Stocks Price Prediction

Guide : Prof. Prithwijit Guha

Deepak Kumar

Department of Physics
Indian Institute of Technology Guwahati

30th April, 2022

Introduction

- Stock Markets are complex & unpredictable in nature due to various social, macroeconomic & political factors, which makes it difficult to predict the future returns, the recent advancements in the field of machine learning & deep neural networks have made it possible to understand & analyze the movements of stocks.
- In this project we worked on developing two regression methods using Random forests & recurrent neural networks (RNNs) based long short term memory (LSTM) model to forecast future stock returns, later we've implemented construction and optimization of portfolios using Markowitz's Modern Portfolio Theory & Particle Swarm Optimization for National Stock Exchange(NSE) listed companies.

Path of Discussion

Historical Stock Datasets
- ✓ From yfinance
- ✓ **OHLCV**
- ✓ Top **350** companies of NSE a/c market capitalization

Data Cleaning
- ✓ Detecting & Removing Outliers

Feature Engineering
- ✓ Drop irrelevant columns
- ✓ Adding new columns
- ✓ **Normalization**

Training Data

Testing Data

Learning Algorithm
- ✓ LSTM
- ✓ Random Forest

Trained Model

Predictions

Evaluation Parameters
- ✓ RMSE
- ✓ MAE
- ✓ MAPE

Portfolio Construction
- ✓ Modern Portfolio Theory

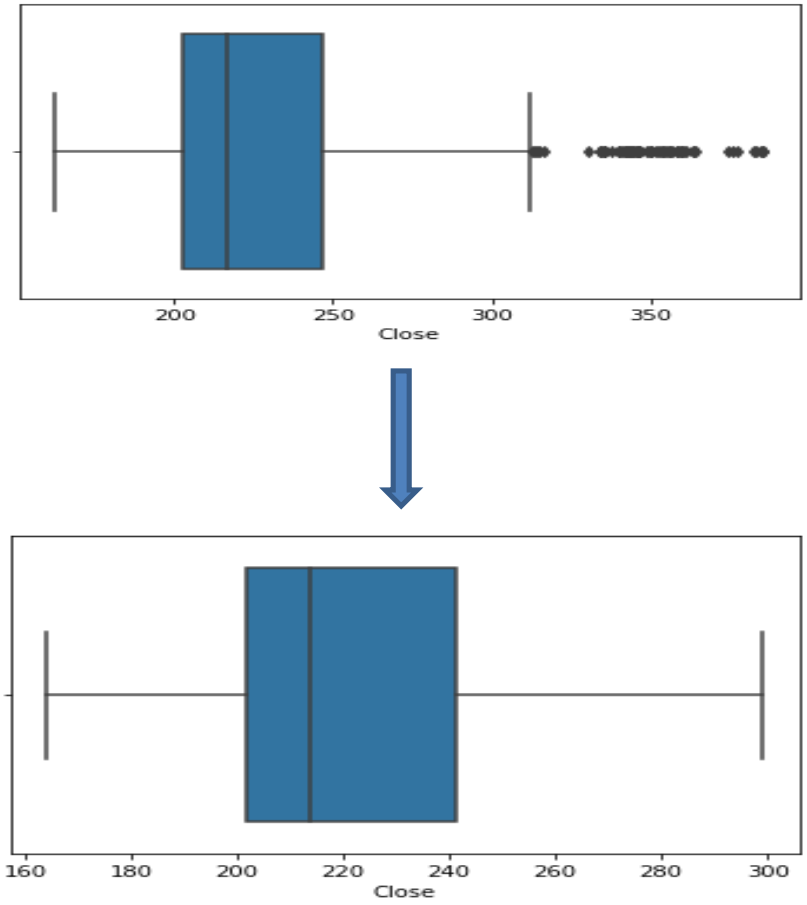
Portfolio Optimization
- ✓ Particle Swarm Optimization
- ✓ Fitness function

Results
- ✓ 7 day trading results for a portfolio of 150 companies

The Data

Data cleaning

The raw dataset's features may have values which in general differ in greater magnitude from the rest of the data points referred to as outlier data points, presence of outliers in the dataset can lead to increased error variance, so most machine learning or deep learning algorithms doesn't works well in the presence of outliers, therefore it's necessary to remove them .



