STOCK PRICE PREDICTION USING PYTHON

Mini Project report submitted
in partial fulfilment of the requirement for the degree of

Bachelor of Technology

By

MRINAB DEY (180610026026)
PANKAJ KUMAR SAH (180610026032)

Under the guidance
of

NISHANT BHARTI, Assistant Professor, ETE Dept
SMITA SARMA, Assistant Professor, ETE Dept

DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
ASSAM ENGINEERING COLLEGE
JALUKBARI- 781013, GUWAHATI

JULY, 2021

ASSAM ENGINEERING COLLEGE, GUWAHATI

CERTIFICATE

This is to certify that the report entitled "Stock Price Prediction Using Python" submitted by Mrinab Dey (180610026026) and Pankaj Kumar Sah (180610026032) of B. Tech 6th semester, Electronics & Telecommunication Engineering, is an authentic work carried out by them under my supervision and guidance.

To the best of my knowledge, the matter embodied in the report has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Signature of Supervisor(s)
Nishant Bharti, Assistant Professor, ETE Dept
Smita Sarma, Assistant Professor, ETE Dept
Electronics and Telecommunication Engineering
Assam Engineering College
26th July,2021

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)	(Signature)
Mrinab Dey	Pankaj Kumar Sah
180610026026	180610026032
Date: 26th July, 2021	Date: 26th July 2021

ACKNOWLEDGMENT

We would like to take this opportunity to thank our Head of the Department, Prof. Dinesh Shankar Pegu, HOD, ETE Department for giving us the opportunity to do a mini project on “Stock Price Prediction using Python”. We would also like to thank our Guide Nishant Bharati Assistant Professor, ETE Department along with our co-guide Smita Sarma, Assistant Professor, ETE Department for guiding us throughout the project, for providing valuable suggestions, for their ongoing support during the project, from initial advice and provision of contacts in the first stages through ongoing advice and encouragement, which led to the final report of this mini project. We are also grateful to other teachers of the department for contribution their inputs in this project.

MRINAB DEY (180610026026)
PANKAJ KUMAR SAH (180610026032)

ABSTRACT

In this project, we worked in the area of artificial intelligence. Artificial intelligence is the method in which we use certain algorithms and data to provide machines with an ability to think like a human. With the aid of artificial intelligence, machines get the ability to analyze a problem and make decisions based on the analysis to solve the problem. The importance of work can be determined from the fact that it will help us in determining the future price of a stock so that we can decide which stocks we need to invest in to earn good profits out of the market. And as all the predictions are made by machines themselves, no human effort is needed which eventually saves us a lot of time.

In the project, we have used the previous price of a stock over a period of five years in order to determine the current price of the stock. This data was then pre-processed and fed into the model in order to train it. After the training is complete, our model was able to predict the stock prices.

The result that was achieved at the end of this project was quite impressive as the model was able to predict the trend successfully, it wasn't 100% accurate but considering that it predicted only on the basis of the past data is quite impressive. The accuracy of the results showed that we could predict the stock prices quite accurately which would mean that we could use the same model to invest in the stock markets to earn good profits without much effort.

Considering the fact that the model used only the previous data to predict the future prices still it gave a decent result. The precision of the result can be increased by using market sentiments into it. Throughout this project we used python as the programming language, Spyder as the IDE and used many python libraries viz. tensorflow, scikit learn, plotly, streamlit, numpy, pandas and keras.

LIST OF FIGURES

CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Introduction:

In this chapter, we will be discussing about the area of work through a brief introduction. We will also discuss about the brief present day scenario with regard to the area of work. A brief discussion on how we got motivated to do the project work is also included. Thereafter, we will discuss about the objectives of our work which will be followed by discussion of target specifications. Finally, we will know about how the project report is organized.

1.2 Introduction to the area of work:

In this project, we worked in the area of artificial intelligence. Artificial intelligence is the method in which we use certain algorithms and data to provide machines with an ability to think like a human. With the aid of artificial intelligence, machines get the ability to analyze a problem and make decisions based on the analysis to solve the problem.

