A Robust Predictive Model for Stock Price Forecasting

Jaydip Sen
Professor
Praxis Business School
Kolkata, INDIA
jaydip@praxis.ac.in

Tamal Datta Chaudhuri
Professor
Calcutta Business School
Kolkata, INDIA
tamalc@calcuttabusinessschool.org

Abstract

Prediction of future movement of stock prices has been the subject matter of many research work. On one hand, we have proponents of the *Efficient Market Hypothesis* who claim that stock prices cannot be predicted accurately. On the other hand, there are propositions that have shown that, if appropriately modelled, stock prices can be predicted fairly accurately. The latter have focused on choice of variables, appropriate functional forms and techniques of forecasting. This work proposes a granular approach to stock price prediction by combining statistical and machine learning methods with some concepts that have been advanced in the literature on technical analysis. The objective of our work is to take 5 minute daily data on stock prices from the National Stock Exchange (NSE) in India and develop a forecasting framework for stock prices. Our contention is that such a granular approach can model the inherent dynamics and can be fine-tuned for immediate forecasting. Six different techniques including three regression-based approaches and three classification-based approaches are applied to model and predict stock price movement of two stocks listed in NSE - Tata Steel and Hero Moto. Extensive results have been provided on the performance of these forecasting techniques for both the stocks.

Keywords: *Stock Price Prediction, Multivariate Regression, Logistic Regression, Decision Tree, Artificial Neural Networks.*

1 INTRODUCTION

Prediction of future movement of stock prices has been the subject matter of many research work. On one hand, we have proponents of the efficient market hypothesis who claim that stock prices cannot be predicted. On the other hand, there are work that have shown that, if correctly modelled, stock prices can be predicted with a fairly reasonable degree of accuracy. The latter have focused on choice of variables, appropriate functional forms and techniques of forecasting. Sen & Datta Chaudhauri proposed a novel approach of stock price forecasting based on a time series decomposition approach of the stock prices time series (Sen & Datta Chaudhuri, 2016a; Sen & Datta Chaudhuri, 2016b; Sen & Data Chaudhuri 2017c; Sen & Datta Chaudhuri, 2017d).

There is also an extent of literature on technical analysis of stock prices where the objective is to identify patterns in stock movements and profit from it. The literature is geared towards making money from stock price movements, and various indicators like Bollinger Band, Moving Average Convergence Divergence (MACD), Relative Strength Index (RSI), Moving Average, Momentum Stochastics, Mesa Sine Wave etc. have been devised towards this end. There are also patterns like Head and Shoulders, Triangle, Flag, Fibonacci Fan, Andrew's Pitchfork etc. which are extensively used by traders for gain. This approach provides visual manifestations of the indicators which helps the ordinary investor to understand which way stock prices may move.

In this paper, we propose a granular approach to stock price prediction by combining statistical and machine learning methods of prediction on technical analysis of stock prices. We present several approaches for short term stock price movement forecasts using various classification and regression techniques and compare their performance in prediction of stock price movement in a very short interval of time. This approach will provide several useful information to the investors in stock market who are particularly interested in short-term investments for profit.

The rest of the paper is organized as follows. In Section 2, we present a clear statement of our problem at hand. Section 3 provides a brief review of the literature on stock price movement modelling and prediction. In Section 4, we present a detailed discussion on the methodology that we have followed in this work. Section 5 describes the details of all the predictive models built in this work and the results they have produced. A comparative analysis has also been presented on the performance of the models. Finally Section 6 concludes the paper.

2 PROBLEM STATEMENT

The objective of our work is to take stock price data at 5 minutes interval from the National Stock Exchange (NSE) in India and develop a robust forecasting framework for the stock price movement. Our contention is that such a granular approach can model the inherent dynamics and can be fine-tuned for immediate forecasting of stock price or stock price movement. We are not looking at forecasting of long-term movement of stock prices. Rather, our framework will be more relevant to a trade-oriented framework.

