Stock Price Prediction

Submitted By
by

Anshul Sanghi - 22M0758
Swamil Jain - 22M0769
Shantanu Gajanan Mapari - 22M0796
Shubham Kumar Sonkar - 22M0799

under the guidance of

Prof. Preethi Jyothi

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay
Powai 400076

1 Introduction

As we know Stock Market is of turbulent nature it's price prediction becomes extremely challenging.The turbulent nature of Stocks is a result of factors involving financial , economic , political conditions , some unpredictable events. With the help of machine learning we can have sophisticated methods for predicting stocks price.A proper model developed with an optimal set of attributes can predict stock price reasonably well and better inform the market situation. Stock price prediction is complex and challenging task but we can observe some behavioural patterns in the Time-Series Data. We can observe that the stock price of the particular stock depends on various factors and some change in one stock may affect other stocks.

2 Problem Statement

Study the impact of adding generalization to a model trained to predict price of a specific stock by using extended dataset containing multiple stocks during training, with the effects of individual stock weighted by factor of its correlation coefficients with the target stock using Time Series Data of other stocks as well.

3 Dataset

We have taken Time Series Data of NiftyFifty stocks from kaggle. We are using OHLC (Open High Low Close) data of the following stocks from four major sectors namely Banking,Pharmaceuticals,Information Technology,Energy.

Stocks from respective sectors are mentioned below :

- SBIN, HDFCBANK, AXISBANK : Banking sector
- SUNPHARMA, DRREDDY, CIPLA : Pharmaceutical sector
- INFY, WIPRO, TCS : Information technology sector
- BPCL, COALINDIA, POWERGRID : Energy sector

We are predicting SBIN using correlation.
Kaggle link of the dataset we have used :

`https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data`

4 Methodology

4.1 Preprocessing of Data

We are splitting the standard OHLC data to convert it into time series data of varying batch lengths (i.e. 30,60,90. . . . days).

4.2 Feature Generation

We are creating custom features from the preprocessed data.Using the Previous Close value to predict the Close value.Taking varying degrees of Previous Close values on time series data we are applying Linear Regression model to predict the next day Close value of the batch provided for feature generation task and using these predicted values to feed it to the next layer of our model. This is to achieve a balance between the effect of overfitting and underfitting on standard price movement which results in smoothening of the graph.

We have trained a Linear regression model for Previous Close values of degrees 2,3 and 4.

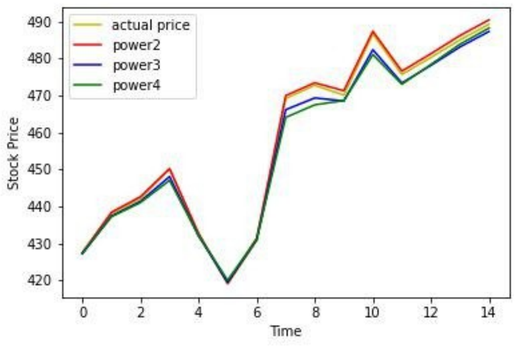


Figure 1: Feature Generation using linear regression

4.3 Finding correlation between various stocks

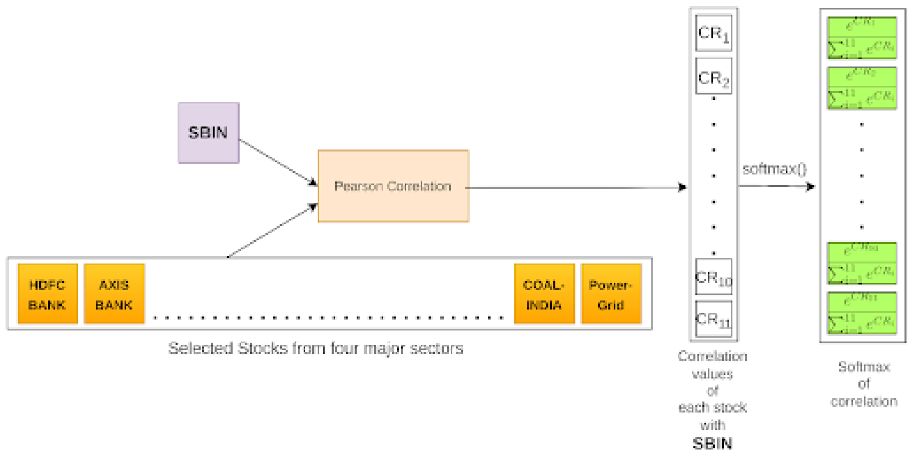


Figure 2: Softmax of correlation

To generalize our model we are using the concept of correlation so that we can better understand the relationship of one stock to another.We have calculated the Pearson Correlation between all stocks with our target stock.We want to use this correlation while training the model for generalized results.

