DELHI TECHNOLOGICAL UNIVERSITY

IT323
Machine Learning
Project Report

Project Title:
Stock Market Prediction

Submitted To;
Dr. Dinesh Kumar Vishwakarma
Associate Professor
Department of Information Technology

Submitted By:
Manish Devkota
2K18/EC/095

CONTENTS

Introduction

Stock market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analyzing the trend over the last few years, could prove to be highly useful for making stock market movements . Traditionally, two main approaches have been proposed for predicting the stock price of an organization. Technical analysis method uses historical price of stocks like closing and opening price, volume traded, adjacent close values etc. of the stock for predicting the future price of the stock. The second type of analysis is qualitative, which is performed on the basis of external factors like company profile, market situation, political and economic factors, textual information in the form of financial new articles, social media and even blogs by

economic analyst . Nowadays, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. Particularly, for stock market analysis, the data size is huge and also non-linear. To deal with this variety of data efficient model is needed that can identify the hidden patterns and complex relations in this large data set. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods.

OBJECTIVE

The Objective of the project is to predict the future price of a given stock using different models (arima model, lstm model) and comparing them on the basis of their result.

Dataset

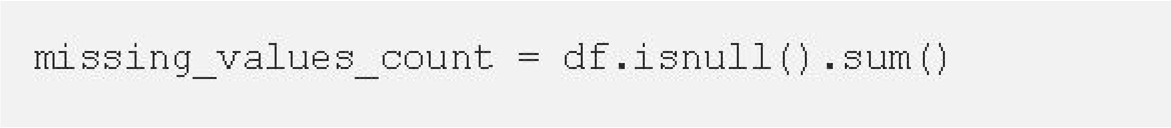
The dataset Used in training the model is from kaggle website stocks in NIFTY-50 index from NSE India. Dataset contain The data is the price history and trading volumes of the fifty stocks in the index NIFTY 50 from NSE (National Stock Exchange) India.

Data Preprocessing

Data Cleaning

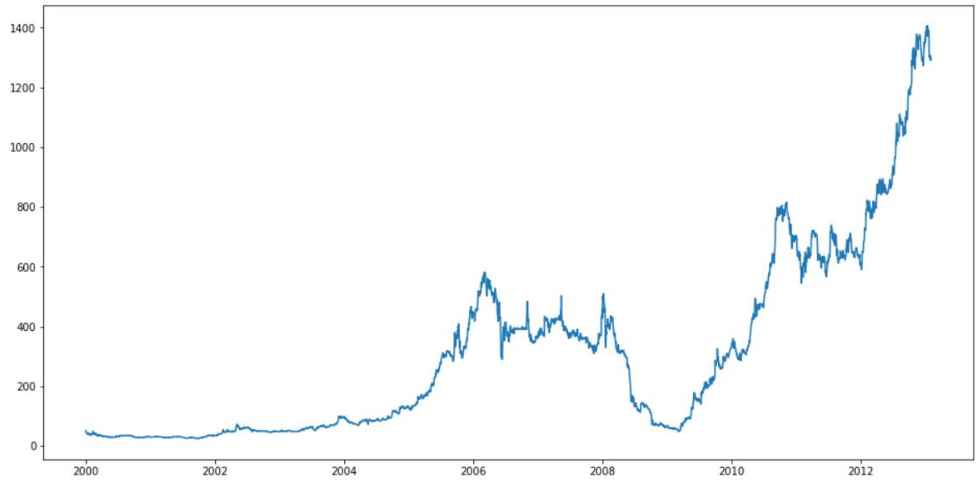
After some simple manipulations and loading of the csv data into pandas DataFrame, we have the following dataset where open, high, low and close represent prices on each date and volume the total number of shares traded. As only needed closing price hence we will take only closing price and drops others.

