Stock Price Prediction

by

Prithviraj Guntha	20BRS1188
Je Sai Kailash Pulipati	20BRS1208
Ganti Venkata Varshini	20BRS1046

A project report submitted to

Dr. PATTABIRAMAN V

SCHOOL OF COMPUTER SCIENCE

in partial fulfilment of the requirements for the course of

CSE3105 – Machine learning and its applications

in

B. Tech. COMPUTER SCIENCE ENGEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

NOVEMBER 2022

Stock Price Prediction

Prithviraj Guntha 20BRS1188, Je Sai Kailash Pulipati 20BRS1208 and Ganti Venkata Varshini 20BRS1046

Abstract: In recent years, we see a lot of people investing into various stocks and this growth is said to be exponential in the coming years as well. But people have to be careful in what they invest and on how much they invest as the stock market is a high risk and high reward field. So understanding this, we have decided to create a model that could predict the future stock price which could help people mitigate losses and have a better chance of earning profits. In this report, we have first tested various models like K- Nearest Neighbours(KNN), moving average, Linear Regression and Long Short Term Memory(LSTM) to understand how they respond to the data about any particular company that is listed on NSE India. We then selected the model that produced the best accuracy and used that algorithm to find the price of any stock for the next 30 days. Since this is a software tool, we have also created a website using Django so that users can log in and check for any stock prices and predictions for that company which they are interested.

Keywords: Machine learning, Stock market, Finance, KNN, moving average, LSTM, linear regression, time series analysis.

1. INTRODUCTION

A stock market is a public market where you can buy and sell shares for publicly listed companies. The stocks, also known as equities, represent ownership in the company. The stock exchange is the mediator that allows the buying and selling of shares. Stock market helps companies raise capital and also help generate personal wealth by building savings and protecting people's money from inflation and taxes by maximise income[1]. Stock markets serve as an indicator of the state of the economy. Hence, we can understand how important stock market is for a country. But stock market is not a steady income field, it has a lot of risks attached to it. People can loose all their savings if they do not invest properly or don't do any background study before they invest into any company. So how can we make sure that we invest in the right companies? We can now introduce the term machine learning, Machine learning is the field of science that fuses robotics, statistics and informatics. It is a part of scientific studies that deal with artificial intelligence (AI). The primary purpose of machine learning is to create an automatic system that can continuously improve itself without human interference. The only thing it uses is collected data that is analyzed by it to find specific patterns impossible to spot by people[2]. In financial markets, a machine learning (ML) has become a powerful analytical tool used to help and manage investment efficiently. ML has been widely used in the financial sector to provide a new mechanism that can help investors make better decisions in both investment and management to achieve better performance of their securities investment[3]. Hence, we have decided to build a model that would help predict the future price of any company so that people can have a better chance of getting profits and mitigating losses. In this report, we have taken four different models; K-nearest neighbours, linear regression, moving average and Long Short term memory algorithms and checked which algorithm gives a better accuracy for prediction. It was found that LSTM gave a better accuracy and that model was used to predict the stock price for the next 30 days for any particular company that is listed in NSE India. We have also created a website so that users can log in and select any company they want and find the approximate price for the next 30 days. This can help people by helping them make profits as well as ensure that they don’t invest in companies that have the potential to give losses.

2. LITERATURE REVIEW

What people think has constantly been an essential piece of records for maximum folks all through the decision-making method. The technology have now (amongst different things) made it viable to discover approximately the reviews and stories of these withinside the substantial pool of humans which are neither our non-public acquaintances nor famous expert critics — this is, humans we've got in no way heard of. And conversely, increasingly more humans are making their reviews to be had to strangers thru the Internet. The hobby that person customers display in on line reviews approximately merchandise and services, and the capacity have an impact on such reviews wield, is something this is riding force for this vicinity of hobby. And there are numerous demanding situations concerned on this method which needs to be walked throughout in an effort to gain right consequences out of them. In this survey we analysed simple technique that generally occurs on this method and measures which are to be taken to conquer the demanding situations being faced.

The preliminary attention of our literature survey became to explore established on-line mastering algorithms and spot if they might be tailored to our use case i.e., operating on real-time inventory rate data. These protected Online AUC Maximization, Online Transfer Learning, and Online Feature Selection. However, as we had been not able to discover any ability adaptation of those for inventory rate prediction, we then determined to appearance at the present systems, examine the principal drawbacks of the same, and spot if we should enhance upon them. We zeroed in at the correlation among inventory data (withinside the shape of dynamic, long-time period temporal dependencies among inventory prices) as the important thing difficulty that we wanted to solve. A brief seek of established answers to the above trouble led us to RNN's and LSTM. After identifying to apply an LSTM neural community to carry out inventory prediction, we consulted a wide variety of papers to observe the idea of gradient descent and its numerous types. We concluded our literature survey by searching at how gradient descent may be used to song the weights of an LSTM community and the way this method may be optimized.