For cleaning the data we used **Inter Quartile Range(IQR)** defined as:

IQR = $Q_3 - Q_1$

Q_1 denotes that about 25% of the values lies below 75% of values lies above this number while Q_3 denotes that about 25% of the values lies above this number 75% lies below this number.

A value in the data is termed as an outlier if it falls in the region given as:

$$X_i < (Q_1 - 1.5 * IQR) \quad or \quad X_i > (Q_3 + 1.5 * IQR)$$

The Data

Feature Engineering

Derived Features :

$$F_i = \frac{X_i - X_{i-t}}{t}$$

$$S_i = \frac{X_i - 2X_{i-t} + X_{i-2t}}{2t}$$

$$SMA = \frac{X_i + X_{i-t} + \cdots + X_{i-t+1} + X_{i-t}}{t}$$

$$WMA = \frac{(X_i * t) + (X_{i-t} * (t-1)) + \cdots + X_{i-t}}{t}$$

$$A/D = \frac{(C_i - L - i) - (H_i - C_i)}{H_i - L_i}$$

Data Normalization :
we scale our data which is beneficial before training the model for keeping all the variables on same scale and also it speeds up the learning which leads to faster convergence.

Min-Max Normalization :

$$X_{scaled,i} = \frac{X_i - X_{Min}}{X_{Max} - X_{Min}}$$

Removing Correlated Features :
The features present in a dataset can be similar or related, so in order to analyze or observe such relationships among the features, we use Pearson's correlation coefficient .

$$\rho_{x,y} = \frac{cov(x,y)}{\sigma_x \sigma_y}$$

Pearson's correlation coefficient ranges from [-1, 1] .

The Data

Preparing training & testing data

Random Forest

Supervised Regression problem:	Variables	Training and Testing data
Final pre-processed data  Closing & Opening Prices Predict	 ✓ Open,Close,High,Volume Traded ✓ Weighted Moving Average, 2nd Order Change	• Training (**65%**) , Testing (**35%**)

LSTM

Supervised Regression problem:	Variables	Training and Testing data
Sequential data of the final pre-processed data  Closing & Opening Prices Predict	 ✓ Open,Close,High,Volume Traded ✓ Weighted Moving Average, 2nd Order Change	• Training (**65%**) , Testing (**35%**)

Stocks Price Forecasting

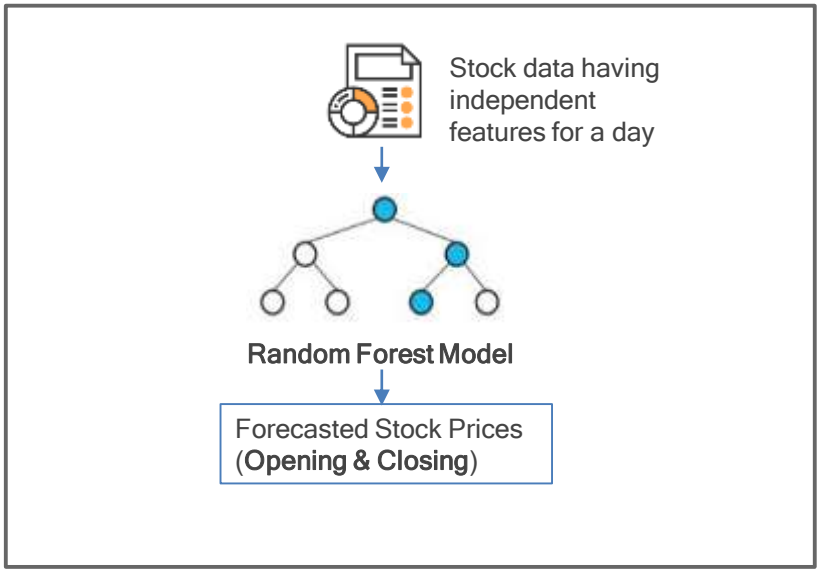
Model Training

Random Forest is an ensemble based model that uses multiple decision trees, for improving predictive accuracy using averaging & to avoid overfitting.