Plotted a graph to show the relation between different stocks using correlation between Axisbank and SBI:

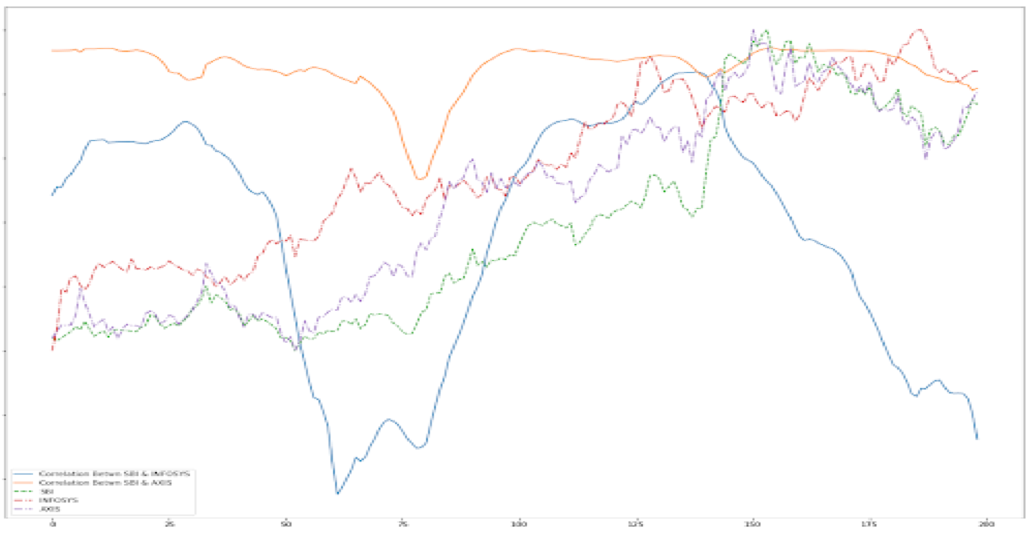


Figure 3: Correlation

Stocks selected : SBIN,INFOSYS,AXISBANK

Observations:

- Correlation between Axis bank and SBI can be seen to be positive overall
- Correlation between infosys and SBI seems to be a bit fluctuating as expected

Final correlation values: For every time series interval we used softmax to convert the correlation of each stock in relative ratios.

4.4 Custom Optimizer

We have created a custom optimizer using APIs provided by tensorflow that takes into account the correlation values while updating the weights of the layers.Used the correlation values to get weighted average of the weight gradients in each batch during training. This eventually acts as the weight or importance that we are giving to certain parts of our training data (certain stocks to be precise).As we have used softmax to convert the correlation of each stock in relative ratios we are now left with 50 % generalization(for every interval data of our target stock with correlation weight 1 and data of all other stocks with net correlation weight as 1).By multiplying softmax correlations with some factor k,we can change the ratio of generalized-specialized from 0-1.

4.5 Designing structure of RNN

Recurrent Neural Networks are best suited for sequential data. RNN is a type of neural network where output from the previous state is given as input to the current state, but if we see carefully it has the problem of Vanishing gradient when the derivative of error is less than one. RNN also suffers from the Exploding gradient problem when the derivative of error is greater than one. We are using LSTM (long short term memory) for training our model. LSTM are special type of RNN which are capable of learning long term dependencies.

LSTM are best suited for time series data. We will read the data for first 60 days and predict the value for 61 day then we will hop ahead by one day and read read the next chunk of data for next 60 days. We trained a LSTM model using the custom optimizer that we created, along with the features that created a custom optimizer that takes into account the correlation values while updating

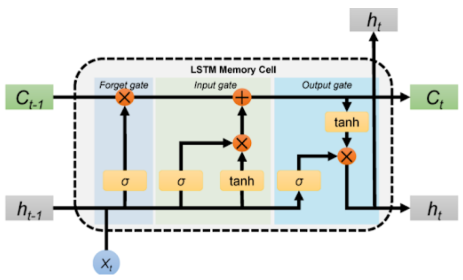
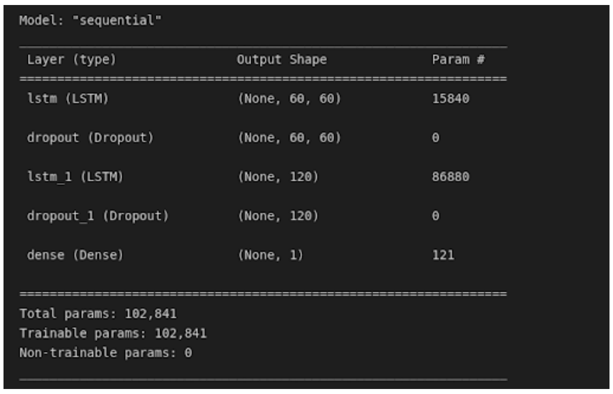