The sub-field of artificial intelligence in which we worked is machine learning. Machine learning enables a machine to execute tasks without being explicitly programmed by a human. As machine learning has its own set of sub-fields, the sub-field of machine learning in which we worked is deep learning. Deep learning is that field of machine learning which enables a machine to mimic the human brain with regard to data processing and making patterns that helps in the process of decision making.

1.3 Brief present day scenario with regard to the area of work:

Today, the entire world is using deep learning extensively in order to solve problems in a variety of domain. Deep learning is used to determine which ads must be shown to the user based on their search history in real time. It is used to predict the stock prices based on a variety of factors which is also our area of interest. It is also used to build systems for self-driving cars. It is also used by credit card companies to detect fraud. In fact, deep learning is used in almost every field that needs to be automated. Looking at the use cases of deep learning at present day, it appears that deep learning is going to be the backbone of many future technologies.

1.4 Motivation to do the project work:

In order to predict stock prices, many algorithms are in use today. Some of the major algorithms are SVM (Support Vector Machines) and Back Propagation. But after a detailed analysis of the results that these algorithms produced, we found that they were not very much accurate. Therefore, we decided to use another algorithm which is LSTM (Long Short Term Memory) that could solve the same problem but with much greater accuracy.

A brief importance of our work can be found by a brief analysis of the topic. Stock price prediction is the method to determine the future value of a stock based on several parameters. Correct predictions about the stock prices help us in making significant monetary profits. But, determining the future value of a stock is considered to be a tough job as it involves the detailed study of several parameters and instruments involved in the stock market. We need to invest a lot of time to study the stock markets in order to make correct predictions. By building a model for the same job, we take away the burden of studying the markets from humans and give it to machines which in turn save us a lot of time.

The methodology that we used in our project is quite unique as we used special kinds of RNN (Recurrent Neural Networks) known as LSTM. LSTM is different in the sense that is solves the problem of long term dependencies that was a major problem with standard RNN's.

The end result is very much significant as it will help us in determining the future price of a stock so that we can decide which stocks we need to invest in to earn good profits out of the market. And as all the predictions are made by machines themselves without any human effort that eventually saves us a lot of time.

1.5 Objective of the work:

The main objective of our work was to build an LSTM model that accepts the day closing price of a stock as an input and processes the input to make predictions on the future price of the stock.

In order to achieve the primary objective, we first had to collect the data and then process the data to make it into a valid input for the model. We also split the data for training and testing purpose. We then trained our model with the training data which enabled our model to predict the price of a stock.

1.6 Importance of the end result:

The importance of the end result can be determined from the fact that our model is able to predict the price of a stock by taking into account several parameters. Thus, humans no

longer need to perform a detailed study on the market to determine future price of a stock which in turn saves a lot of time. It also increases the profits made from the stock market significantly.

1.7 Organisation of the project report:

The project report is further organised into the following chapters:

- Literature Review
- Methodology
- Result analysis
- Conclusion and future scope

CHAPTER 2
LITERATURE REVIEW

2.1 Introduction:

In this chapter, we will first briefly discuss about the title of the project. We will then have a look at the literature review. Under literature review, we will have a look on the present state and recent developments in the work area which will be followed by a brief discussion on background theory. We will then discuss about literature survey. Thereafter, we will have a look on the summarized outcome of the literature survey. This will followed by a theoretical discussion. We will then perform a general analysis and lastly we will lastly have a look on the mathematical derivations.

2.2 Introduction to the project title:

The title of our project is 'Stock price prediction'. Stock price prediction is the method to determine the future price of a stock based on several parameters such as closing price, opening price, momentum, volatility etc. of a stock. In order to predict the price of a stock, we built an LSTM model that can give us predictions with good accuracy. This prediction is made by taking into account the previous prices of a stock to determine its future price. Stock price prediction is very helpful in the sense it helps us in making significant profits from the stock market.

2.3 Literature review:

There have been many recent developments in the area of deep learning. We will be discussing a few them.