In the finance international stock trading is one of the maximum essential activities. Stock marketplace prediction is an act of looking to decide the destiny fee of an inventory different economic device traded on a economic exchange. This paper explains the prediction of an inventory the usage of Machine Learning. The technical and essential or the time collection evaluation is utilized by the maximum of the stockbrokers whilst making the inventory predictions. The programming language is used to are expecting the inventory marketplace the usage of gadget mastering is Python. In this paper we recommend a Machine Learning (ML) technique to be able to gain knowledge of from the to be had shares records and benefit intelligence after which makes use of the received understanding for a correct prediction. In this context this study uses a machine learning technique called LSTM (Long short-term memory) to predict stock prices for the large and small capitalizations.

Existing Method

Stock Market prediction using AI techniques

The susceptible shape of Efficient Market hypothesis (EMH) states that it's far not possible to forecast the destiny charge of an asset primarily based totally at the data contained withinside the historic charges of an asset. This way that the marketplace behaves as a random stroll and as a end result makes forecasting not possible. Furthermore, economic forecasting is a tough venture because of the intrinsic complexity of the economic machine. The goal of this paintings become to apply synthetic intelligence (AI) strategies to version and are expecting the destiny charge of a inventory marketplace index. Three synthetic intelligence strategies, namely, neural networks (NN), guide vector machines and neuro-fuzzy structures are carried out in forecasting the destiny charge of a inventory marketplace index primarily based totally on its historic charge data. Artificial intelligence strategies have the cap potential to think about economic machine complexities and they're used as economic time collection forecasting tools.

Two strategies are used to benchmark the AI strategies, namely, Autoregressive Moving Average (ARMA) that's linear modelling method and random stroll (RW) method. The experimentation became completed on facts received from the Johannesburg Stock Exchange. The facts used became a chain of beyond last expenses of the All-Share Index. The consequences confirmed that the 3 strategies have the capacity to are expecting the destiny fee of the Index with an appropriate accuracy. All 3 synthetic intelligence strategies outperformed the linear model. However, the random stroll technique out completed all of the different strategies. These strategies display an capacity to are expecting the destiny fee however, due to the transaction expenses of buying and selling withinside the marketplace, it isn't feasible to expose that the 3 strategies can disprove the vulnerable shape of marketplace efficiency. The consequences display that the rating of performances guide vector machines, neuro-fuzzy systems, multilayer perceptron neural networks is depending on the accuracy degree used.

Stock market prediction in India using Artificial Neural Networks

A stock marketplace is a platform for buying and selling of a company's shares and derivatives at an agreed fee. Supply and call for of stocks power the inventory marketplace. In any us of a inventory marketplace is one of the maximum rising sectors. Nowadays, many humans are in a roundabout way or without delay associated with this sector. Therefore, it will become crucial to recognize approximately marketplace trends. Thus, with the improvement of the inventory marketplace, humans are interested by forecasting inventory fee. But, because of dynamic nature and at risk of brief modifications in inventory fee, prediction of the inventory fee will become a hard task. Stock m Prior paintings has proposed powerful techniques to study occasion representations that could seize syntactic and semantic facts over textual content corpus, demonstrating their effectiveness for downstream obligations together with script occasion prediction. On the alternative hand, occasions extracted from uncooked texts lacks of common-experience information, together with the intents and feelings of the occasion participants, which can be beneficial for distinguishing occasion pairs while there are best diffused variations of their floor realizations. To cope with this issue, this paper proposes to leverage outside common-experience information approximately the purpose and sentiment of the occasion.

Experiments on 3 occasion-associated tasks, i.e., occasion similarity, script occasion prediction and inventory marketplace prediction, display that our version obtains lots higher occasion embeddings for the tasks, reaching 78% enhancements on difficult similarity task, yielding greater specific inferences on next activities beneath given contexts, and higher accuracies in predicting the volatilities of the inventory market1. Markets are by and large a non-parametric, non-linear, noisy and deterministic chaotic system. As the era is increasing, inventory buyers are shifting in the direction of to apply Intelligent Trading Systems in place of essential evaluation for predicting costs of stocks, which enables them to take instant funding decisions. One of the primary ambitions of a dealer is to expect the inventory rate such that he can promote it earlier than its fee decline, or purchase the inventory earlier than the rate rises. The green marketplace speculation states that it isn't always feasible to expect inventory costs and that inventory behaves withinside the random walk. It appears to be very hard to update the professionalism of an skilled dealer for predicting the inventory rate. But due to the provision of a exquisite quantity of information and technological improvements we are able to now formulate the proper set of rules for prediction whose outcomes can boom the earnings for buyers or funding firms. Thus, the accuracy of an set of rules is without delay proportional to profits made with the aid of using the usage of the set of rules.