At any point of time in the Indian economy, given the appetite of financial market players including individuals, domestic institutions and foreign financial institutions, there is a finite amount of fund that are deployed in the stock market. This amount discounts the entire macroeconomics of that time. This fund would be distributed among various stocks. Thus, in the very short run, ,if some stock prices are rising, some other stock prices should be falling. In this work, we propose a framework where it will be possible to predict price of a stock in the next slot of time, given the historical movement pattern of the stock. The approach builds in indicators like momentum, pivot points and range, all based on daily data on stock prices at 5 minutes interval of time.

3 RELATED WORK

The literature trying to prove or disprove the efficient market hypothesis can be classified in three strands according to choice of variables and techniques of estimation and forecasting. The first strand consists of studies using simple regression techniques on cross sectional data (Basu, 1983; Jaffe et al., 1989; Rosenberg et al., 1985; Fama & French, 1995; Chui & Wei 1998). The second strand of the literature has used time series models and techniques to forecast stock returns following economic tools like Autoregressive Integrated Moving Average (ARIMA), Granger Causality Test, Autoregressive Distributed Lag (ARDL) and Quantile Regression to forecast stock prices (Jarrett & Kyper, 2011; Adebiyi et al., 2014; Mondal et al., 2014; Mishra, 2016). The third strand includes work using machine learning tools for prediction of stock returns (Mostafa, 2010; Dutta et al., 2006; Wu et al., 2008; Siddiqui & Abdullah, 2015; Jaruszewicz and Mandziuk, 2004).

4 METHODOLOGY

In Section 2, we have mentioned that the goal of this work is to develop a robust forecasting framework for short-term price movement of stocks. We use the Metastock tool (Metastock Website) for collecting data on short-tem price movement of stocks. Particularly, we collected the stock data for two companies - Tata Steel and Hero Moto Corp. The data is collected at every 5 minutes interval in a day, for all the days in which the National Stock Exchange (NSE) was operating during the years 2013 and 2014. The raw data for each stock consisted of the following variables: (i) Date, (ii) Time, (iii) Open value of the stock, (iv) High value of the stock, (v) Low value of the stock, (vi) Close value of the stock, and (viii) the Volume of the stock traded in a given interval. The variable *Time* refers to the time instance at which the stock values are noted as each record is collected at 5 minutes interval of time. Hence, the time interval between two successive records in the raw data was 5 minutes. The raw data in this format is collected for two stocks - Tata Steel and Hero Moto Corp- for two years. In addition to the six variables in the raw data that we have mentioned, we collected also the NIFTY index at 5 minutes interval for the same period of two years in order to capture the overall market sentiment at each time instant so that more accurate and robust forecasting can be made using the combined information of historical stock prices and the market sentiment index. Therefore, the raw data for both the stocks now consists of seven variables. Since 5 minutes interval is too granular, we make some aggregation of the raw data. We break the total time interval in a day into three slots as follows: (1) *morning* slot that covers the time interval 9:00 AM till 11:30 AM, (2) *afternoon* slot that covers the time interval 11:35 AM till 1:30 PM and (3) *evening* slot that covers the time interval 1:35 PM till the time of closure of NSE in a given day. Hence, the daily stock information now consists of three records, each record containing stock price information for a time slot.

Using the seven variables in the raw data and incorporating the aggregation of data using time slots, we derive the following variables that we use, later on, in our forecasting models. We have used two approaches in forecasting - regression and classification. The two approaches involved a little differences in some of the variables, which we will describe and explain at appropriate point.

The following eleven variables are derived and used in our forecasting models:

a) Month : it refers to the month to this a given record belongs. This variable is coded into a numeric data, with "1" referring to the month of January and "12" referring to the month of December. The value of the variable "Month" lies in the range [1, 12].

b) Day_Month : this variable refers to the day of the month to which a given record belongs. It is a numeric variable lying within the range [1, 31]. For example the date 14th February 2013, will have a value 14 against the variable "Day_Month".

c) Day_Week : it is numeric variable that refers to the day of the week corresponding to a given stock record. This variables lies in the range [1, 5]. Monday is coded as 1, while Friday is coded as 5.

d) Time : it is a numeric variable that refers to the time slot to which a given record belongs. The *morning*, *afternoon* and *evening* slots are coded as 1, 2 and 3 respectively. Thus if a stock record is noted at time instance 2:45 PM, the value of the "Time" variable corresponding to that record would be 3.

