Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

Stock Performance and Price Prediction using Machine Learning

Prof. Mainuck Das, Swagata Biswas, Vikrant Banerjee, Shamik Banerjee, Sudip Kr. Pal, Subhrashis Mallick.
Vikrant Banerjee.
Department of ECE, JIS College of Engineering, Kalyani, West Bengal.

***Abstract*: The stock market is a steady & long-term way of growing your money exponentially. But there do exists some steps that could destroy someone financially. To avoid mishaps, we need to think twice before making a financial decision. For example, whenever the stock prices are in the dip, we should buy them. Whenever the stock prices are in the overbought zone or hits the upper circuit, we should sell them to make a profit. If we correctly predict the time to trade the stocks, we can earn a profit by large margins. We can make use of the latest technology of Machine Learning to predict stock prices to aid in equity investing. Using the LSTM (Long Short-Term Memory) method can be helpful in prediction of stocks. It uses previous stock opening and closing price to analyse the trend and predict the future outcome. Here the API of yahoo finance is used as to get a very authentic & close prediction of the model with the most accuracy possible**

***Keywords*: Machine Learning, Python, Stock market, Prediction, Numpy, Pandas, Matplotlib, Plotly, Tensorflow.**

Overview of Technology: The purpose of this project report is to provide an overview of the development and implementation of a stock performance and price prediction model using machine learning techniques. The model uses Python libraries such as Pandas, NumPy, Matplotlib, Scikit-learn, and TensorFlow to analyze historical stock data and make predictions on future stock performance.

1. *NumPy*: 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. It also has functions for working in domain of linear algebra, Fourier transform, and matrices.

2. *Matplotlib*: It is a plotting library for the Python programming language. It is used for creating static, animated and interactive visualizations in Python. Matplotlib makes easy things easy and hard things possible. Create publication quality plots. Make interactive figures that can zoom, pan, update.

3. *Pandas*: Pandas is an open-source Python package that is most widely used for data science/data analysis and machine learning tasks. It is built on top of another package named NumPy, which provides support for multidimensional arrays.

4. *Datetime:* Python Datetime module supplies classes to work with date and time. These classes provide a number of functions to deal with dates, times and time intervals. Date and datetime are an object in Python, so when you manipulate them, you are actually manipulating objects and not string or timestamps.

5. *Scikit-learn*: Scikit-learn is probably the most useful library for machine learning in Python. The sklearn library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

6. TensorFlow: TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining. Using production level tools to automate and track model training over the lifetime of a product, service, or business process is critical to success.

INTRODUCTION

Every now & then we hear people (especially) the Gen Z & the millennials about trading & investing in the stock market, noticeably the US Stock exchanges, because of the current USD-INR exchange rates. But rather they have little knowledge in terms of the nitty gritty of the market fundamentals & how it behaves. Also, they're not really

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

researching on the fundamentals & rather investing on FOMO, or fear of missing out. It's a psychological phenomenon in which people trend to do something that they're either not interested in, or they have little to no knowledge on what they're doing, as subjects around them do the same thus creating an aura of social pressure. This project is our little effort to build a ML based model to predict the performance of a stock of the NYSE by analysing it's previous performances and giving an overview of how the stock will perform in long-mid-term or very long-term period of time. This in turn will help the millennials to make an investing decision rather than to just invest in FOMO, without a single valid point.

OBJECTIVES

The main objectives of the project are as follows:

- Develop a model that can accurately predict future stock performance and prices.
- Analyse historical stock data to identify trends and patterns that can inform predictions.
- Evaluate the performance of the model and compare it to traditional stock analysis methods.
- Provide insights and recommendations to assist investors in making informed investment decisions.

METHODOLOGY

The development of the stock performance and price prediction model involved the following steps:

1. *Data Collection*: Historical stock market data has been collected from reliable and reputable sources. This data includes variables such as stock prices, trading volume, market indexes, and news sentiment.

2. *Data Preprocessing*: Collected data is pre-processed using Pandas and NumPy. This step involves cleaning the data, dealing with missing values, normalizing numeric features, and encoding categorical variables as needed.

3. *Feature Engineering*: Additional features are taken from existing data to improve the predictive power of the model. This process involves the calculation of technical indicators, such as moving averages, relative strength index (RSI), and Bollinger bands.

