International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

Stock Market Forecasting using Natural Language Processing and Long Short Term Memory

Mohit Chandorkar
Department of Computer Science
Vidyalankar Institute of Technology
Mumbai,India

Saamiya Newrekar
Department of Information Technology
Vidyalankar Institute of Technology
Mumbai, India

***Abstract*— Stock market is the backbone of economy. It has been a misconception of people in the past that investing money in stock market is followed by greater risks. Analyzing, making predictions, and deciding which stock to invest in has never been easy. Therefore, lots of factors like supply and demand, company related factors, investor sentiment, interest rates, current events, exchange rates, declaring of bonuses etc. play an important role in the movement of stock prices and analyzing the trend is an inherent part of the market. The primary purpose of the paper is to extract information from the news and latest trends and forecast the market from point of view of the investor. There are two stock markets in India, The Bombay Stock Exchange and The National Stock Exchange, both are regulated by SEBI(Securities and Exchange Board of India). It is now possible to predict prices of stock market using various algorithms. We incorporate various text pre processing methods such as stopwords removal, normalization, lemmatization, stemming ,tokenization, bag of words ,TF/IDF. Later we use LSTM (Long Short Term Memory),LSTM are a special kind of Recurrent Neural Network, capable of learning long-term dependencies in time series. They can be widely used and also work well on huge variety of problems.**

Keywords—Stock market; LSTM; Text Preprocessing; Neural Networks

I. INTRODUCTION

Prediction of future stock prices has always been a conundrum. Investing in stock market is sometimes risky because one can make money quickly as well as lose money quickly. The stock market is unpredictable and difficult to analyze. An accurate forecast of future prices may lead to a greater yield of profit for investors through stock market investments. As per the predictions, investors will be able to pick the stocks that may give a higher return.Therefore, investing in stock market requires a proper analysis. According to Efficient Market which is also known as the efficient market theory states that share prices reflect all information and consistent alpha generation is impossible and stocks always trade at their fair value on exchanges, making it impossible for investors to purchase undervalued stocks or sell stocks for inflated prices. Therefore, it should be impossible to outperform the overall market through expert stock selection or market timing, and the only way an investor can obtain higher returns is by purchasing riskier investments**[5].**Stock market analysis is of two types: fundamental and technical. We use sentiment analysis on news headlines and analyze previous stock prices for determining the future trends of stocks. Sentiment analysis is determining whether a given text is positive or negative. We have done a detailed literature survey and after reviewing various methods, provided an optimal solution for the prediction. We use text preprocessing methods and LSTM(Long Short Term Memory) for this purpose.

II. PROPOSED METHODOLOGY

In the previous sections, we defined the goal and need for this research. Our methodology includes the following steps-

A. Data Collection
B. Data Preprocessing
C. Splitting the dataset into training and test data.
D. Building a LSTM model
E. Make predictions

A. Data Collection

We have used pandas_reader to collect the data as it extracts data from various sources across the internet into a pandas data frame. Using the documentation provided, we have used tiingo as the tracing platform to extract historical data on closing prices, equities, mutual funds and ETFs on a daily basis. For our research, we have collected five years of stock market data of the company AAPL from 02/08/2016 up to 31/07/2021 consisting of 1258 rows.

B. Data Preprocessing

The real world data collected from various sources is incomplete, noisy and inconsistent. To provide uniformity and accuracy, the dataset needs to be cleaned, parsed and standardized.**[6]**
Steps involved in data preprocessing includes-

B.1- Data Cleaning

Missing or noising data needs to be cleaned. Either the tuples with missing values are ignored or the missing values are filled using the mean, median mode of the respective column. Noisy data is smoothed out using regression techniques or clustering techniques.

B.2- Data Transformation

Data needs to be transformed in appropriate forms to make data mining effective. One such technique used for

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

transforming data is normalization is used to scale the data values in a specified range for example -1.0 to 1.0.

B.3- Data Reduction

If the volume of data is large, working with such a dataset can become difficult, hence data reduction is used to increase storage efficiency and analysis of the dataset. Dimensionality reduction used to reduce the data size by encoding mechanisms and attribution subset selection used to select on the most relevant features are some of the techniques used for data reduction.