Increasing integration of European economic markets is probable to bring about even stronger correlation among fairness expenses in extraordinary European nations. This procedure also can result in convergence in financial improvement throughout European nations if tendencies in inventory markets impact actual financial components, inclusive of funding and consumption. Indeed, our vector autoregressive fashions advise that the effective correlation among adjustments fairness expenses and funding is, in general, significant. Hence, economic government need to display reactions of proportion expenses to economic coverage and their outcomes at the enterprise cycle.

Neural network approach for Stock market prediction

To expand a revolutionary neural community method to gain better inventory marketplace predictions. Data have been acquired from the stay inventory marketplace for real-time and off-line evaluation and effects of visualizations and analytics to illustrate Internet of Multimedia of Things for inventory evaluation. To look at the have an impact on of marketplace traits on inventory prices, conventional neural community algorithms may also incorrectly are expecting the inventory marketplace, for the reason that preliminary weight of the random choice hassle may be without problems liable to wrong predictions. Based at the improvement of phrase vector in deep learning, we exhibit the idea of “inventory vector.” The enter is not a unmarried index or unmarried inventory index, however multi-inventory high-dimensional historic data. We advocate the deep lengthy short-time period reminiscence neural community (LSTM) with embedded layer and the lengthy short-time period reminiscence neural community with computerized encoder to are expecting the inventory marketplace. In those fashions, we use the embedded layer and the automated encoder, respectively, to vectorize the data, in a bid to forecast the inventory thru lengthy short-time period reminiscence neural community. The experimental effects display that the deep LSTM with embedded layer is better. Specifically, the accuracy of fashions is 57.2% and 56.9%, respectively, for the Shanghai A-stocks composite index. Furthermore, they're 52.4% and 52.5%, respectively, for person stocks

An intelligent technique for Stock market prediction

A stock market is a unfastened community of financial transactions among consumers and dealers primarily based totally on shares additionally called stocks. In inventory markets, shares constitute the possession claims on businesses. These can also additionally encompass securities indexed on a inventory alternate in addition to the ones simplest traded privately. A inventory alternate is an area in which agents should purchase and/or promote shares, bonds, and different securities. Stock marketplace is a completely prone region for funding because of its unstable nature. In the close to past, we confronted big monetary issues because of big drop-in fee of stocks in inventory markets worldwide. This phenomenon delivered a heavy toll at the worldwide in addition to on our countrywide monetary structure. Many humans misplaced their final financial savings of cash at the inventory marketplace. In 2010–2011 monetary year, Bangladeshi inventory marketplace confronted big disintegrate. This phenomenon may be delivered below manage specifically through strict tracking and example inventory marketplace evaluation. If we will examine inventory marketplace successfully in time, it is able to end up a discipline of huge income and can end up relatively much less prone for the investors. Stock marketplace is all approximately prediction and fast choice making approximately funding, which can't be completed without thorough evaluation of the marketplace. If we will are expecting the inventory marketplace through analysing historic statistics properly, we will keep away from the outcomes of significant marketplace disintegrate and with a purpose to take vital steps to make marketplace proof against such situations.

3. ARCHITECTURE

We start the process[fig. 1], we first get the data about any stock from the NSE India website. In our case, we have taken the stock data of HDFC bank. Then we only take useful features like the date and Closing price each day. We then split the data into the training and valid datasets so that we can check the root mean squared error (RMSE) value obtained using different machine learning algorithms. We then apply the Moving average, linear regression, K-nearest neighbours (KNN) and Long Short Term Memory (LSTM). We get an RMSE value of 187.00 for moving average, 154.38 for linear regression, 678.88 for KNN and 58.09 for LSTM. We take the model with the least RMSE value which the LSTM model. We then scale the data by preparing it for prediction. We use LSTM for predicting the price for the next 30 days and plot the predictions so that user can infer from the graph and take decisions.