e) Open_Perc : it is a numeric variable that is computed as a percentage change in the value of the *Open* price of the stock over two successive time slots. The computation of the variable is done as follows. Suppose, we have two successive slots: S_1 and S_2. Both of them consist of several records at 5 minutes interval of time. Let the *Open* price of the stock for the first record of S_1 is X_1 and that for S_2 is X_2. The Open_Perc for the slot S_2 is computed as $(X_2 - X_1)/X_1$ in terms of percentage.

f) Sensex_Perc : it is a numeric variable that is computed as a percentage change in the NIFTY index over two successive time slots. The computation of the variable is done as follows. We compute the means of the NIFTY index values for two successive time slots S_1 and S_2. Let us assume the means are M_1 and M_2 respectively. Then the Sensex_Perc for the slot S_2 is computed as $(M_2 - M_1)/M_1$ in terms of percentage.

g) Low_Diff : it is a numeric value that is computed as the difference between the *Low* values of two successive slots. For two successive slots S_1 and S_2, first we compute the mean of all *Low* values of the records in both the slots. If L_1 and L_2 refers to the mean of the *Low* values for S_1 and S_2 respectively, then Low_Diff for S_2 is computed as $(L_2 - L_1)$.

h) High_Diff : it is a numeric value that is computed as the difference between the *High* values of two successive slots. The computation is identical to that of Low_Diff except for the fact that *High* values are used in this case.

i) Close_Diff : it is a numeric value that is computed as the difference between the *Close* values of two successive slots. It computation is similar to the Open_Perc variable, except for the fact that we use the *Close* values in the slots and we don't compute any percentage. Hence, if two successive slots S_1 and S_2 have close values C_1 and C_2 respectively, then Close_Diff for S_2 is computed as $(C_2 - C_1)$.

j) Vol_Diff : it is a numeric value that is computed as the difference between the *Volume* values of two successive slots. For two successive slots S_1 and S_2, we compute the mean values of *Volume* for both the slots, say V_1 and V_2 respectively. Now, the Vol_Diff for S_2 is computed as $(V_2 - V_1)$.

k) Range_Diff : it is a numeric value that is computed as the difference between the *Range* values of two successive slots. For two successive slots S_1 and S_2, suppose the *High* and *Low* values are H_1, H_2, L_1 and L_2 respectively. Hence, the *Range* value for S_1 is $R_1 = (H_1 - L_1)$ and for S_2 is $R_2 = (H_2 - L_2)$. The Range_Diff for the slot S_2 is computed as $(R_2 - R_1)$.

After, we compute the values of the above eleven variables for each slots for both the stocks for the time frame of two years (i.e., 2013 and 2014), we develop the forecasting framework. As mentioned earlier, we followed two broad approach in forecasting of the stock movements - Regression and Classification.

In regression approach, based on the historical movement of the stock prices we predict the stock price in the next slot. We use Open_Perc as the response variable, which is a continuous numeric variable. The objective of the regression technique is to predict the Open_Perc value of the next slot given the stock movement pattern and the values of the predictors till the previous slot. In other words, if the current time slot is S_1, the regression techniques will attempt to predict Open_Perc for the next slot S_2. If the predicted Open_Perc is positive, then it will indicate that there is an expected rise the stock price in S_2, while a negative Open_Perc will indicate a fall in the stock price in the next slot. Based on the predicted values, an potential investor can make his/her investment strategy in stocks.

In the classification approach, the response variable Open_Perc is a discrete variable belonging to one among two or more classes. For developing the classification-based forecasting approaches, we converted Open_Perc into a categorical variable that takes one of the two values 0 and 1. The value "0" indicating negative Open_Perc values and "1" indicating positive Open_Perc values. Hence, if the current slot is S_1 and if the forecast model expects a rise in the Open_Perc value in the next slot S_2, then the Open_Perc value for S_2 will be 1. An expected negative value of the Open_Perc in the next slot will be indicated by a 0 value for the response variable.

For both classification and regression approaches, we experimented with three cases which are described below.