- LSTM (Long Short Term Memory) and GRU (Gated Recurrent Units) are two enhanced RNN models. Despite RNN's being powerful, they are not good at handling long range sequence of data due to the vanishing gradient problem. In order to solve these issues, LSTM and GRU use gate units in order to decide which information is to be kept and which is to be removed from the previous state.
- LSTM and CNN models are being used for prediction of DNA protein binding sites in DNA sequence.
- Recent deep learning based approaches are being used to remove rain streaks from an input image.

Deep learning is a sub-field of machine learning that uses algorithms to mimic the working of the human brain in order to process data and creating patterns in the data that further helps in the process of decision making. The key factor in case of deep learning is its ability to learn without any human supervision and it can draw from data that is unlabeled. In order to learn, deep learning uses a hierarchical level of ANN (Artificial Neural Networks). In deep learning, neural codes are linked together, similar to the human brain.

2.4 Summarized outcome of the literature review:

Deep learning is an ever evolving field of machine learning as innovations are made every day in order to improve the field. The shortcomings of standard RNN's not being able to handle long data sequence were solved by the introduction of LSTM and GRU. Since then, LSTM have been widely used in various fields including medical, image processing etc. Deep learning overcomes some of the limitations of machine learning through its ability to draw from data which that is unlabeled.

2.5 Theoretical discussions:

Deep learning is a type of machine learning algorithm that uses more than one layer to extract higher-level features from the raw input data. Most of the deep learning models are based on ANN (Artificial Neural Networks). In deep learning, each level transforms its input data into more abstract and composite representation. The word deep learning is in reference to the number of layers through which the data is being transformed. Deep learning architectures are made using a greedy layer by layer method. Deep learning algorithms can be applied to unsupervised learning tasks. This gives us a huge benefit as unlabeled data is in more abundance as compared to labelled data.

2.6 General analysis:

On analysis of deep learning, it is found to be an extremely beneficial tool that can imitate the human brain in terms of data processing and extraction of features from raw data. The same cannot be achieved using machine learning models. Deep learning models can improve themselves through constant feedback without the intervention from human. Deep learning models are the best choice when it comes to unstructured or unlabeled data.

CHAPTER 3
METHODOLOGY

3.1 Introduction

In this chapter we are going to discuss how we implemented all the methods for achieving the desired result, the methods which we're going to discuss are data collection, data pre-processing, model making, tuning the model, predicting the results and forecasting for a particular range of time.

3.2 Detailed Methodology

For this project we've used the past data of the stock as our factor, we took last 5 years of stock's performance data. The data was collected from yahoo finance's website (https://in.finance.yahoo.com/), it is a trusted site and the data is easily available and can be directly downloaded in .csv format.

After downloading the data we cleaned the data i.e., the data contained some null values and if raw data was fed in the model then it could either throw an error or the model wouldn't had trained efficiently. So, the rows containing null values were deleted and data was cleaned.

After cleaning the data, the thing we did was pre-processed the data i.e., we took only the closing prices column and then splitted it into two subgroups training data and testing data. Training data contained the data which we were going to use for training our model and the testing data contained the data upon which we were going to test our model. We took 67% of the data as our training data and rest for testing.

This data was still not ready to be served in the model as our model expected time series data, so we converted the linear data into time series data.

After all the data pre-processing now we started to make our model, we made the model using tensorflow library present in python and used LSTM layers for making the model.

After creating the model we trained it on our training model for a particular number of epochs and after that checked the model's accuracy for ensuring it efficiently trained. If the model's performance was not up to the mark then we had done some hyperparameter tuning where we tuned some parameters such as number of epochs, number of LSTM layers, number of nodes present in the layers etc.

After all this our model was ready to be predicted, so we fed it with our testing data and predicted whether it yielded satisfying results or not. After the prediction work we then

moved on to the forecasting part where the model future forecast performance for the chosen stock.