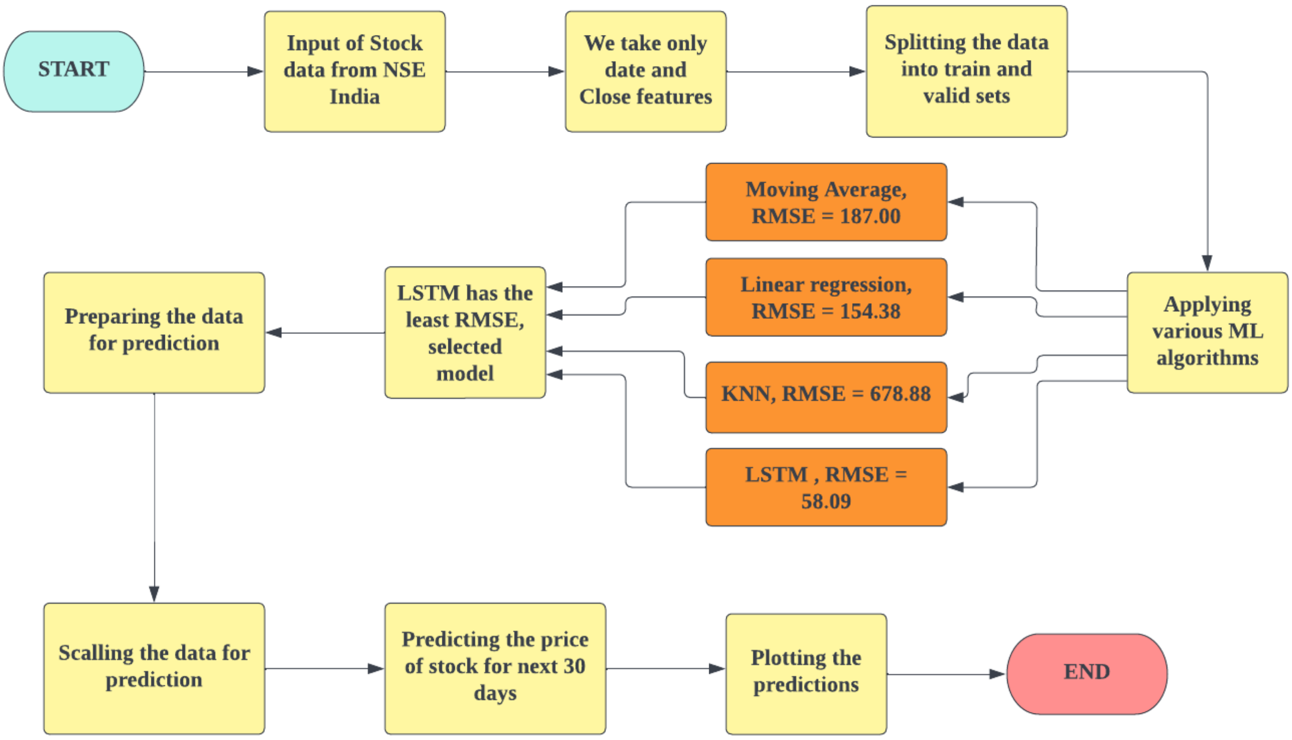

Fig 1. this diagram shows the entire process of the project.

4. MODEL IMPLEMENTATION

We have used four different models and compared them against each other to see which model gave the better accuracy and the least root mean squared error. We have applied K-nearest neighbour, moving average, linear regression and Long short term memory(LSTM) algorithms to the data.

5.1 Data used in this analysis

We have obtained stock data of different companies from the NSE India website. We have utilized a package available in python that extracts the data from this

Website and saves them as a table. We then use this table and store it as a CSV file so that we can use the pandas library against the data.

Since **nsepy** package is not pre installed into python, we have to install the package using the following command in the terminal:

```
!pip install nsepy
```

Sample data extraction,

```
from nsepy import get_history
from datetime import date
import pandas as pd
data = get_history(symbol="HDFC", start=date(2015,1,1), end=date(2020,1
,31))
data.to_csv("data.csv")
df = pd.read_csv("data.csv")
df.head()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2015-01-01	HDFC	EQ	1135.90	1130.0	1131.15	1120.10	1125.90	1124.00	1124.23	401576	4.514650e+13	11804	128793	0.3207
1	2015-01-02	HDFC	EQ	1124.00	1127.3	1176.95	1125.35	1171.05	1171.90	1159.93	2019816	2.342845e+14	59071	1258847	0.6232
2	2015-01-05	HDFC	EQ	1171.90	1168.8	1175.00	1150.40	1154.95	1156.40	1159.24	2219458	2.572880e+14	57749	1414720	0.6374
3	2015-01-06	HDFC	EQ	1156.40	1148.6	1148.60	1096.10	1098.00	1101.95	1117.84	2531748	2.830100e+14	81436	1547523	0.6112
4	2015-01-07	HDFC	EQ	1101.95	1097.5	1114.15	1095.00	1097.65	1099.25	1105.58	2406880	2.660992e+14	133109	1533304	0.6371

Fig 2. This image shows the data that has been obtained after running the code by getting data from NSE India.

for predicting the future price, we will use the Close variable and set the date variable as the index,

```
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']
data = df.sort_index(ascending=True, axis=0)
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'
])

for i in range(0,len(data)):
     new_data['Date'][i] = data['Date'][i]
     new_data['Close'][i] = data['Close'][i]
```

we will now split the data into the train and valid sets,