4. *Model Development:* A machine learning model based on TensorFlow and Scikit-learning has been developed to predict stock price and performance. Various algorithms, such as Random Forest, Gradient Enhancement, and Neural Networks, have been explored and evaluated to determine the best performing model.

5. *Model Training and Evaluation*: Historical data has been divided into training and test sets. The model is trained on the training set and evaluated on the test set. Evaluation measures such as root mean square error (MSE), root mean square error (RMSE) and accuracy were used to evaluate the performance of the model.

6. *Deployment and Integration*: The trained model has been integrated into the user-friendly interface, allowing the user to enter relevant parameters, such as the stock symbol and the predicted horizon, and get the performance and predicted stock prices based on the trained model.

- *Software workflow:*

1. Information is extricated from Yahoo back API utilizing web scratching utilizing data-reader of Pandas. We utilize a variable to store the company code and after that continue to gather its stock opening and closing costs for a certain period.

2. Information is at that point cleaned and organized utilizing MinMaxScaler of the sklearn bundle. It is at that point put away in cluster arrange utilizing NumPy.

3. The demonstrate is built by utilizing the LSTM calculation from TensorFlow bundle.

4. The demonstrate is at that point prepared through age and cruel squared mistake is assessed.

5. After that, the anticipated information is compared to genuine information. Genuine information is scratched from Yahoo fund API and at that point compared to the anticipated cost.

6. The chart of genuine vs. anticipated information is plotted utilizing the pyplot module of the matplotlib bundle.

ALGORITHMS USED

We've used many algorithms to find out the best possible prediction, i.e. the max RSME & MSE score in order to achieve best results. Below are the listings of the algorithms used and their respective scores.

- *Linear Regression* – It's one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression makes predictions for continuous/real or numeric variables such as sales, salary, age, product price, etc.

$$y = a_0 + a_{1x} + \varepsilon$$

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

Y= Dependent Variable (Target Variable)
X= Independent Variable (predictor Variable)
a0= intercept of the line (Gives an additional degree of freedom)
a1 = Linear regression coefficient (scale factor to each input value).
ε = random error

[The values for x and y variables are training datasets for Linear Regression model representation.]

RSME for this model = 0.258976390 (least)

- *Random forest* – It's a popular algorithm when it comes to ML, is a non linear and supervised learning technique. As the name suggests, "Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. *The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.*

$$RFfi_i = (\sum_{j \in all\ trees} normfi_{ij}) \ / \ T$$

- RFfi sub(i) – the importance of feature i calculated from all trees in the random forest model.
- Normfi sub(ij) – the normalized feature importance for I in tree j.
- T – total number of trees.

RSME for this model = 0.628922109 (best)

- *KNN (K – Nearest Neighbors) - K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.*

It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

RSME for this model = 0.498117645 (moderate)

SOFTWARE USED

- *Visual Studio Code:* Visual Studio Code is a free coding editor that helps you start coding quickly. Use it to code in any programming language, without switching editors. Visual Studio Code has support for many languages, including Python. It is made by Microsoft with the Electron Framework, for Windows, Linux, and macOS.

- *Anaconda Navigator:* Anaconda is an open-source distribution for python and R. It is used for data science, machine learning, deep learning, etc. With the availability of more than 300 libraries for data science, it becomes fairly optimal for any programmer to work on anaconda for data science.

- *Jupyter Notebook:* The Jupyter Notebook is a web-based interactive computing platform. A Jupyter notebook has two components: a front-end web page and a back-end kernel. The front-end web page allows data scientists to enter programming code or text in rectangular "cells." The browser then passes the code to the back-end kernel which runs the code and returns the results.

RESULTS

Using a sizable dataset, the effectiveness of the created stock performance and price prediction model was assessed. In comparison to conventional analysis methods, the model produced good findings with a high level of accuracy in predicting stock performance and pricing. The promise of machine learning in stock research and prediction can be shown in the model's ability to spot trends and patterns in the data that were not obvious using more conventional analysis techniques.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

