Stock Analysis and Prediction

Course: PGDBDA

By
Akhil Kushwaha(PRN 240320525002)
Vedant Gharat(PRN 240320525012)

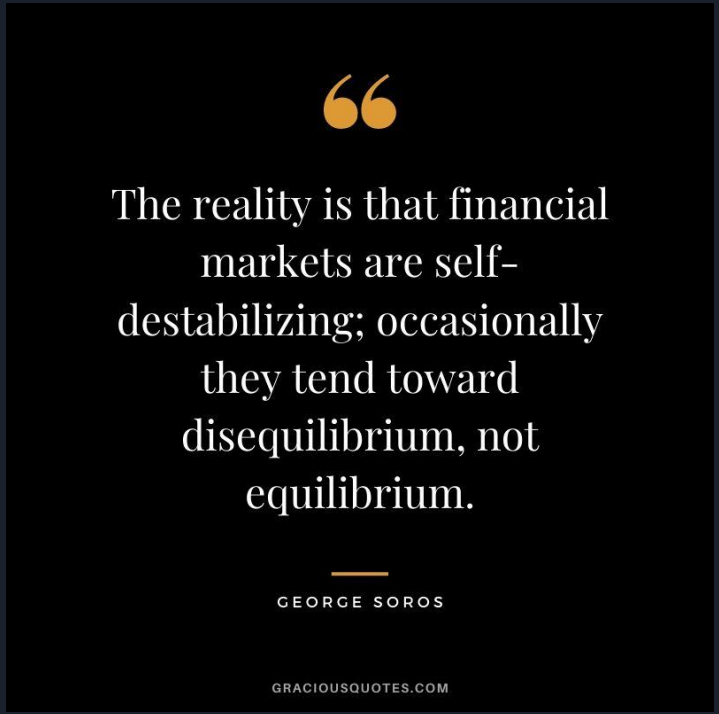
PROBLEM STATEMENT

The challenge of this project is to accurately predict the future closing value of a given stock across a given period of time in the future. For this project we will use a Linear Regression Model to predict the closing price of the Nifty 50 using a dataset of past prices.

Analyze opening and closing price to oversee trends over time

Stock Market has many uncertainties.

So, there can not be a perfect model for prediction of stock price of every company. We need to develop models which suits to almost every data with more accuracy.



Goals

Explore stock prices.

Preprocess the data.

Implement Linear regression

Evaluate the performance of model and update the Model.

Compare the results and submit the report.

Linear Regression Model

Linear regression is a statistical method used to model and analyze the relationship between two or more variables. It's one of the simplest and most widely used techniques in data analysis and machine learning.