```
train = new_data[:900]
valid = new_data[900:]
```

5.2 Applying Moving average algorithm

Moving average is one of the most commonly used tools in stock market prediction. The predicted closing price for each day will be the average of a set of previously observed values. Instead of using the simple average, we will be using the moving average technique which uses the latest set of values for each prediction. In other words, for each subsequent step, the predicted values are taken into consideration while removing the oldest observed value from the set.

$$Simple\ Moving\ Average = (A_1 + A_2 + \ldots\ldots + A_n) / n \qquad (1)$$

In this above formula, A is the data point and n is the number of observations taken into consider.

We will use the above processed data as the input for this algorithm and the expected output would be the RMSE value and the appropriate prediction graph. The code for this implementation is given in the Appendix as well as the GitHub link that is provided below.

5.3 Applying Linear regression

Linear regression is one of the simplest algorithms that can be used. The main idea behind this algorithm is that it returns an equation that determines the relationship between the independent variable and the dependent variable. In our dataset, we do not have a proper independent variable, so we can extract information like the day, month or year and use it in our model.

The formula for slope is given as,

$$Y = mx + b \qquad (2)$$

$$b = ((\sum xy) + \sum x \sum y)/(\sum x^2 + (\sum x)^2) \qquad (3)$$

$$m = (1/n)(\sum y - b\sum x) \qquad (4)$$

we will use the above formula to compute the future stock data. The input will be the Stock data that's processed accordingly and the expected output will be the RMSE value and the prediction graph. The code for this implementation is given in the Appendix and given in the Github link that's provided.

5.4 K-Nearest Neighbours

KNN assumes the similarity between the new data and the already available data and put the new case into the category that is most similar to it. We calculate the distance between the test data and each row of the training data. We will use the Euclidean distance for measuring the distance between the two points. The k value will be the square root of the number of observations in the dataset.

The Euclidean distance is given as,

$$d = \sqrt{[(x_2 - x_1)^2 + (y_2 - y_1)^2]} \qquad (5)$$

The input will be the Stock data that's processed accordingly and the expected output will be the RMSE value and the prediction graph. The code for this implementation is given in the Appendix and given in the Github link that's provided.

5.5 Long Short-Term memory

LSTM is useful for time series analysis. Long Short Term Memory is a kind of recurrent neural network. LSTM works so well because it is able to store past information that is important forget the information that is not important. LSTM has 3 gates in its architecture; The **input gate** adds information to the cell state. The **forget gate** removes the information that is no longer required by the model. The **output gate** selects the information to be shown as output.