Case I: We used the data for the year 2013 which consisted of 19, 385 records at five minutes interval. These records were aggregated into 745 time slot records for building the predictive model. We used the same dataset for testing the forecast accuracy of the models for both the stocks and made a comparative analysis of all the models.

Case II : We used the data for the year 2014 which consisted of 18, 972 records at five minutes interval. These granular data were aggregated into 745 time slot record for building the predictive model. We used the same dataset for testing the forecast accuracy of the model for both the stocks and carried out an analysis of the performance of the predictive models.

Case III: We used that data for 2013 as the training dataset for building the models and test the models using the data for the year 2014 as the test dataset. We, again, carried out an analysis on the performance of different models in this approach.

We used three approaches to classification and three approaches to regression for building our forecasting framework. The classification techniques used were: (i) Logistic Regression, (ii) Random Forest and (iii) Support Vector Machine (SVM). For measuring accuracy and effectiveness in these approaches, we used several metrics such as, sensitivity and specificity, positive predictive value and negative predictive value.

The three regression methods that we used are: (i) Multivariate Regression, (ii) Artificial Neural Network (ANN), and (iii) Decision Tree. For comparing their performance, we used several metrics such as *root mean square error* (RMSE), and correlation coefficient between the actual and predicted values of the response variable, e.g., Open_Perc.

5 ANALYSIS OF RESULTS

In this section, we provide a detailed discussion on the forecasting techniques that we have used and the results obtained using those techniques. We first discuss the three regression techniques and then the classification techniques.

5.1 Multivariate Regression

In this regression approach, using Open_Perc as the response variable and the remaining ten variables as the predictors, we built a predictive model for three cases mentioned in Section 4. We discuss the three cases separately. In all these cases, we use the programming language R for data management, model construction, testing of models and visualization of results.

Case I: We use 2013 data as the training data set for building the model and test the model using the same data set. For both the stocks, we used two approaches of multivariate regression - (i) *backward deletion* and (ii) *forward addition* of variables. Both the approaches yielded the same results for both the stocks.

For Tata Steel data set for the year 2013, we applied the *vif* function in the *farway* library to detect the collinear variable in order to get rid of the multicollinearity problem. It was observed that the variables Low_Diff, High_Diff and Close_Diff exhibited multicollinearity. We retained the variable Close_Diff and left out the other two variables. Using the *drop1* function in case of the *backward deletion* technique, and *add1* in case of the *forward addition* technique, we identified the variables that were not significant in the model and did not contribute to the information content of the model. For identifying the variables that contributed least to the information content in the model at each iteration, we used the Akaike Information Criteria (AIC) - the variable that had least AIC value and non-significant *p*-value at each iteration, was removed from the model, in case of backward deletion process. On the other hand, the variable that had the lowest AIC and a significant *p*-value was added to the model at each iteration for the forward addition technique. It was found that Close_Diff and Sensex_Perc are the two predictors which were finally retained in the model.

For Hero Moto for the year 2013, the variables Sensex_Perc, Close_Diff and Vol_Diff were found to be significant for the model.

For testing the prediction accuracy the models built for both the stocks, we used the *predict* function for forecasting the values of the response variables. We computed the correlation coefficient between the *actual* Open_Perc values and the *predicted* Open_Perc values. We also computed the RMSE among the actual and predicted values, and hence derived the percentage of the RMSE with respect to the mean of the *actual* Open_Perc values. The ratio of the RMSE with respect to the mean of the *actual* Open_Perc values in percentage provides a very useful metric for measuring the accuracy of a predictive model.

Case II : For Tata Steel, in this model, the predictors Close_Diff, Sensex_Perc and Day_week were found to be significant using both the backward deletion and forward addition technique. However, for Hero Moto, the significant variables were Sensex_Perc, Low_Diff and Vol_Diff. We computed correlation between the actual and predicted Open_Perc values and also the ratio of the RMSE between these values with respect to the mean of the actual values of Open_Perc.

Case III: In this case the model is identical to that in Case I. However, since the test data is different here, we will find different values for the correlation coefficient and RMSE.