The core of the ensemble is decision trees which uses hierarchical mode for learning, consisting of decision nodes and leaf nodes.

Decision nodes consists of the independent features used for training the model with certain conditions on them through which the tree is further split through two branches, the process continues in a similar way, the leaf nodes consists of numerical values or categorial output.

For a regressive decision tree the criteria used for splitting the tree is Mean Squared Error.



Hyper parameter Tuning

- ✓ No of estimators or decision trees
- ✓ Maximum depth of tree
- ✓ Maximum Features

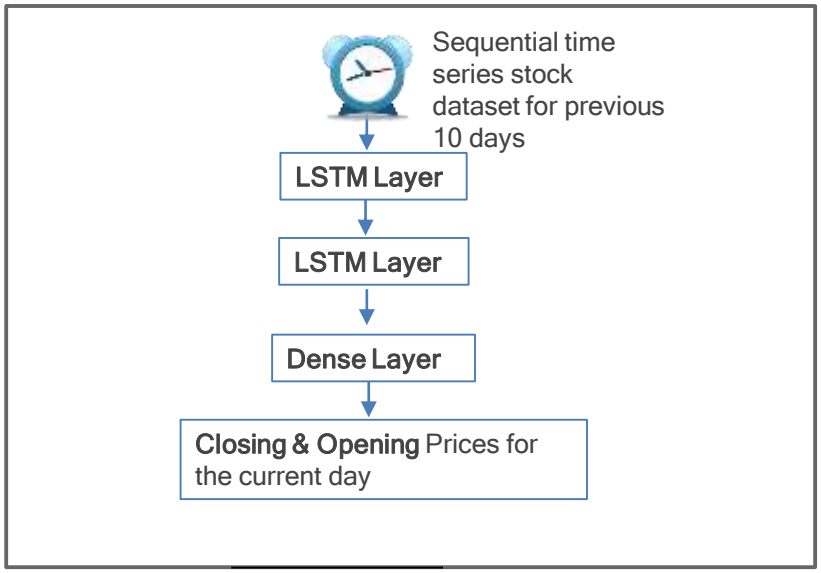
Stock Price Forecasting

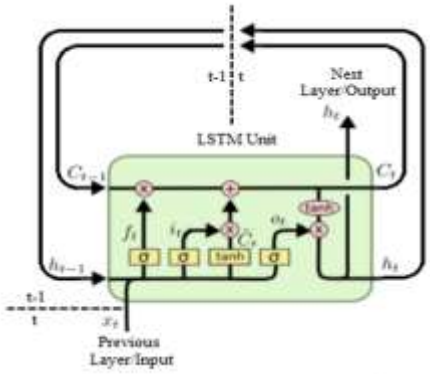
Model Training

Stock market data is essentially a time series data which are sequential in nature, therefore we used **LSTM** model .

Long **S**hort **T**erm **M**emory is an sequential deep learning model architecture considered to be highly successful for dealing with sequential data due to it's advantages over other sequential models.

Inside the **LSTM** cell, two inputs are provided one from adjacent cell while other from previous layer , for first layer they are input data used for training, from which the cell's parameters are trained. This operation continues through all the layers .





$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t] + b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Stock Price Forecasting

Model Training

Sequential data for training the model is passed as input to the first layer of LSTM cells, through which it learns & determines the parameters of the cells, then output passes on to the next layers & to the next adjacent cells , finally the last cell's output then passes through a dense layer having activation function(**ReLU**) for forecasting the output .