The input will be the Stock data that's processed accordingly and the expected output will be the RMSE value and the prediction graph. The code for this implementation is given in the Appendix and given in the Github link that's provided.

5. WEBSITE

We have also created a website for our model so that the user can easily get the prediction for any stock that he/she wants. We have created our website using the

Django framework using python, HTML and CSS. In our website we have a home page which is the main page, basically the first page when anyone opens our website. Then we have a contact page from where users can contact us. The next page will be the about us page where the users can get information about us. Then there will be a login page from which the users can login, after the user logs in they will have a stock option. After the user clicks that option, he/she will have a text box to enter the company for which the user wants the stock prediction. After the user confirms the selection, a graph with the prediction for the next 30 days will be displayed.

6. RESULT AND DISCUSSION

We have successfully implemented the four algorithms on our data and found the RMSE value and the prediction graphs.

Table 1: RMSE values obtained from different models

S.no	algorithm	RMSE
1	Moving average	187.00
2	Linear regression	154.38
3	K-nearest neighbour	678.88
4	LSTM	58.09

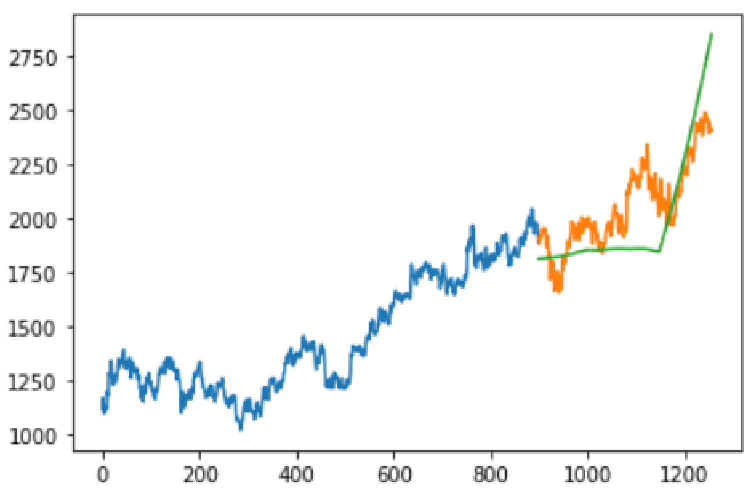

Fig 3. Graph for Moving average

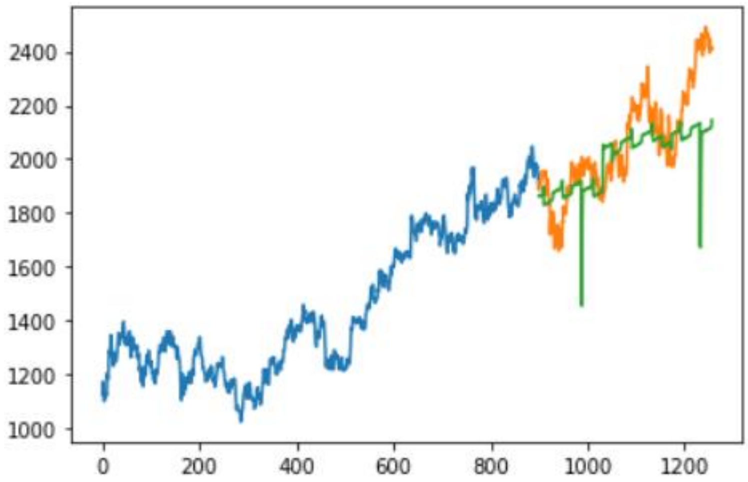

Fig 4. Graph for linear regression

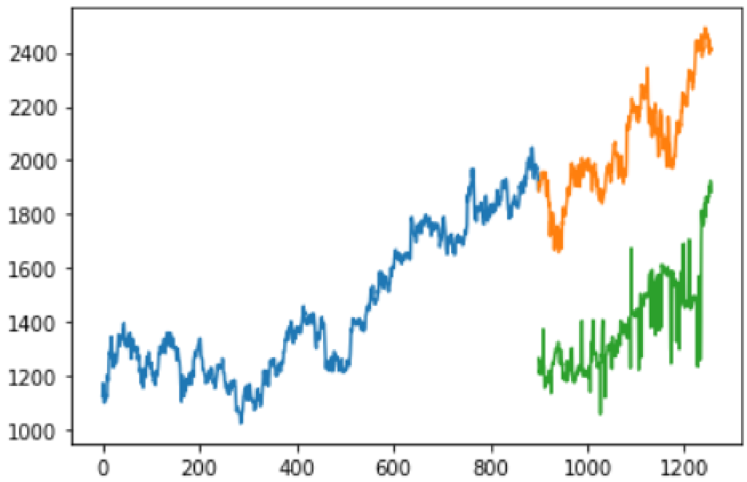

Fig 5. Graph for K- Nearest neighbour

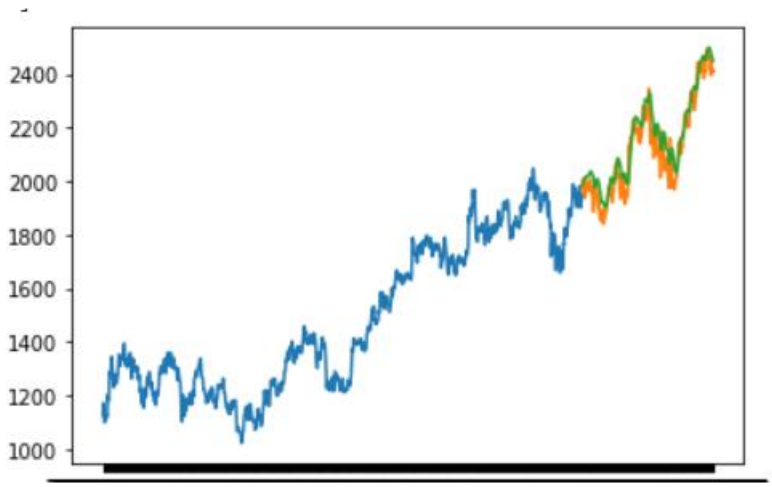

Fig 6. Graph for Long Short term memory

Hence, from the above graphs and RMSE value we can infer that moving average[fig. 3] was able to predict the movement to an extent but not accurately. In the linear regression model[fig. 4], the model was able to predict when there is a drop in the price but we can still aim for an even better model. In the KNN model[fig. 5], the model was able to predict how the market is moving but the price value hugely varied as we can see from the RMSE value. The KNN is a pretty convincing model but we can still aim for a better model. Now lets look at the LSTM model[fig. 6], the LSTM has predicted the price close to accurate ! we can see that the predictions given by the LSTM followed same movement as well as the price compared to the actual data. The RMSE value for LSTM model was the lowest compared to all the other models.

Hence, we can choose the LSTM model for future price prediction, now using the LSTM model we will predict the price for the next 30 days. We got the data from January 1st 2015 to January 31st 2020 for HDFC bank. Now we will predict the price for HDFC bank from January 1st 2020 to March 1st 2020.

Graph that we have obtained after prediction,

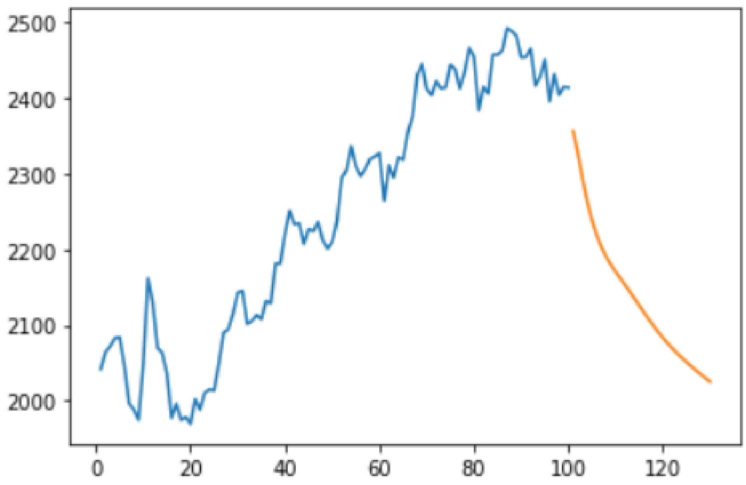

Fig 7. Prediction using the LSTM model