Figure 4: LSTM Structure

the weights of the layers. We mapped the correlation values that we found to the learning rate of the specific batch of data. We are using hidden layers first with 60 units and second with 120 units. Activation function is RELU Loss function is ‘mean squared loss’ We are using dropout for overfitting with value 0.2


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 60, 60)            15840

 dropout (Dropout)           (None, 60, 60)            0

 lstm_1 (LSTM)               (None, 120)               86880

 dropout_1 (Dropout)         (None, 120)               0

 dense (Dense)               (None, 1)                 121

=================================================================
Total params: 102,841
Trainable params: 102,841
Non-trainable params: 0
_________________________________________________________________
```

Figure 5: Model Description

5 Model Training

After Feature Generation and applying Correlation we are feeding data to two kinds of RNN + LSTM models namely Specialized model and Generalized Model.

5.1 Specialized Model

What Specialized model does is, it emphasizes more on a single stock (i.e. SBIN in our case) making our model more specific to predict the values of SBI stock only .Here we are not using that generalization factor.After features are generated using linear regression we are feeding those features to RNN + LSTM to get specialized result only for SBI stock data.We have used an Adam optimizer with parameter values as :

$\beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10 - 7, \eta = 0.001.$

Model:

Batch size : 256

Optimizer : Adam

Loss Function : MSE

Dataset : SBI

Epoch : 150

We have trained this model using SBI Stock data and predicted for test data set split and above results.

Results achieved:

Test Data RMSE: 0.06293

Train Data MSE : 0.0012

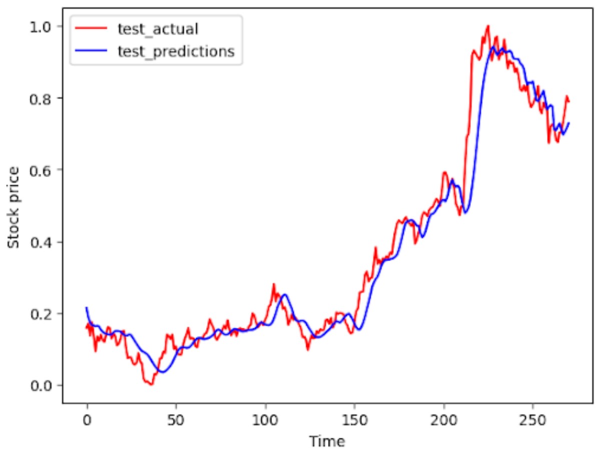


Figure 6: Test Predictions

5.2 Generalized Model

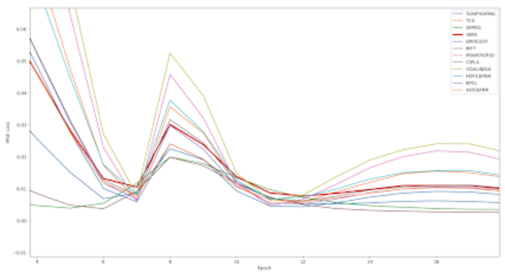