Table 1 Multivariate regression results

Model **Stock**	**Model I**		**Model II**		**Model III**	
	Case 1: Training Data 2013 Test Data 2013		Case II: Training 2014 Test Data 2014		Case III: Training Data 2013 Test Data 2014	
Tata Steel	Correlation	**0.97**	Correlation	**0.98**	Correlation	**0.98**
	RMSE/Mean of Actual	**32.40**	RMSE/Mean of Actual	**4.01**	RMSE/Mean of Actual	**24.08**
Hero Moto	Correlation	**0.56**	Correlation	**0.84**	Correlation	**0.53**
	RMSE/Mean of Actual	**116.23**	RMSE/Mean of Actual	**116.03**	RMSE/Mean of Actual:	**116.03**

5.2 Artificial Neural Networks

We have built ANN models for all three cases for the both the stocks using the *neuralnet* function defined in the *neuralnet* package in R. For Tata Steel, for all the three cases, we needed only one node in the hidden layer in order to achieve an extremely high prediction accuracy. From Table 2, it may be observed that ANN regression method has produced high accuracy in prediction in all three cases. However, for Hero Moto Corp, the prediction task appeared to be more challenging. For Case I in Hero Moto, we used two nodes in the hidden layer to achieve a correlation value of 0.73 while one node in the hidden layer yielded a correlation value of 0.69. Further increase in the number of nodes in the hidden layer resulted in a fall in the correlation value and increase in the RMSE value. Hence, we chose two nodes in the hidden layer for Case I (i.e., Model I) of Hero Moto Corp. The number of nodes in the hidden layer was set at two using the parameter "hidden" = 2 in the *neuralnet* function. For Case II in Hero Moto, we needed three hidden nodes to arrive at a decent correlation value of 0.75. For Case III in Hero Moto, we applied the model built in Case I with two hidden nodes to predict on 2014 data. The task appeared to be quite challenging and the prediction was poor, yielding a correlation value of 0.45 and the percentage of RMSE to the mean of the actual values of 50.32. As an illustration of a case, we have presented the ANN model of Case II for Hero Moto in Figure 1.

Table 2 ANN regression results

Model Stock	Model I		Model II		Model III	
	Case 1: Training Data 2013 Test Data 2013		Case II: Training 2014 Test Data 2014		Case III: Training Data 2013 Test Data 2014	
Tata Steel	Correlation	**0.98**	Correlation	**0.98**	Correlation	**0.98**
	RMSE/Mean of Actual	**5.16**	RMSE/Mean of Actual	**5.16**	RMSE/Mean of Actual	**8.61**
Hero Moto	Correlation	**0.73**	Correlation	**0.75**	Correlation	**0.45**
	RMSE/Mean of Actual	**18.30**	RMSE/Mean of Actual	**12.94**	RMSE/Mean of Actual:	**50.32**