After that will we check for missing data using this code:

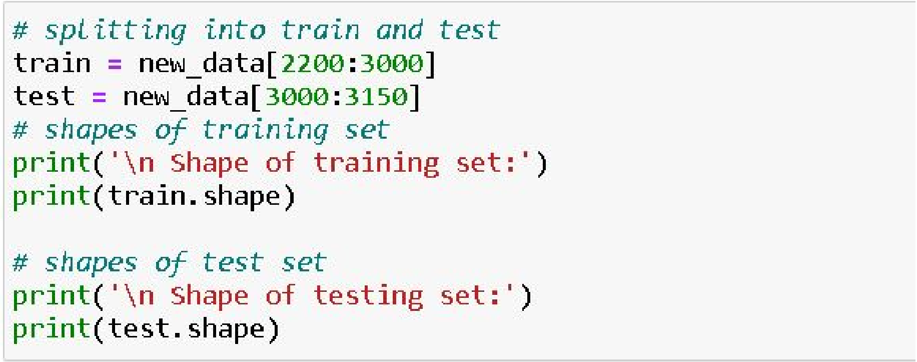


```
missing_values_count = df.isnull().sum()
```

Visualising the Time Series data:



Then splitting the data into training set and testing set



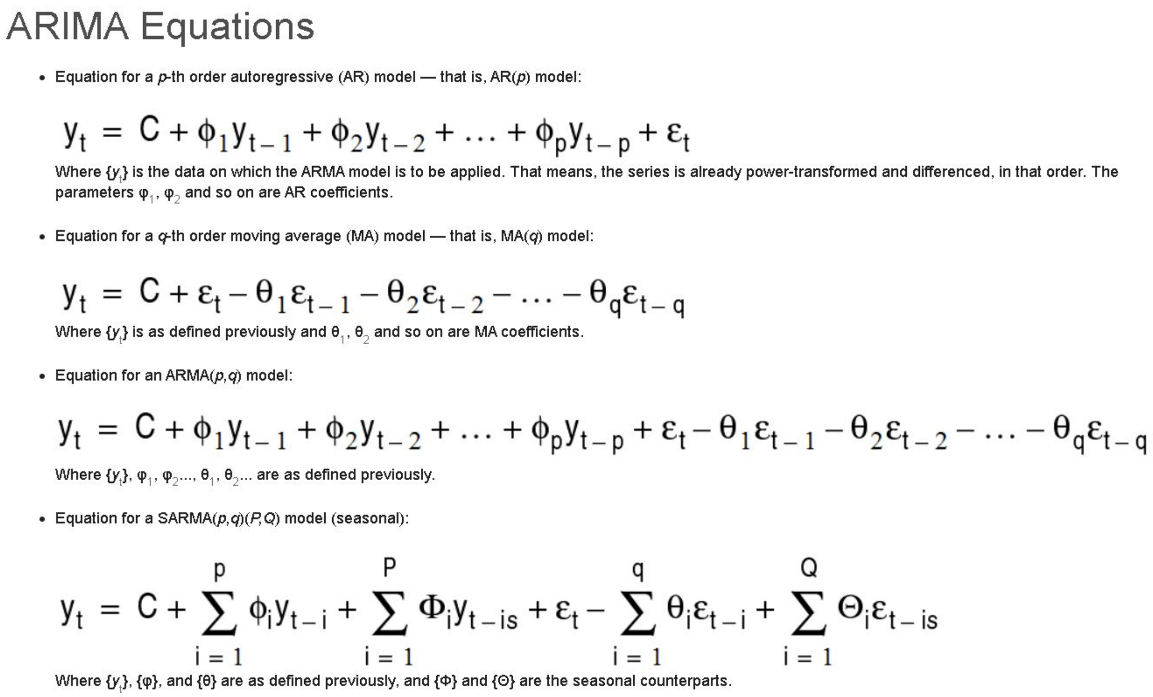
```
# splitting into train and test
train = new_data[2200:3000]
test = new_data[3000:3150]
# shapes of training set
print('\n Shape of training set:')
print(train.shape)

# shapes of test set
print('\n Shape of testing set:')
print(test.shape)
```

Model Equation and Architecture

!. Arima Model

an **autoregressive integrated moving average (ARIMA)**[1][1]model is a generalization of an autoregressive moving average (ARMA) model. Both of these models are fitted to time series data either to better understand the data or to predict future points in the series (forecasting). ARIMA models are applied in some cases where data show evidence of non-stationarity, where an initial differencing step (corresponding to the "integrated" part of the model) can be applied one or more times to eliminate the non-stationarity.

Non-seasonal ARIMA models are generally denoted ARIMA(p,d,q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p,d,q)(P,D,Q)$_m$, where m refers to the number of periods in each season, and the uppercase P,D,Q refer to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.