The data collected using pandas_datareader and tiingo is of consistent nature and hence most of the data preprocessing is not required as the dataset does not have any missing values or noisy data. /the dataset which is indexed as per the date needs to be reindexed according to the closing prices. For time series data before we split the dataset into training and test data, we need to scale it. We have used MinMaxScaler in the range 0 to 1 for our research. This would scale each feature in the range of 0.0 to 1.0.

C. Splitting the dataset into training and test data

Splitting the data into training and test sets is an important step as it prevents over familiarization of the model and prevents overfitting. Training data is the data the machine learning model uses to learn, it is the data that the model sees. It is used to fit the model. Test data is used to evaluate the performance of the model, it is the data that the model makes predictions on. Test data is the new, unseen data for the machine learning model as it did on train on it. For our research, we have split the data into 65% training data X_train, y_train and 35% test data X_test, y_test . Matrix X consists of the independent features and matrix Y is the dependent feature, the one that we are predicting based on the features in X. 817 rows are in the training set and 441 are in the test set.

Before we build the lstm model, we need to process the training datasets using a time step.

```
import numpy
def creatingDataset(dataset,
timeStep=1):
dataset_X, dataset_Y = [], []
for i in range(len(dataset)-timeStep-
1):
a = dataset[i:(i+timeStep), 0
dataset_X.append(a)
dataset_Y.append(dataset[i + timeStep,
0])
return numpy.array(dataset_X),
numpy.array(dataset_Y)

timeStep = 100
X_train, y_train =
creatingDataset(train_data, timeStep)
X_test, y_test =
creatingDataset(test_data, timeStep)
```

The resulting `X_train, y_train` are two dimensional and `X_test, y_test` one dimensional datasets respectively. Before we feed the `X_train` and `X_test` into the LSTM model we need to make them into three-dimensional datasets. This is done using the `.reshape()` function.

D. Building the stacked LSTM model

Recurrent Neural Networks, commonly known as RNN's suffer from short term memory problem. This can be treated with LSTM's. LSTM is a special version of RNN which solves the short term memory problem. In short, Long term dependency problem can be solved with LSTM.

Long Short-Term Memory or LSTM is a complex deep learning model which is a specialized recurrent neural network capable of learning long-term dependencies in a sequence prediction scenario.**[4]**

As our dataset is of time series nature and we are essentially researching time series forecasting, we have used the LSTM model.

We would need the Sequential model from keras and the Dense and LSTM layers from keras as well.

```
from tensorflow.keras.models import
Sequential
from tensorflow.keras.layers import
Dense
from tensorflow.keras.layers import
LSTM
```

The sequential model will have three LSTM layers stacked one after the other followed by a dense layer used for the output. The first LSTM layer would be the input layer and the output of which would act as the input to the next layer, doing so in a sequential manner. Finally, we have compiled the model using the `mean_squared_error` loss function and `adam` optimizer.**[7]**

The model is then trained using `X_train, y_train` validated against `X_test, y_test` using `epochs=100` and `batch_size=64.`

E. Make predictions

Our LSTM model is trained and now we will use the `model.predict()` function to predict the model and discuss the results in the next section of this research paper.

Now we'll make the predictions and check the performance of our model

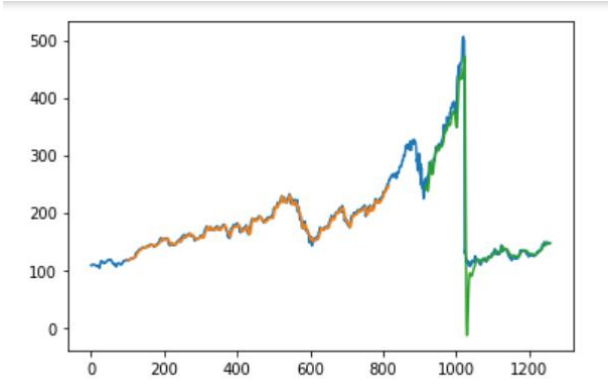
```
train_predict=model.predict(X_train)
test_predict=model.predict(X_test)
```