3.3 Model architecture

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 30, 20)            1760
_________________________________________________________________
lstm_1 (LSTM)                (None, 30, 15)            2160
_________________________________________________________________
lstm_2 (LSTM)                (None, 10)                1040
_________________________________________________________________
dense (Dense)                (None, 1)                 11
=================================================================
Total params: 4,971
Trainable params: 4,971
Non-trainable params: 0
```

Fig.3.3.1 Model architecture

3.4 Block Diagram

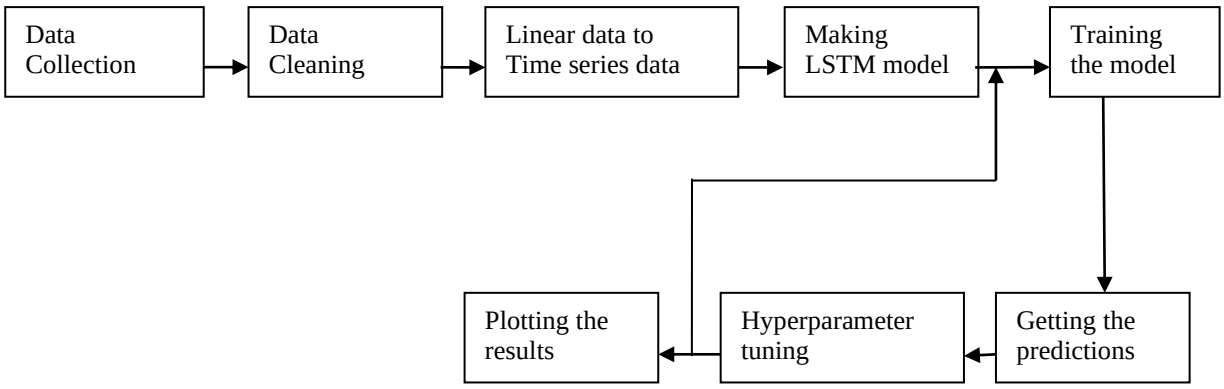

Fig 3.4.1 Block Diagram

3.5 Tools and libraries used

- Programming Language
    - Python
- IDE
    - Spyder
- For Data cleaning and Data preprocessing:
    - numpy (1.19.2) – NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a

large collection of high-level mathematical functions to operate on these arrays.

    - pandas (1.1.3) - Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series.
    - scikit_learn (0.24.2) - Scikit-learn is a free software machine learning library for the Python programming language. In this project it is used for data scaling.

- For making the model:
    - tensorflow (2.5.0) - TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks
    - keras (2.5.0) - Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library.
- For Data visualization:
    - plotly (5.1.0) - Plotly provides online graphing, analytics, and statistics tools for individuals and collaboration, as well as scientific graphing libraries for Python, R, MATLAB, Perl, Julia, Arduino, and REST.
- For the web app:
    - streamlit (0.83.0) - Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science

CHAPTER 4
RESULT ANALYSIS

4.1 Introduction

In this section we'll be discussing about the result we got after doing the project. We made the predictions for 5 public sector banks viz. Union Bank of India, Punjab National Bank, Bank of Baroda, Bank of India and State Bank of India.

The result analysis of the stocks is shown below:

4.2 Result Analysis

For State Bank of India:

We took 30 days stamp for arranging the original data into time series data. The model gave very nice result for this time series stamp. We made the model which consisted of 3 LSTM layers and one dense layer for getting the output, this model was trained for 100 epochs which in result yielded a decent forecast for next 30 days.

As we took 30 days as our time stamp, so we forecasted the prices for 30 days, we tried to forecast the prices for next 10, 60, 100 days but the result was not satisfactory as the model yielded some impractical results.

Moreover, if we wish to forecast the prices for next 60 or 100 days then we must train it for data using 60 or 100 time stamps respectively. The predicted and the forecasted plots are shown below:

4.3 Graphical representation of the result

- Prediction Graph

Fig 4.3.1 Prediction Graph

After training the model for 100 epochs we got the train error as 0.134 and the test error as 0.4721. This was a decent value because when we trained for a higher number of epochs then it showed some anomalous behaviour which must have been due to over fitting of the model.
In this plot:
Blue coloured line shows the values used for training the model.
Red coloured line shows predicted values of the stock.
Green coloured line shows actual values of the stock on the same date.

- Forecasted Graph

Fig 4.3.2 Forecasted Graph

This graph shows the forecasted result for future 30 days of the stock. As it can be seen from the plot the price first jumps above 450 and then starts to decrease till 400 at 30th day.

4.4 Significance of the result

The result we got for the forecast shows quite a similar trend which has happened in real stock market but the result was not as precise as in the real market, the reason could be that the model didn't got that much specific data to get trained and to achieve that precise results. The forecast can get more efficient if we take some more factors into consideration while training the model eg. Market sentiment analysis, News of the particular stock etc.

CHAPTER 5
CONCLUSIONAND FUTURE SCOPE OF WORK

5.1 Brief Summary of Work

The project was about the prediction of stock market using python. It was done by the help of Deep Learning algorithm which is LSTM networks.

For completing this project we used the idea of training the model with time series data. i.e., training the model with past days performance for predicting the output the very next day.

5.2 Conclusions

After completing the project we received quite impressive results of the forecasted price as it showed familiar trend of actual stock market but the backlash was it wasn't precise enough to make practical decisions out of it. The reason for this backlash must be lack of independent factors upon which the model has to be trained. The market sentiments, customers trust for the company, recent deals etc. plays a vital role in predicting the future prices of the stocks and these kind of data were missing in this project. But on the positive side, with only time series data we were successful to scrap some information for future prices of stocks which is very impressive at this stage.

Moreover, by completing this project we got much insights from the field of Artificial Intelligence, Stock Market etc. We came to know about many python libraries which were alien to us eg., plotly and streamlit.

5.3 Future Scope of the work

Predicting the stock market is a tuff job to do and we made our utmost effort to do so by time series analysis of past stock prices. But we also realised that only time series data analysis won't be enough for predicting the stock market, so, we plan to take this project a level up by integrating NLP (Natural Language Processing) algorithms for getting the sentiments of the market news, twitter trends etc and integrating it into this project and making it more capable to predict precise trends/prices of the stocks.

REFERENCES

Web

[1] Introduction, https://www.ibm.com/cloud/learn/deep-learning

[2] Literature Review, https://link.springer.com/article/10.1007/s13042-020-01096-5#Sec3

[3] Theoretical discussions, https://en.wikipedia.org/wiki/Deep_learning

[4] LSTM architecture, https://www.analyticsvidhya.com

[5] Time series forecasting, https://towardsdatascience.com

[6] Forecasting future prices, https://www.youtube.com/watch?v=H6du_pfuznE

[7] Time series prediction using deep learning, https://machinelearningmastery.com

ANNEXURES

- ## Project Code