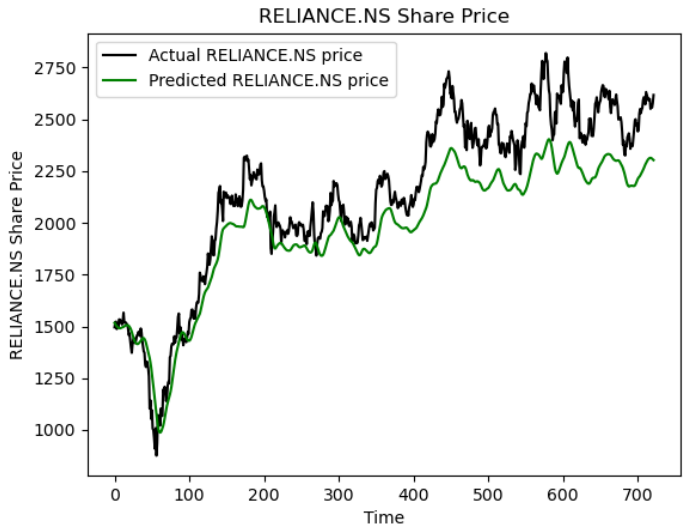


Fig.1 – Reliance (NSE) fetched from Yahoo finance API

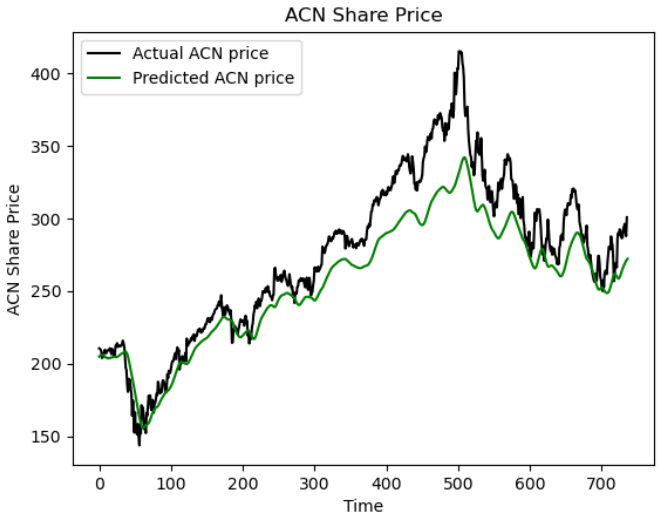
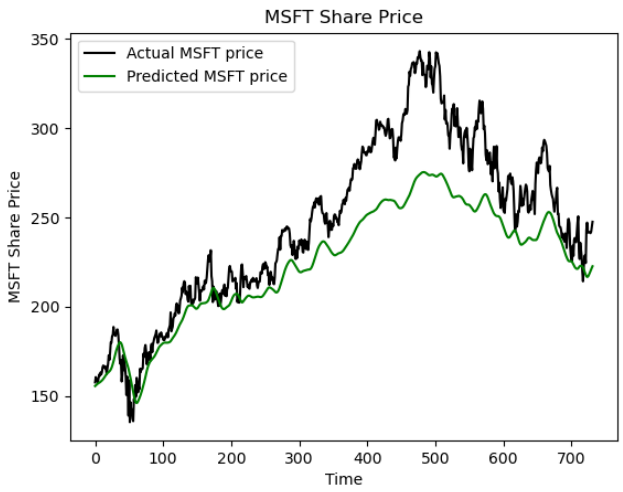


Fig.2 – Accenture (NYSE) fetched from Yahoo finance API



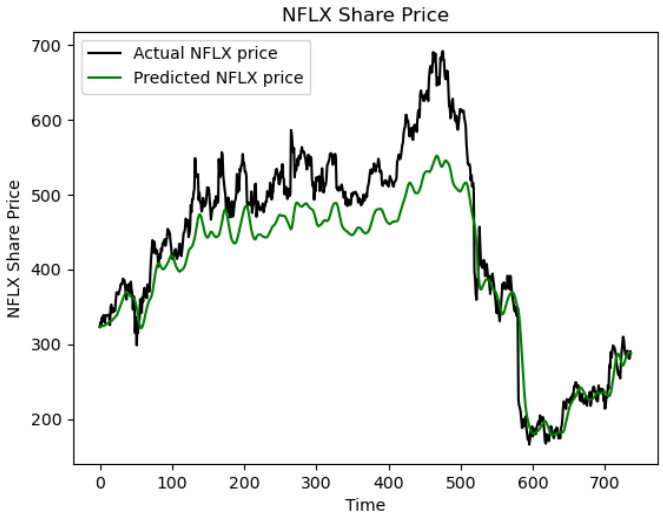


Fig. 3 – Netflix (NYSE) fetched from Yahoo finance API

Fig. 4 – Microsoft (NYSE) fetched from Yahoo finance AP

PROJECT SOURCE CODE

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import pandas_datareader as web
import datetime as dt

