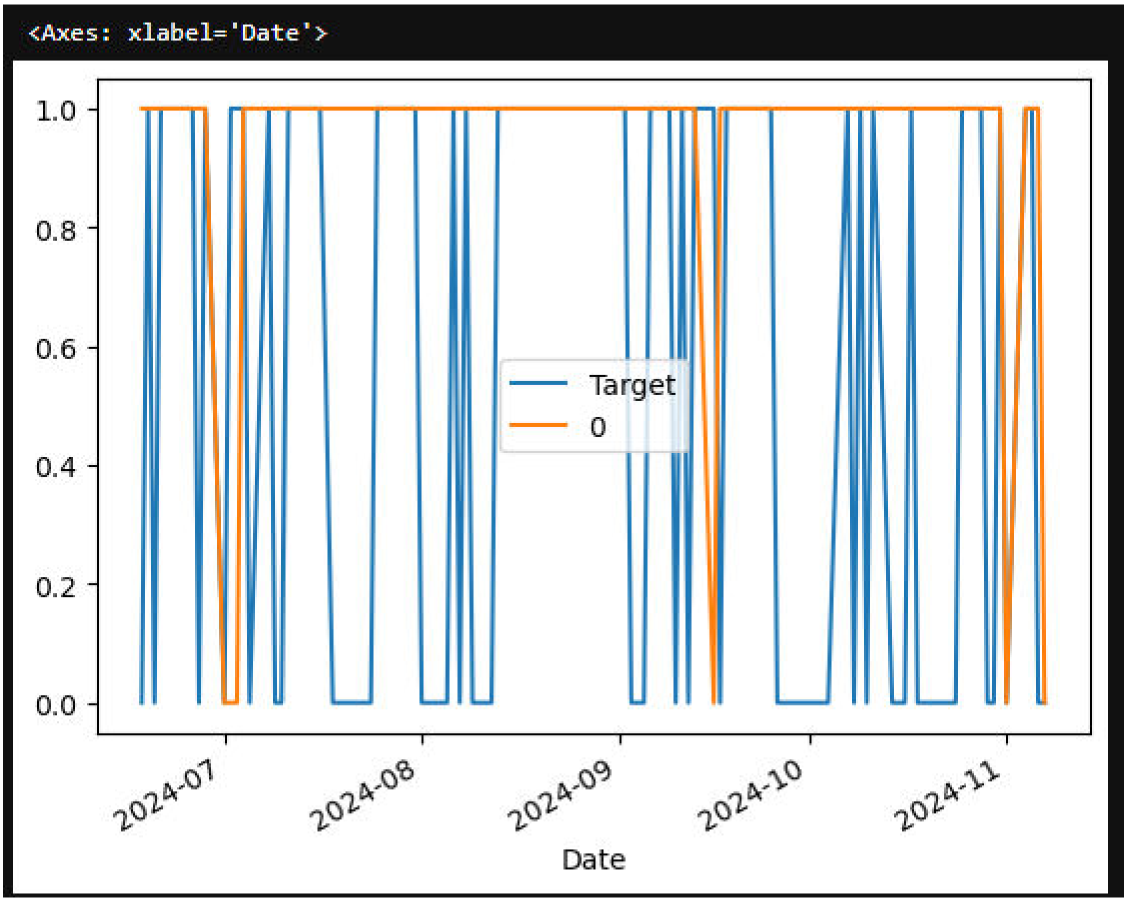
701 rows × 2 columns

The TARGET[0] specifies a down trend and TARGET[1] specifies up trend for the next day.

Precision score of the model :



```
Target
1    0.550852
0    0.449148
Name: count, dtype: float64
```



```
<Axes: xlabel='Date'>
```

Variations of target price which is what actually happened as per the given formula on training that is closing price was greater than opening price and zero line or orange line shows what actually our model predicted .

V. CONCLUSION

Machine learning algorithms present a promising approach to stock price prediction by leveraging their ability to analyze vast amounts of historical data and identify intricate patterns that may elude traditional methods. These insights offer investors a data-driven foundation for making informed decisions in a highly dynamic market environment. However, to unlock the full potential of these models, challenges such as data quality, feature selection, and model complexity must be carefully managed. High-quality data is essential, as even small inconsistencies can significantly impact prediction accuracy. Additionally, as models grow more complex to capture market nuances, they may become harder to interpret and require greater computational resources.

REFERENCES

[2] Stock Market Prediction Using LSTM Recurrent Neural Network – Science Direct (IWSMAI 2020) April 6-9, 2020, warsaw, Poland.

[3] A Survey on Stock Market Prediction Using Machine Learning Techniques Polamuri Subba Rao1(&), K. Srinivas2, and A. Krishna Mohan3.

[4] Stocks Market Prediction Using Support Vector Machine Zhen Hu 1, Jie Zhu 2, and Ken Tse 3 2013 6th International Conference on Information Management.

[5] CNNpre d: CNN-base d stock market prediction using a diverse set of variables Ehsan Hoseinzade, Saman Haratizadeh 20 March 2019

Follow this project on github @ -https://github.com/pawannshinde/stock-market-prediction

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

www.ijircce.com

Scan to save the contact details