```
import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)


df = pd.read_csv("../input/bank-dataset/SBIN_5Y.csv")
df.head()

df = df.drop(["Open", "Low", "Adj Close", "Volume"], axis=1)
df

df.isnull().sum()
df.shape

df = df.dropna()

df.isnull().sum()

df.shape
Date = df["Date"]
Date
sbi_df = df
close = df["Close"]
close.shape
close
close = close.dropna()
close.shape

from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler(feature_range=(0,1))
tmp = scaler.fit(np.array(close).reshape(-1,1))
new_df = scaler.transform(np.array(close).reshape(-1,1))


print(new_df)

training_size=int(len(new_df)*0.67)
test_size=len(new_df)-training_size
```

```
train_data,test_data=new_df[:training_size],new_df[training_size:]
Date_train, Date_test = Date[:training_size], Date[training_size:]
print(train_data.shape)
print(test_data.shape)
print(Date_train.shape)
print(Date_test.shape)

def create_dataset(dataset, time_step=1):
        dataX, dataY = [], []
        for i in range(len(dataset)-time_step-1):
                a = dataset[i:(i+time_step), 0]   ###i=0, 0,1,2,3-----99   100
                dataX.append(a)
                dataY.append(dataset[i + time_step, 0])
        return np.array(dataX), np.array(dataY)

n_steps = 30

time_step=n_steps
X_train, Y_train = create_dataset(train_data, time_step)
X_test, Y_test = create_dataset(test_data, time_step)
print(X_train.shape, Y_train.shape, X_test.shape, Y_test.shape)
#print(X_train, Y_train)