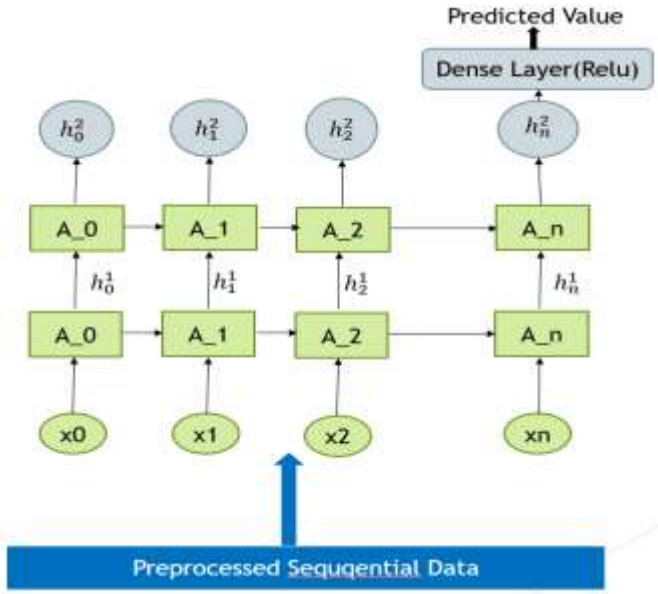
Through the dense layer the stock forecast is calculated as :
where $g(\theta)$ is the activation function for dense layer, defined as

$$g(\theta) = max\ (0, \theta)$$

where, θ is calculated as :

$$\theta = wx + b$$

where , w & b represents the hidden parameters of the cell in vector of h dimensional space and x represents the input for the dense layer.



Hyper parameter Tuning

- ✓ No of LSTM cells & layers
- ✓ Dropout ratio
- ✓ Activation Function
- ✓ Number of Epochs

Stock Price Forecasting

Predictions

Results

The below table contains the average of errors for the testing dataset .

Model	RMSE	MAE	MAPE
Random Forest	1.870	1.627	0.075
LSTM Model	2.233	1.919	0.088

Portfolio Construction & Optimization

Portfolio Construction

American Economist Harry Markowitz introduced the modern portfolio theory to optimize the risks & returns for a risk averse investor aiming to maximize the returns for a given level of risks, using variance of asset prices as a proxy for risk, stating the importance of diversification of portfolio which eventually minimizes the risks for an investor.

$$F_0 = \sum_{i=1}^{N} w_i s_i^o$$

$$r_i = s_i^c - s_i^o$$

An intra day investor having a portfolio of N stocks, then for a trading day, let F_0 be the initial value of portfolio, w_i are the weights for distribution of capital into the portfolio & s_i^o is the value of r_i stock at the time of buying. Let s_i^c be the stock price at the time of selling then unit returns per stock then can be written by , then overall return can be written as r_i .

$$P = \sum_{i=1}^{N} w_i r_i$$

$$\sum_{i=1}^{N} w_i = 1 \qquad 0 \leq w_i \leq 1$$

Portfolio Construction & Optimization

For maximizing the profit while the risk being minimum, we will have to find the optimal weights of the portfolio , for this we use the covariance of stocks in the portfolio, by assuring minimum covariance will lead to building a diverse portfolio eventually resulting in optimizing the portfolio to lesser risk. The covariance for N stocks can be written as σ^2 & and standard deviation as σ.