After the feature generation phase we are calculating the Pearson correlation of each stock with SBI and taking a softmax of these values such that we can use this information to include only that much contribution of a stock in SBI's prediction as much as we get its correlation with SBIN.Considering a factor k for generalizing our model (i.e. 1:k ratio is used for Generalization).As we increase k we will be getting a more generalized model such that other stock's variance will be contributing more in our target stock's variability. In this model we have to feed all stocks data for training and for testing we only feed target stock's data(SBIN in our case).

6 Analysis

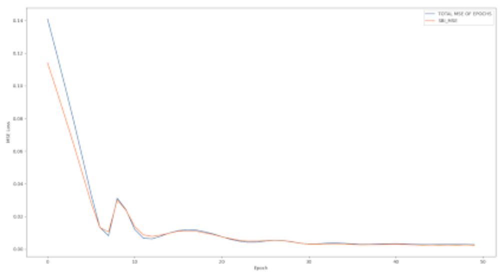
6.1 Equally Specialized and Generalized Model

Model Parameters:
Batch size : 256
Optimizer : Custom Adam Optimizer
Loss Function : MSE
Dataset : All Available Stocks in our dataset
Generalization ratio : 1:1
Epochs : 50
Below is the table for MSE for individual stocks training loss of different stocks individually:

EPOCH 1 - MSE for individual stocks	**EPOCH 50 - MSE for individual stocks**
Training loss(SUNPHARMA) : 0.070949949	Training loss(SUNPHARMA) : 0.002489584
Training loss(TCS) : 0.202458158	Training loss(TCS) : 0.003304091
Training loss(WIPRO) : 0.014483192	Training loss(WIPRO) : 0.000765817
Training loss(SBIN) : 0.114078894	Training loss(SBIN) : 0.002449574
Training loss(DRREDDY) : 0.134438484	Training loss(DRREDDY) : 0.002050151
Training loss(INFY) : 0.028960427	Training loss(INFY) : 0.000831711
Training loss(POWERGRID) : 0.264529109	Training loss(POWERGRID) : 0.003989082
Training loss(CIPLA) : 0.133931800	Training loss(CIPLA) : 0.003185378
Training loss(COALINDIA) : 0.300816088	Training loss(COALINDIA) : 0.004780080
Training loss(HDFCBANK) : 0.181554675	Training loss(HDFCBANK) : 0.003552331
Training loss(BPCL) : 0.127414822	Training loss(BPCL) : 0.002733220
Training loss(AXISBANK) : 0.117133319	Training loss(AXISBANK) : 0.002634565



((a)) MSE error for individual stocks



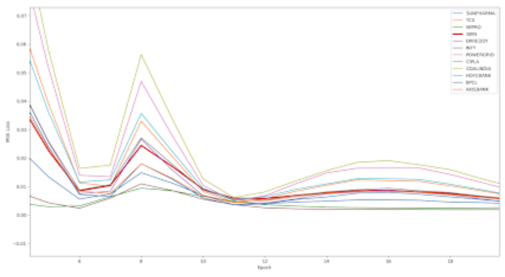
((b)) MSE SBI and overall model

Figure 7: MSE error

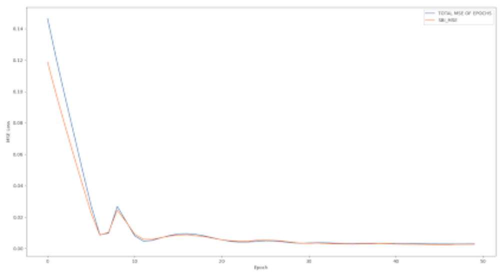
6.2 More Generalized Model

Model Parameters:

Batch size : 256
Optimizer : Custom Adam Optimizer
Loss Function : MSE
Dataset : All Stocks in our dataset
Generalization ratio : 1:3
Epoch : 50