ARIMA Equations

- Equation for a *p*-th order autoregressive (AR) model — that is, AR(*p*) model:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t$$

Where $\{y_t\}$ is the data on which the ARMA model is to be applied. That means, the series is already power-transformed and differenced, in that order. The parameters φ_1, φ_2 and so on are AR coefficients.

- Equation for a *q*-th order moving average (MA) model — that is, MA(*q*) model:

$$y_t = C + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q}$$

Where $\{y_t\}$ is as defined previously and θ_1, θ_2 and so on are MA coefficients.

- Equation for an ARMA(*p*,*q*) model:

$$y_t = C + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q}$$

Where $\{y_t\}$, φ_1, φ_2..., θ_1, θ_2... are as defined previously.

- Equation for a SARMA(*p*,*q*)(*P*,*Q*) model (seasonal):

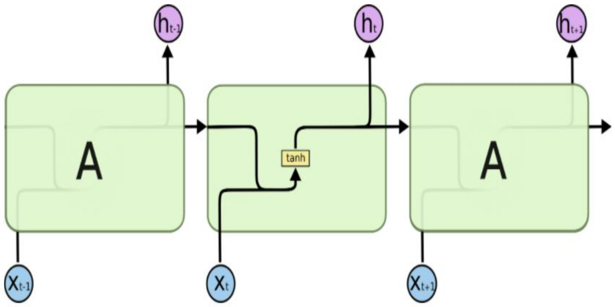
$$y_t = C + \sum_{i=1}^{p} \phi_i y_{t-i} + \sum_{i=1}^{P} \Phi_i y_{t-is} + \varepsilon_t - \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \sum_{i=1}^{Q} \Theta_i \varepsilon_{t-is}$$

Where $\{y_t\}$, $\{\varphi\}$, and $\{\theta\}$ are as defined previously, and $\{\Phi\}$ and $\{\Theta\}$ are the seasonal counterparts.

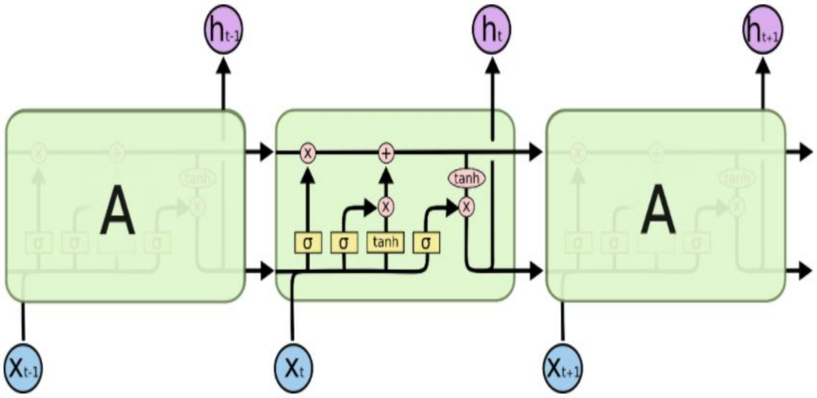
LSTM Model:

Long short-term memory (**LSTM**)[2][2] is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM has feedback connections. It can not only process single data points (such as images), but also entire sequences of data (such as speech or video). For example, LSTM is applicable to tasks such as unsegmented, connected handwriting recognition, speech recognition and anomaly detection in network traffic or IDSs (intrusion detection .

A problem with normal rnn network is vanishing gradient problem In such cases, where the gap between the relevant information and the place that it’s needed is small, RNNs can learn to use the past information. But as that gap grows, RNNs become unable to learn to connect the information. To over come this problem we use Lstm.



The repeating module in a standard RNN contains a single layer.



The repeating module in an LSTM contains four interacting layers.

The **cell state** is kind of like a conveyor belt. It runs straight down the entire chain, with only some minor linear interactions. It's very easy for information to just flow along it unchanged.

Gates are a way to optionally let information through. They are composed out of a sigmoid neural net layer and a pointwise multiplication operation.