$$\sigma^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij}$$

$$\sigma^2 = W^T S(R) W$$

$$\sigma = \sqrt{(W^T S(R) W)}$$

Portfolio Optimization

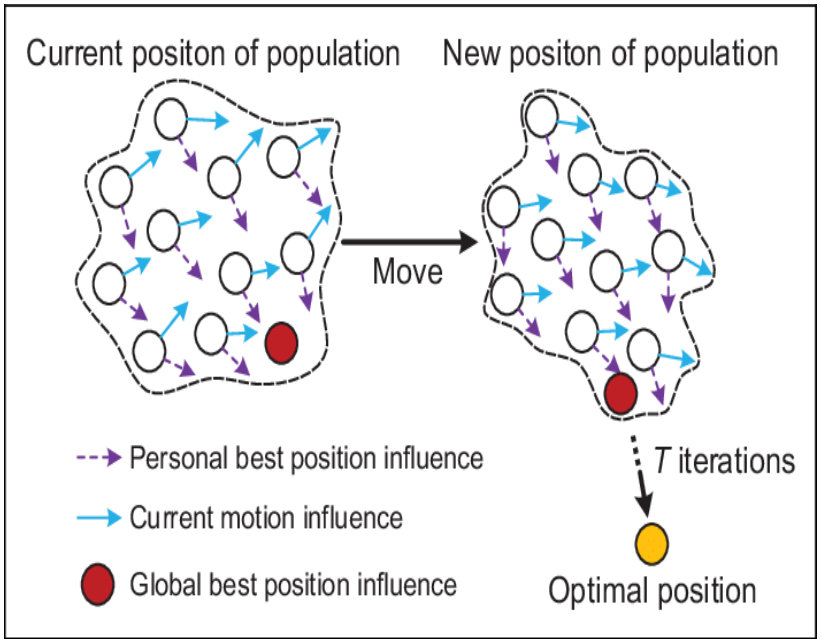
This is a case of quadratic programming which can be done using few optimization techniques like Particle Swarm Optimization, etc.

Portfolio Construction & Optimization

Portfolio Optimization

Particle Swarm Optimization(PSO) is a computational method used in optimizing a problem by iteratively aiming to improve a candidate solution with respect to a given measure of quality, by defining a fitness function. A particle swarm can be referred to a population of particles where each particle can be considered to be a moving object that is through the search space and is attracted to its previously visited locations & these particles are assumed to neither gets replaced by other particles nor reproduce.

Now, at each iteration the position of the particles changes along with the velocity to find their optimal positions & globally optimal position inside the search space.



After, the end of the iterations, we get the global optimal location inside the search space. So, for optimizing our portfolio having N companies, we use N dimensional search space to optimize the weights of the portfolio by using the fitness functions to maximize the returns while minimizing the risks .

Portfolio Construction & Optimization

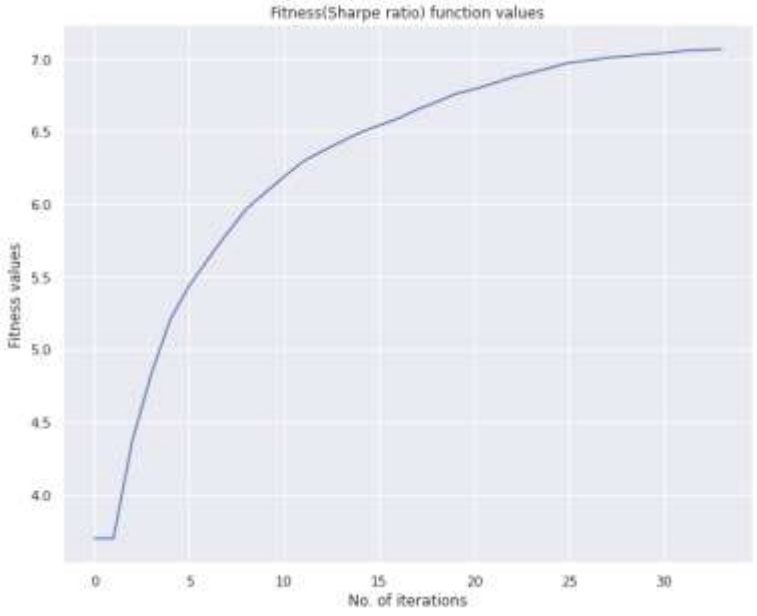
Portfolio Optimization

After predicting the forecasts prices using the models, we decided upon the companies to invest in and then constructed a portfolio using those companies & actual returns for any trading day and then optimized the portfolio weights, while keeping the fitness factor maximum using particle swarm optimization in a N dimensional search space where N being number of companies in portfolio, where the position of particles in the search space signifies the weights for N stocks of the portfolio.