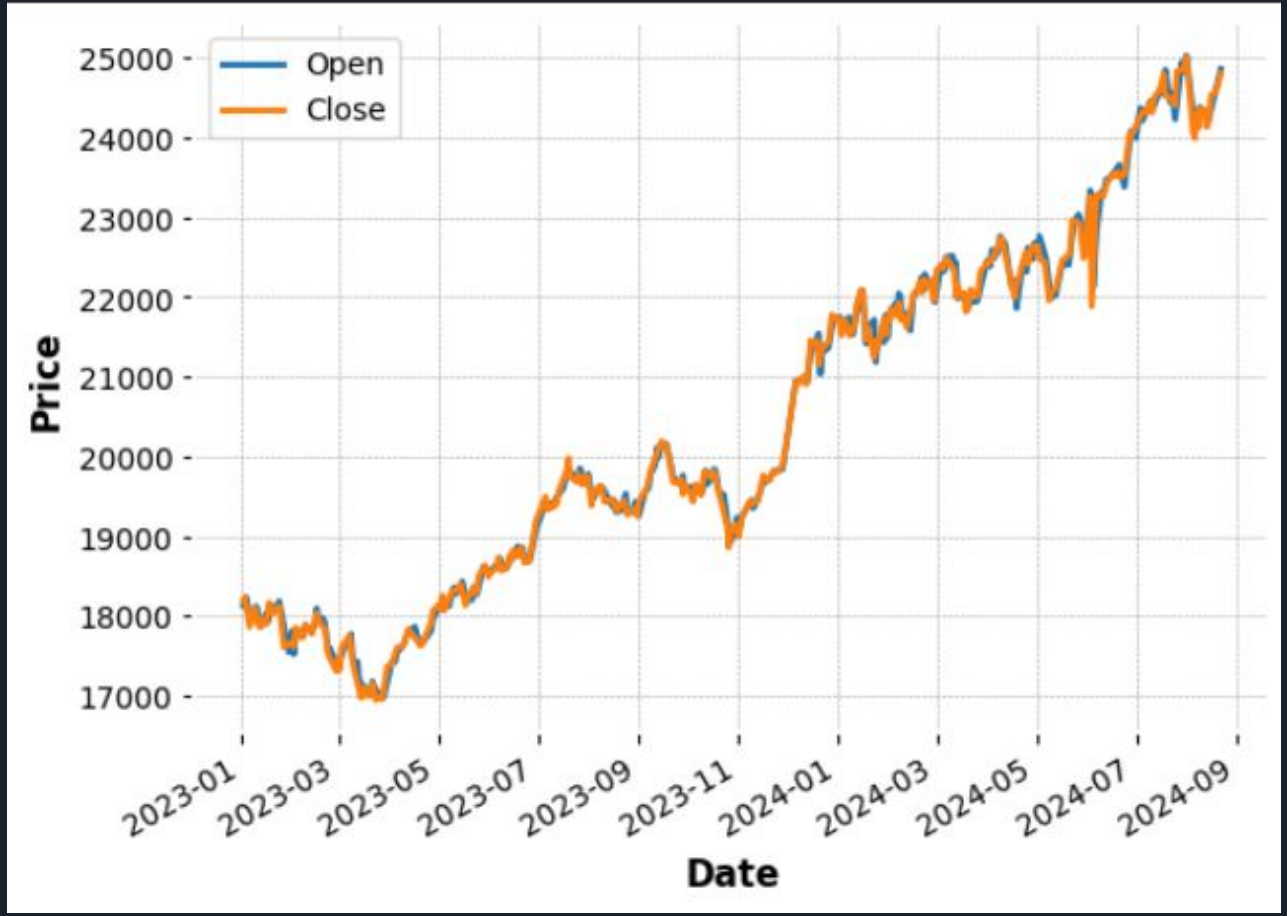
At its core, linear regression aims to find the linear relationship between a dependent variable (also called the target or outcome) and one or more independent variables (also known as predictors or features). The goal is to fit a line (or hyperplane in the case of multiple predictors) to the data that best represents this relationship.

Dataset

The data is pulled live using Yahoo Finance API we can pull data till current date. For the purpose of our analysis and to minimize the number of outliers we will focus on the data of last one year

Data set is in following form:

Date	Open	High	Low	Close
2023-01-02	18131.699219	18215.150391	18086.500000	18197.449219
2023-01-03	18163.199219	18251.949219	18149.800781	18232.550781
2023-01-04	18230.650391	18243.000000	18020.599609	18042.949219
2023-01-05	18101.949219	18120.300781	17892.599609	17992.150391
2023-01-06	18008.050781	18047.400391	17795.550781	17859.449219

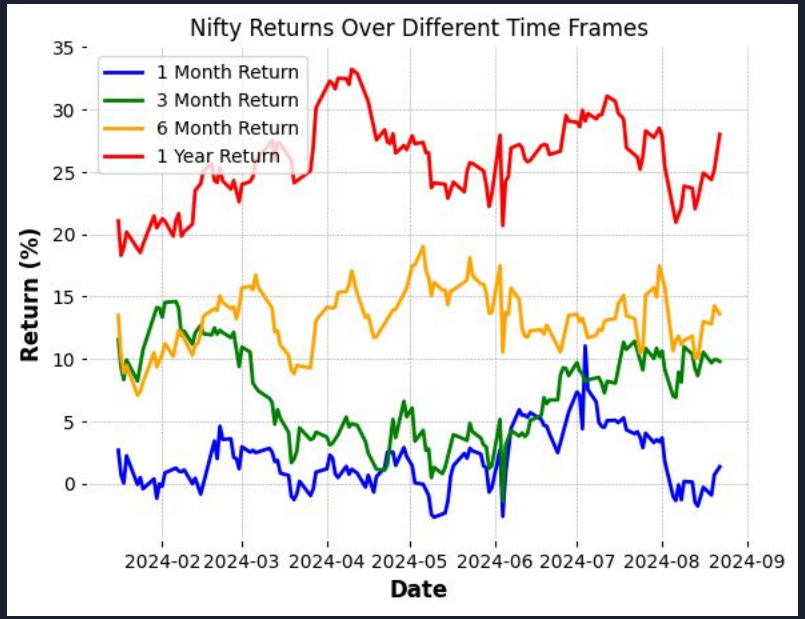

Open
Close
25000
24000
23000
22000
21000
20000
19000
18000
17000
Price
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
Date