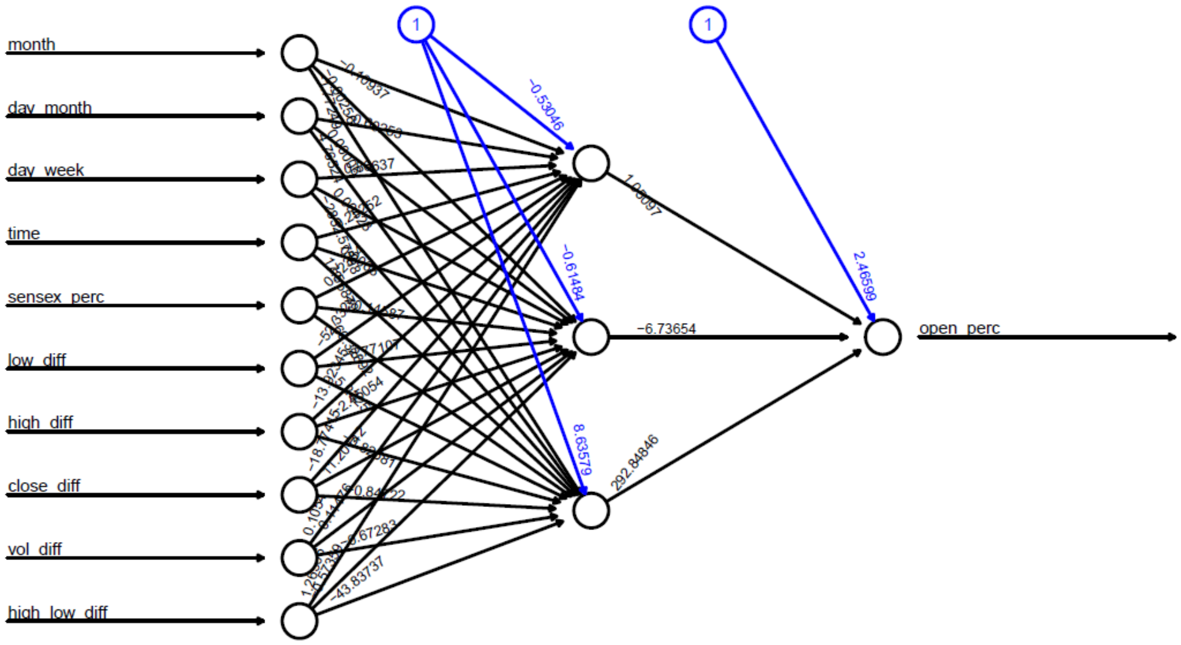

Figure 1 ANN model with three hidden node for Hero Moto Model II

5. 3 Decision Tree

We have used the *tree* function in the *tree* library in R to carry out regression using Decision Tree approach. For Tata Steel, this approach produced 15 nodes including 9 leaf nodes for all the three cases Case I, Case II and Case III. The functions *cor* and *rmse* defined in the library Metrics are used to compute the correlation coefficient and the RMSE value for computing the prediction accuracy of the predictive tree models. Table 3 presents the results that indicate the method worked extremely well in all cases for both the models. As an illustration, Figure 2 depicts the Decision Tree model for Case II of Tata Steel.

Table 3 Decision Tree regression results

Model **Stock**	**Model I** Case 1: Training Data 2013 Test Data 2013		**Model II** Case II: Training 2014 Test Data 2014		**Model III** Case III: Training Data 2013 Test Data 2014	
Tata Steel	Correlation RMSE/Mean of Actual	**0.96** **30.12**	Correlation RMSE/Mean of Actual	**0.97** **33.65**	Correlation RMSE/Mean of Actual	**0.96** **33.72**
Hero Moto	Correlation RMSE/Mean of Actual	**0.98** **30.92**	Correlation RMSE/Mean of Actual	**0.97** **33.65**	Correlation RMSE/Mean of Actual:	**0.96** **32.72**