X_train =X_train.reshape(X_train.shape[0],X_train.shape[1] , 1)
X_test = X_test.reshape(X_test.shape[0],X_test.shape[1] , 1)
print(X_train.shape, X_test.shape)

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout
from tensorflow.keras.layers import LSTM

model=Sequential()
#model.add(LSTM(60,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2]),
return_sequences=True))
#model.add(LSTM(20,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2]),
return_sequences=True))
```

```
model.add(LSTM(20,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2]),
return_sequences=True))
#model.add(LSTM(20,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2]),
return_sequences=True))
model.add(LSTM(15,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2]),
return_sequences=True))
model.add(LSTM(10,
activation='relu',input_shape=(X_train.shape[1],X_train.shape[2])))
model.add(Dense(1))
model.compile(loss='mean_squared_error',optimizer='adam')

model.summary()

model.fit(X_train,Y_train,validation_data=(X_test,
Y_test),epochs=100,batch_size=32,verbose=1)

train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
print(train_predict.shape, test_predict.shape)

from sklearn.metrics import mean_squared_error
print(f'Train error - {mean_squared_error(train_predict, Y_train)*100}')
print(f'Test error - {mean_squared_error(test_predict, Y_test)*100}')

train_predict=scaler.inverse_transform(train_predict)
test_predict=scaler.inverse_transform(test_predict)
#print(test_predict)

import plotly.graph_objects as go

X_train=X_train.reshape(-1, 1)
X_test=X_test.reshape(-1, 1)
close_train=scaler.inverse_transform(train_data)
close_test=scaler.inverse_transform(test_data)
```

```
close_train = close_train.reshape(-1)
close_test = close_test.reshape(-1)
prediction = test_predict.reshape((-1))

trace1 = go.Scatter(
    x = Date_train,
    y = close_train,
    mode = 'lines',
    name = 'Data'
)
trace2 = go.Scatter(
    x = Date_test[n_steps:],
    y = prediction,
    mode = 'lines',
    name = 'Prediction'
)
trace3 = go.Scatter(
    x = Date_test,
    y = close_test,
    mode='lines',
    name = 'Ground Truth'
)
layout = go.Layout(
    title = "Not Google Stock",
    xaxis = {'title' : "Date"},
    yaxis = {'title' : "Close"}
)
fig = go.Figure(data=[trace1, trace2, trace3], layout=layout)
fig.show()

test_data = close
test_data = scaler.transform(np.array(close).reshape(-1,1))
test_data = test_data.reshape((-1))

def predict(num_prediction, model):
    prediction_list = test_data[-n_steps:]

    for _ in range(num_prediction):
        x = prediction_list[-n_steps:]
        x = x.reshape((1, n_steps, 1))
```

```
        out = model.predict(x)[0][0]
        prediction_list = np.append(prediction_list, out)
    prediction_list = prediction_list[n_steps-1:]

    return prediction_list

def predict_dates(num_prediction):
    last_date = df['Date'].values[-1]
    prediction_dates = pd.date_range(last_date, periods=num_prediction+1).tolist()
    return prediction_dates

num_prediction =30
forecast = predict(num_prediction, model)
forecast_dates = predict_dates(num_prediction)

forecast = forecast.reshape(1, -1)
forecast = scaler.inverse_transform(forecast)

test_data = test_data.reshape(1, -1)
test_data = scaler.inverse_transform(test_data)

test_data = test_data.reshape(-1)
forecast = forecast.reshape(-1)

forecast
date = df["Date"]
trace1 = go.Scatter(
    x = date,
    y = test_data,
    mode = 'lines',
    name = 'Data'
)
trace2 = go.Scatter(
    x = forecast_dates,
    y = forecast,
    mode = 'lines',
    name = 'Prediction'
)
```

```
fig = go.Figure(data=[trace1, trace2], layout=layout)
fig.show()

model.save("BankOfBaroda.h5")
```