Analyzing Nifty 50 Price Data

```
def calculate_return(df,period):
    df['Return_{}'.format(period)] = df['Close'].pct_change(periods=period)*100
    return df

# 1 Month return
nifty_return = calculate_return(df,21)

# 3 Month Return
nifty_return = calculate_return(df,63)

# 6 Month Return
nifty_return = calculate_return(df,126)

# 1 Year Return
nifty_return = calculate_return(df,256)

nifty_return = nifty_return.dropna()


plt.plot(nifty_return.index,nifty_return['Return_21'],label = '1 Month Return', color = "Blue")
plt.plot(nifty_return.index,nifty_return['Return_63'],label = '3 Month Return', color = "Green")
plt.plot(nifty_return.index,nifty_return['Return_126'],label = '6 Month Return', color = "Orange")
plt.plot(nifty_return.index,nifty_return['Return_256'],label = '1 Year Return', color = "Red")

plt.xlabel('Date')
plt.ylabel('Return (%)')
plt.title('Nifty Returns Over Different Time Frames')
plt.legend()
plt.grid(True)
plt.show()
```

Analyzing Nifty 50 Price Data

The following code calculates and visualizes the percentage returns for a stock (or index) over various time periods (1 month, 3 months, 6 months, 1 year). It helps in understanding how the stock's price has changed over these periods, which can be useful for analyzing trends, performance, and making informed investment decisions.



Trends

EMA helps traders and analysts identify the direction of a trend. EMA sloping upwards indicates uptrend and vice versa.

A line graph used in stock market analysis, showcasing the price movements of a stock over time along with three Exponential Moving Averages (EMAs). Here’s a breakdown of the key elements:

Price Line (Blue): This line represents the actual price of the stock over the given period (from January 2023 to September 2024). It fluctuates, showing the daily closing prices.

EMA 50 (Red): This is the 50-day Exponential Moving Average. It smooths out the price data by giving more weight to recent prices

EMA 100 (Green): This is the 100-day Exponential Moving Average. It provides a medium-term view of the stock's trend, balancing between short-term fluctuations and long-term trends.

EMA 200 (Orange): This is the 200-day Exponential Moving Average. It offers a long-term perspective, smoothing out even more of the short-term noise to highlight the overall trend.


Price
EMA_200
EMA_100
EMA_50
25000
24000
23000
22000
21000
20000
19000
18000
17000
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
Date

Preprocessed Data

Date	Open	High	Low
2024-03-28	22163.599609	22516.000000	22163.599609
2023-02-27	17428.599609	17451.599609	17299.000000
2023-10-19	19545.199219	19681.800781	19512.349609
2023-12-18	21434.800781	21482.800781	21365.349609
2024-07-11	24396.550781	24402.650391	24193.750000

Splitted the dataset into the training and test datasets for the Linear Regression model. The split had following shape

Xtrain (281,3) Xtest(281,)

Ytrain(121,3) Ytest(121,)

Model Building

```
[91] x = df.iloc[:,0:3]
     y = df["Close"]
```

```
# Linear Regression Algorithm
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.3,random_state=1)
```

```
# checking accuracy of the model

from sklearn.metrics import r2_score
print("Accuracy of the linear Regression Model",r2_score(ytest,ypred))
```

```
Accuracy of the linear Regression Model 0.9995836627710786
```

Why use Linear Regression

```
Linear Regression MAE: 34.41296634293975
Random Forest MAE: 64.17503712551654
LSTM MAE: 20630.732506350054
```

Linear Regression: Has the lowest MAE, meaning it performs relatively well in this setup.

Random Forest: Performs better than the LSTM but might still have room for improvement.