III. RESULTS

The graph (Fig[1]) and code represent the predictions of training and test datasets made by our LSTM model. The green line represents predictions made on the test data, the orange line represents predictions made on the training data, the blue line is the complete datasets.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

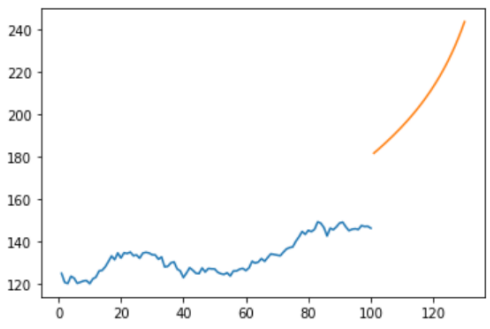
```
### Plotting
lookBack=100

train_predict_plot =
numpy.empty_like(df1)
train_predict_plot [:, :] = np.nan
train_predict_plot
[look_back:len(train_predict)+look_back
, :] = train_predict
train_predict_plot =
numpy.empty_like(df1)
train_predict_plot [:, :] = numpy.nan
train_predict_plot
[len(train_predict)+(lookBack*2)+1:len(
df1)-1, :] = test_predict
# plot baseline and predictions
plt.plot(scaler.inverse_transform(df1))
plt.plot(train_predict_plot)
plt.plot(train_predict_plot)
plt.show()
```



Fig[1]

The given graph (Fig[2]) represents the forecasting of next 30 days based on past 100 days.

```
from numpy import array

output1=[]
n_steps=100
i=0
while(i<30):

    if(len(temp_input)>100):
        #print(temp_input)

x_input=np.array(temp_input[1:])
        print("{} day input
{}".format(i,x_input))
        x_input=x_input.reshape(1,-1)
        x_input = x_input.reshape((1,
n_steps, 1))
        #print(x_input)
        yhat = model.predict(x_input,
verbose=0)
        print("{} day output
{}".format(i,yhat))

temp_input.extend(yhat[0].tolist())
        temp_input=temp_input[1:]

        output1.extend(yhat.tolist())
        i=i+1
    else:
        x_input = x_input.reshape((1,
n_steps,1))
        yhat = model.predict(x_input,
verbose=0)
        print(yhat[0])

temp_input.extend(yhat[0].tolist())
        print(len(temp_input))
        output1.extend(yhat.tolist())
        i=i+1

print(output1)
```



Fig[2]

The root mean squared error for training dataset is 181.670
The root mean squared error for test dataset is 241.4237

IV. CONCLUSION

In this paper presented above, we have successfully managed to predict the stock price movement based on previous close for 100-120 days. We have built our machine learning model using LSTM's which stand for Long Short-Term Memory. First, we process the large chunks of data taken from various sources. The model learns from previous stock price close and improves in terms of accuracy. We used API to get data of AAPL stocks which can be used well with Bombay Stock Exchange, National Stock Exchange, or NASDAQ. We have performed with a overall good accuracy and low error rate. There are many more approaches to this research, but this one was found to be more useful and easier to understand. People can now invest in stocks without fear of losing money with a proper prior knowledge. The code snippets as well as graphs have been provided above in result section.

V. REFERENCES

[1] Sidra Mehtab, Jaydip Sen," A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing"

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 10 Issue 08, August-2021

[2] S. S. Abdullah, M. S. Rahaman and M. S. Rahman, "Analysis of stock market using text mining and natural language processing," 2013 International Conference on Informatics, Electronics and Vision (ICIEV), 2013, pp. 1-6, doi: 10.1109/ICIEV.2013.6572673.

[3] Nishant Verma, S G David Raj, Ackley J Lyimo, Kakelli Anil Kumar," Stock Market Prediction and Risk Analysis using NLP and Machine Learning", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-9 Issue-5, June 2020.

[4] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

[5] https://www.investopedia.com/terms/e/efficientmarkethypothesis.asp

[6] https://www.geeksforgeeks.org/data-preprocessing-in-data-mining/

[7] https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

[8] Ghosh, Pushpendu & Neufeld, Ariel & Sahoo, Jajati. (2020). Forecasting directional movements of stock prices for intraday trading using LSTM and random forests.