Our portfolio was constructed using the historical stock dataset of top150 companies registered in NSE from various sectors, based upon the price forecasts through the forecasting model, companies were selected for the final portfolio, then optimal weights for capital distribution was calculated using particle swarm optimization using both of the fitness functions. The plots shows the various fitness function values for global optimal locations at end of each iterations.

$$sharpe\ ratio = \frac{\sum_{i=1}^{N} w_i x_i}{\sigma}$$

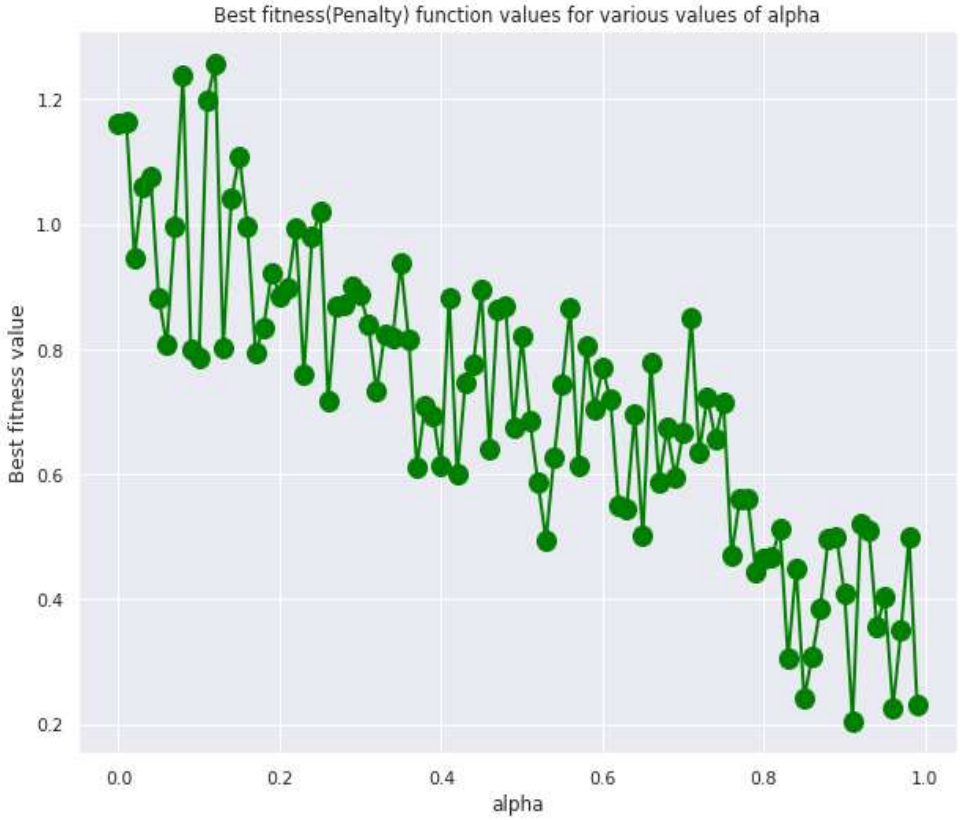
$$f_n = \sum_{i=1}^{N} w_i x_i - (\alpha * covariance)$$



Portfolio Construction & Optimization

Results

The returns of the portfolio were calculated for 7 trading days from 16th September to 24th September by finding the optimal weights of the portfolio for each day & then using individual returns per stock by assuming the buying & selling of stocks at open & close prices respectively .



Conclusion

The average performance of portfolio can be attributed to wrong predicted values by the models which ultimately can be improved by using better architecture models like Generative Adversarial Networks(GANs) which we were not able to implement due to insufficient data related to financial sentiment analysis for such large number of stocks .

The assumption of selling the stocks at the closing prices do attributes to lower returns in contrast to actual intraday traders where selling happens at somewhat near the high price.

Immediate future work in this regard can be done to improve the returns by adapting better fitness functions or improving the buying & selling decisions during trading .

-

Thank You !