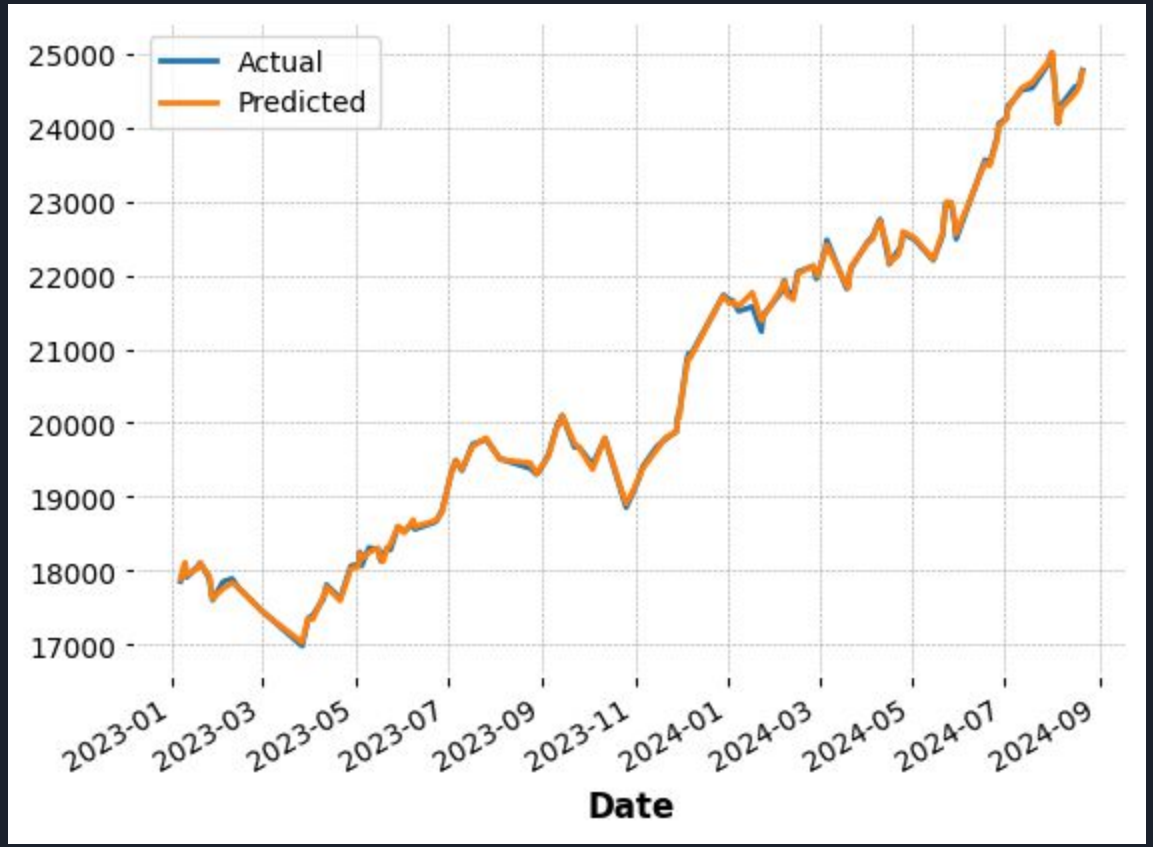
LSTM: The high MAE indicates that the LSTM model isn't performing well with the current setup. This could be due to various factors like inadequate hyperparameter tuning, sequence length, or data preprocessing.

Future Scope

Improve LSTM: Consider experimenting with hyperparameters (e.g., number of LSTM units, sequence length, learning rate) or trying other neural network architectures.

Feature Engineering: Add more relevant features like additional technical indicators or even sentiment analysis data.

Model Tuning: Further optimize the Random Forest and Linear Regression models with hyperparameter tuning.


Actual
Predicted
25000
24000
23000
22000
21000
20000
19000
18000
17000
2023-01
2023-03
2023-05
2023-07
2023-09
2023-11
2024-01
2024-03
2024-05
2024-07
2024-09
Date

CONCLUSION

There is an overall upward trend from January 2023 to September 2024. This suggests that the asset's value has generally increased during this period.

Traders and investors can use these EMAs as indicators. For instance, if the price crosses above EMA_50 and EMA_100, it could signal a bullish trend. Conversely, crossing below might indicate a bearish trend.

Remember that no single indicator guarantees success, so always consider other factors and risk management strategies.

Use of this model

Linear regression is used in:

- Real Estate
- Social Sciences
- Environmental Science
- Healthcare and Medicine
- Marketing

Etc.

Thank you!