We can see that there is a price drop between January and march 2020 after applying our LSTM model[fig. 7]. Now let us plot the actual graph from January to march 2020 timeline to see if our model has predicted the price to an extent.

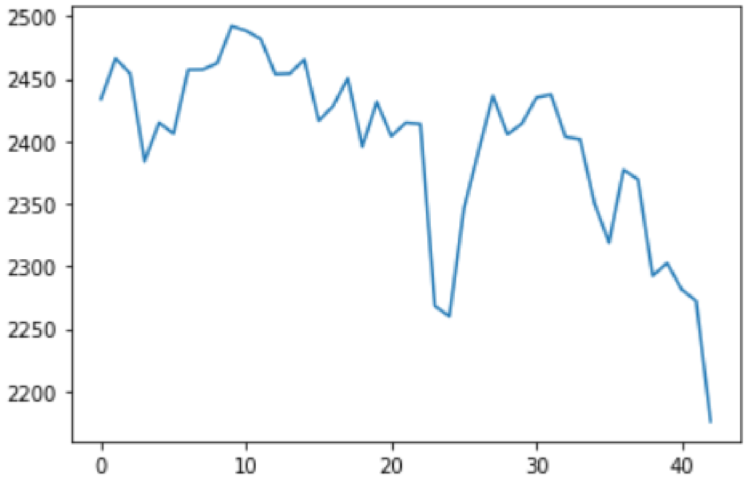

Fig 8. Actual data from January to march 2020

We can see that there was actually a price drop from January to march 2020[fig. 8]. Our model might have not predicted the price accurately but it gave a general sense on whether the price will go up or down in the next few days.

7. CONCLUSION

we have understood that stock market prediction is something that has the potential to bear great importance in the future, keeping this in mind we decided to create a model that can predict the price for the next 30 days for any stock that the user wishes to. We used a package that helped us get stock data from the NSE India website. We tested the dataset on four different models and used the best model to predict the price. The models that we used were Moving average, Linear regression, KNN and LSTM. We found that the LSTM model gave the best results with an RMSE value of 58 which was the lowest amongst all the other models used. So we then used the LSTM model to predict the price for the next 30 days, from the prediction graph that we obtained we can see that the model was able to predict the movement correctly, the model wasn't able to predict the price accurately but it for sure gave an idea to the user on whether the price will go up or down in the future. This could be very helpful for the user to as it can also be another contributing factor upon which he/she can decide if they want to invest in the stock or not. It can surely help the user to reduce the risk of losses to an extent and try to give the user profits. Of course, there is huge scope of development for the model, we can add more LSTM layers to it so that the accuracy of the model also increases. This model also comes with a website through which the user can check the future price of any stock. It can also be noted that one of the biggest advantage of this model is that it can add as an extra helping hand for novice traders as they are prone to making mistakes and face sever losses in the market compared to experienced traders. Hence, we would like to conclude by saying that we would consider this report a success if we were able to help even a single person by getting some profits by investing in stocks!

8. REFERENCES

1. Zexin Hu ,Yiqi Zhao and Matloob Khushi. A Survey of Forex and Stock Price Prediction Using Deep Learning. 2 February 2021
2. Weiwei Jiang. Applications of deep learning in stock market prediction. 15 July 2021
3. Hiransha M, Gopalakrishnan E.A., Vijay Krishna Menon and Soman K.P. NSE Stock Market Prediction Using Deep-Learning Models. 8 June 2018
4. https://www.analyticsvidhya.com/blog/2018/10/predicting-stock-price-machine-learningnd-deep-learning-techniques-python/
5. https://analyticsindiamag.com/hands-on-guide-to-lstm-recurrent-neural-network-for-stock-market-prediction/
6. https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/

9. APPENDIX

8.1 getting the data and splitting it into train and valid datasets