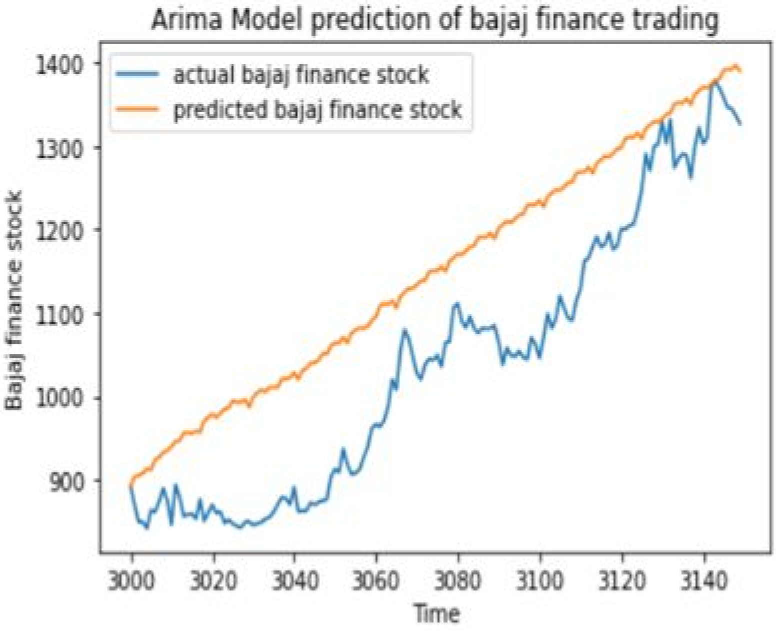
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

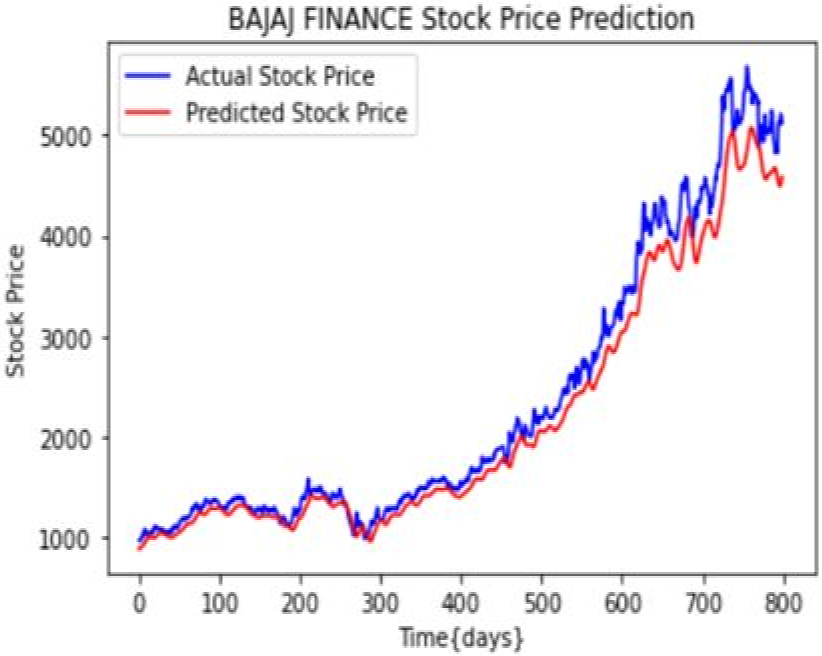
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma\left(W_o \ [\, h_{t-1}, x_t] \ + \ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Result:



Arima Model Output



LSTM model Output

Conclusion

By comparing the two forecasting plots, we can see that the ARIMA model has predicted the closing prices very lower to the actual prices. This large variation in prediction can be seen at the majority of the places across the plot. But in the case of the LSTM model, the same prediction of closing prices can be seen higher than the actual value. But this variation can be observed at few places in the plot and majority of the time, the predicted value seems to be nearby the actual value. So we can conclude that, in the task of stock prediction, the LSTM model has outperformed the ARIMA model.

We can also conclude this by comparing Root Mean Squared Error.

RMSE with Arima 127.640563493
RMSE with LSTM 26.923716650748005

Dataset link : https://www.kaggle.com/rohanrao/nifty50-stock-market-data
Libraries Used: keras tensorflow Lstm pmdarima pandas numpy matplotlib.

References

[1] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock Price Prediction Using the ARIMA Model,” in *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, Cambridge, United Kingdom, Mar. 2014, pp. 106–112.

[2] D. M. Q. Nelson, A. C. M. Pereira, and R. A. de Oliveira, “Stock market’s price movement prediction with LSTM neural networks,” in *2017 International Joint Conference on Neural Networks (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 1419–1426.