Below is the table for MSE for individual stocks training loss of different stocks individually:
Epoch 50 - MSE for individual stocks
Training loss(SUNPHARMA) : 0.0025505803
Training loss(TCS) : 0.0037160136
Training loss(WIPRO) : 0.0008235373
Training loss(SBIN) : 0.0026573899
Training loss(DRREDDY) : 0.0020819921
Training loss(INFY) : 0.0009117067
Training loss(POWERGRID) : 0.0046091396
Training loss(CIPLA) : 0.0034590598
Training loss(COALINDIA) : 0.0054432083
Training loss(HDFCBANK) : 0.0036973648
Training loss(BPCL) : 0.0026982298
Training loss(AXISBANK) : 0.0025992493



((a)) MSE error for individual stocks



((b)) MSE SBI and overall model

Figure 8: MSE error

7 Results

Below are the results achieved by our Generalized model after training it for 50 Epochs

7.1 Equally Generalized Model

Batch size : 256
Optimizer : Custom Adam Optimizer
Loss Function : MSE
Dataset : All Available Stocks in our dataset

Generalization ratio : 1:1
Results achieved:
Test Data RMSE : 0.08639936655296294
Train Data RMSE : 0.03785213249684822

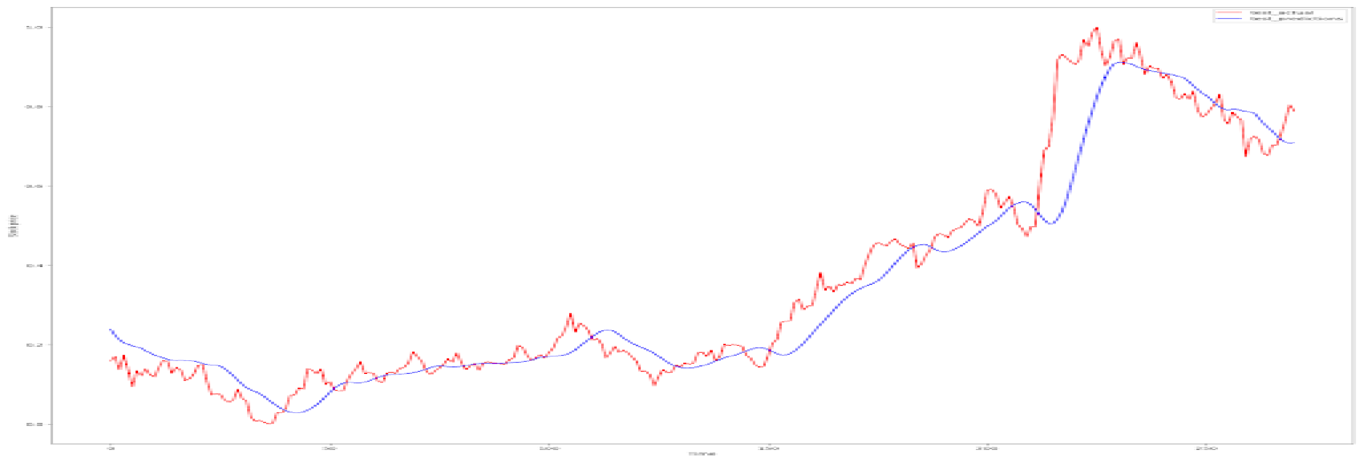


Figure 9: Test Predictions

7.2 More Generalized Model

Batch size : 256
Optimizer : Custom Adam Optimizer
Loss Function : MSE
Dataset : All Available Stocks in our dataset

Generalization ratio : 1:3
Results achieved:
Test Data RMSE : 0.07673
Train Data RMSE : 0.03097

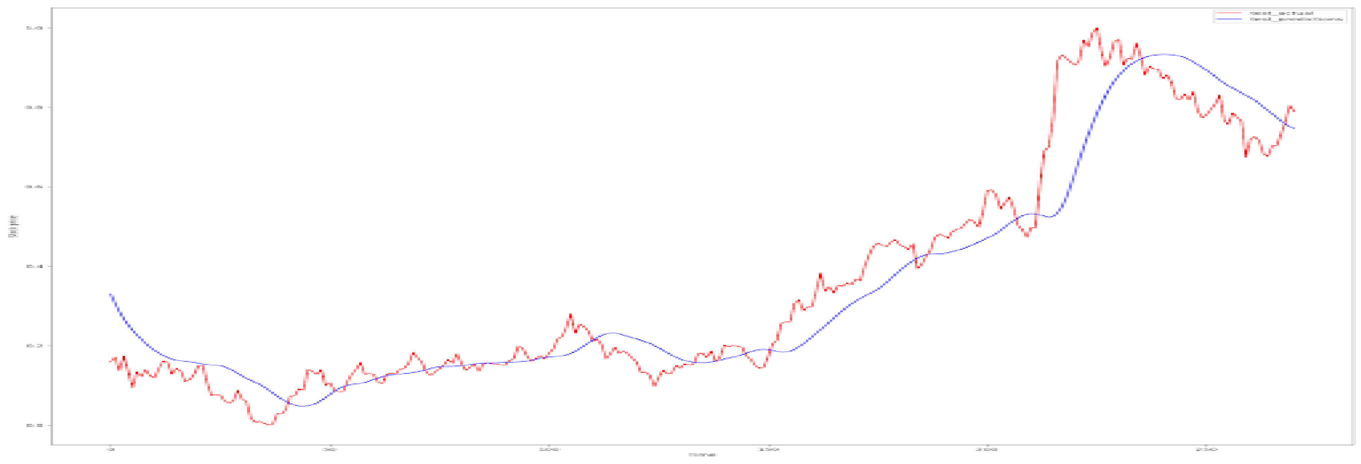


Figure 10: Test Predictions

8 Conclusion

We trained both RNN-LSTM models using custom optimizers.At 50 epoch we got RMSE for 1:3 generalized model better than 1:1 model (i.e. RMSE) but not better than the specialized model.We are getting some good results but this is not enough to prove our claims as we are not getting results that can beat the Specialized model.Hence the results are not fully conclusive.

9 Bibliography

[1] https://kgptalkie.com/google-stock-price-prediction-using-rnn-lstm/

[2] https://aditi-mittal.medium.com/understanding-rnn-and-lstm-f7cdf6dfc14e

[3] https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-f

[4] https://machinelearningmastery.com/tour-of-ensemble-learning-algorithms/

10 GitHub Link

Here is the github code link:
https://github.com/sks-shubham/Stock-Market-Predictor