```
inp = input("type code: ")

data = get_history(symbol=inp, start=date(2015,1,1), end=date(2020,1,31))
data.to_csv("data.csv")
df = pd.read_csv("data.csv")
df.head()
df['Date'] = pd.to_datetime(df.Date,format='%Y-%m-%d')
df.index = df['Date']
```

```
data = df.sort_index(ascending=True, axis=0)
new_data = pd.DataFrame(index=range(0,len(df)),columns=['Date', 'Close'])

for i in range(0,len(data)):
     new_data['Date'][i] = data['Date'][i]
     new_data['Close'][i] = data['Close'][i]
train = new_data[:900]
valid = new_data[900:]
```

8.2 code for moving average

```
preds = []
for i in range(0,valid.shape[0]):
    a = train['Close'][len(train)-248+i:].sum() + sum(preds)
    b = a/248
    preds.append(b)

rms=np.sqrt(np.mean(np.power((np.array(valid['Close'])-preds),2)))
print('\n RMSE value on validation set:')
print(rms)
valid['Predictions'] = 0
valid['Predictions'] = preds
plt.plot(train['Close'])
plt.plot(valid[['Close', 'Predictions']])
```

```
 RMSE value on validation set:
187.00795055430112
```

8.3 code for linear regression

```
from fastai.tabular.core import add_datepart
add_datepart(new_data, 'Date')
new_data.drop('Elapsed', axis=1, inplace=True)
train = new_data[:900]
valid = new_data[900:]

x_train = train.drop('Close', axis=1)
y_train = train['Close']
x_valid = valid.drop('Close', axis=1)
y_valid = valid['Close']

from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(x_train,y_train)
preds = model.predict(x_valid)
rms=np.sqrt(np.mean(np.power((np.array(y_valid)-np.array(preds)),2)))
rms
dat = valid['Close']
check = pd.DataFrame(dat)
check['pred'] = preds
plt.plot(train['Close'])
```

```
plt.plot(valid['Close'])
plt.plot(check['pred'])
```

```
rms
```

```
154.3808738607781
```

8.4 code for KNN

```
x_train_scaled = scaler.fit_transform(x_train)
x_train = pd.DataFrame(x_train_scaled)
x_valid_scaled = scaler.fit_transform(x_valid)
x_valid = pd.DataFrame(x_valid_scaled)

params = {'n_neighbors':[2,3,4,5,6,7,8,9]}
knn = neighbors.KNeighborsRegressor()
model = GridSearchCV(knn, params, cv=5)

model.fit(x_train,y_train)
preds = model.predict(x_valid)
rms=np.sqrt(np.mean(np.power((np.array(y_valid)-np.array(preds)),2)))
rms
dat = valid['Close']
check = pd.DataFrame(dat)
check['pred'] = preds
plt.plot(train['Close'])
plt.plot(valid['Close'])
plt.plot(check['pred'])
```

```
rms
```

```
678.8880780635088
```

8.5 code for LSTM

```
scaler = MinMaxScaler(feature_range=(0, 1))
scaled_data = scaler.fit_transform(dataset)
x_train, y_train = [], []
for i in range(60,len(train)):
    x_train.append(scaled_data[i-60:i,0])
    y_train.append(scaled_data[i,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 = Sequential()
model.add(LSTM(units=50, return_sequences=True,
input_shape=(x_train.shape[1],1)))
model.add(LSTM(units=50))
```

```
model.add(Dense(1))

model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(x_train, y_train, epochs=1, batch_size=1, verbose=2)

inputs = new_data[len(new_data) - len(valid) - 60:].values
inputs = inputs.reshape(-1,1)
inputs  = scaler.transform(inputs)

X_test = []
for i in range(60,inputs.shape[0]):
    X_test.append(inputs[i-60:i,0])
X_test = np.array(X_test)

X_test = np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
closing_price = model.predict(X_test)
closing_price = scaler.inverse_transform(closing_price)

rms=np.sqrt(np.mean(np.power((valid-closing_price),2)))
rms
pd.options.mode.chained_assignment = None
train = new_data[:987]
valid = new_data[987:]
valid['Predictions'] = closing_price
plt.plot(train['Close'])
plt.plot(valid[['Close','Predictions']])
```

```
rms
```

```
58.095838753716265
```

10. GITHUB LINK

https://github.com/P4ITHV1/Stock-market-price-prediction