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

company='FB'
```

```
start=dt.datetime(2012,1,1)
end=dt.datetime(2020,1,1)
data=web.DataReader(company,'yahoo',start,end)

# Preparing Data
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)):
```

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

```
        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))

# Building the model
model=Sequential()
model.add(LSTM(units=50,return_sequences=True,input_s
hape=(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))

#Prediction of next closing value
model.compile(optimizer='adam',loss='mean_squared_error'
)
model.fit(x_train,y_train,epochs=25,batch_size=32)

''' Testing the model accuracy on existing data '''
# Loading Test Data
test_start=dt.datetime(2020,1,1)
test_end=dt.datetime.now()
test_data=web.DataReader(company,'yahoo',test_start,test_e
nd)

actual_prices=test_data['Close'].values
total_dataset=pd.concat((data['Close'],test_data['Close']),axi
s=0)
model_inputs=total_dataset[len(total_dataset)-
len(test_data)-prediction_da
ys:].values
model_inputs=model_inputs.reshape(-1,1)
model_inputs=scaler.transform(model_inputs)

# Making predictions on Test Data
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))
predicted_prices=model.predict(x_test)
predicted_prices=scaler.inverse_transform(predicted_prices)

# Plotting Test Predictions
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()
```

CONCLUSION

The ability of machine learning to forecast stock prices is illustrated by the stock performance and price prediction model created utilizing these techniques. Investors may make wise investing selections thanks to the model's useful insights and forecasts. It is crucial to keep in mind that predicting the stock market is difficult by nature and is impacted by a number of external factors.

This is often a show based on the most recent innovation of Machine learning for foreseeing the execution of a stock of the NYSE/NSE/BSE by analyzing it's past exhibitions. The most objective of our thought and a bit of effort is to supply an diagram of exchanging and contributing within the stock showcase conjointly how it'll perform in long-mid-term or exceptionally long-term period to the millennials. As most of them are ignorant of nitty abrasive of the showcase basics and it's behaviour. Till presently there's no such arranging for commercialization, as right now we are incapable to anticipate future costs. After making a show which is able have the determination of future cost forecast effectively and in case that will be vigorously sufficient, at that point we'll be applying for obvious and giving it in rent or deal for a Stock broking company.

FUTURE WORK

Future work can focus on the following areas:

- Incorporating additional features, such as news sentiment and social media data, to enhance the predictive power of the model.
- Implementing advanced deep learning techniques to further improve the accuracy of the model.
- Integrating into real life scenarios.
- Currently this particular model is only capable of predicting present prices, hereinafter we are about to design a model which would be capable of long-term prediction of the nearing few years as well, as short-term predictions are really risky & almost impossible to pinpoint any change in the stock market.
- We will be checking onto the price fluctuations in the forthcoming years by taking the data of past 20 years approximately & the term period of prediction will be large, i.e. 10 to 15 years.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 12 Issue 05, May-2023

ACKNOWLEDGMENTS

The authors are grateful to the authority of JIS College of Engineering to encourage them to perform the research and development.

We would also like to thank our project mentor Mr. Mainuck Das for giving his valuable inputs and guiding us how to improve the project. We would like to thank our digital library archive for providing resources.

Finally, we would like to thank our co-companions for supporting us.

REFERENCES

[1] https://www.geeksforgeeks.org/stock-price-prediction-using-machine-learning-in-python/
[2] https://www.analyticsvidhya.com/blog/2021/10/machine-learning-for-stock-market-prediction-with-step-by-step-implementation/
[3] https://www.datacamp.com/tutorial/lstm-python-stock-market
[4] https://www.javatpoint.com/machine-learning-random-forest-algorithm
[5] https://www.analyticsvidhya.com/blog/2022/01/the-complete-lstm-tutorial-with-implementation/
[6] https://ieeexplore.ieee.org/abstract/document/10061120
[7] The Python Bible Volume 5: Python For Finance (Stock Analysis, Trading, Share Prices) by Florian Dedov
[8] Python for Finance: Mastering Data-Driven Finance by Yves Hilpisch.