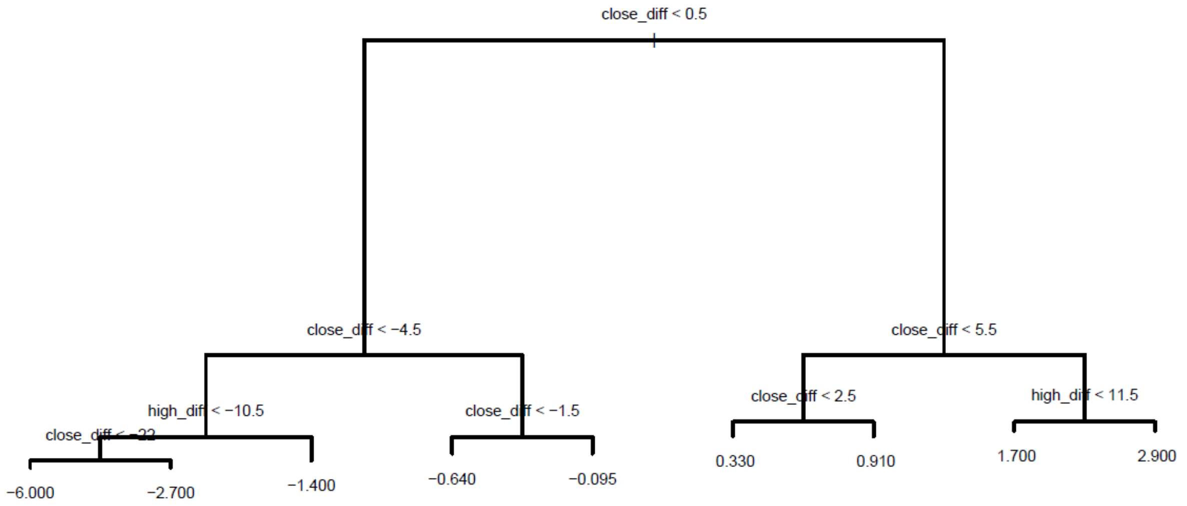

Figure 2 Decision Tree model for regression for Tata Steel Model II

5.4 Logistic Regression

This being a classification technique, we transformed the response variable Open_Perc to a discrete domain from continuous domain. In other words, we transformed the response variable into a categorical variable that can assume values "0" or "1". We converted all negative or zero values of Open_Perc to the class "0" and all non-zero positive values to class "1". We used the function *glm* in R for building the logistic regression model with three parameters being passed in the function: (i) the first parameter is the *formula* which is Open_Perc ~. to include Open_Perc as the response variable and all the remaining variables as the predictors, (ii) the second parameter is "family = binomial" indicating that model is a binary logistic regression that involves two classes, and (iii) the third parameter is the R data object containing the training data set. We used the *predict* function in R to compute the probability of the test records to belong to the two classes. We assumed a threshold value of 0.5 as the probability. In other words, when the probability of a record belonging to a class exceeds 0.5 we assume that record belongs to that class.

For both the stocks and for all the three cases, we computed the prediction accuracy of the models in the form of confusion matrices. Table 4 presents the results. We have used certain well known terms for measuring the performance of the classification techniques. We define them below.

Sensitivity: It is the ratio of the true positives to the total number of positives in the test dataset. Here, "positive" refers to the records belonging to the class "1". The term "true positives" refers to the number of positive cases that the model correctly identified.

Specificity: It is the ratio of the true negatives to the total number of negatives in the test dataset. Here "negative" refers to the records belonging to class "0". The term "true negatives" refers to the number of negative cases that the model correctly identified.

Positive Predictive Value (PPV): It is the ratio of the number of true positives to the sum of the true positive cases and false positive cases.

Negative Predictive Value (NPV): It is the ratio of the number of true negative cases to the sum of the true negative cases and false negative cases.

Table 4 Logistic Regression classification results

Stock	Metrics	Model I	Model II	Model III
		Case I : Training 2013 Test Data 2013	Case II : Training 2014 Test Data 2014	Case III : Training 2013 Test Data 2014
Tata Steel	Sensitivity	93.00	94.20	94.20
	Specificity	97.25	98.65	93.63
	PPV	92.54	96.53	85.33
	NPV	97.43	97.71	97.59
Hero Moto	Sensitivity	71.43	59.47	71.98
	Specificity	84.01	91.31	67.95
	PPV	59.09	75.84	47.30
	NPV	90.10	83.77	85.85

5.5 Random Forest

The random forest is a way to combine information across an ensemble. If we assume that each of the *k* classifiers in the ensemble is a decision tree for classifying a new element (i.e., record) into one of the *m* possible outcome groups. In the problem at hand, the value of *m* is 2. At each node, an individual decision tree determines the split on the basis of a smaller, random selection of attributes, and not from the set of all attributes. Each tree in the forest then votes on the classification of a new item, and the most popular class is returned as the ensemble solution. We have used the *randomForest* function defined in the *randomForest* library in R for carrying out classification in the stock movement pattern for both the stocks in all the three cases. After building the random forest model, the function *predict* was used to predict the class to which a particular record belonged.

5.6 Support Vector Machine

Support Vector Machine (SVM) is a classification techniques that can cover both linear and non-linear classification problems. SVM transforms the data into a higher dimensional space so that data is *linearly separable*. SVM achieves this by finding *support vectors* that identify the data points on the *hyperplane* with the highest margin so that linear separation of the data points is possible. We have used *ksvm* function defined in the *kernlab* package in R for carrying out robust classification of data points so that reliable and accurate forecasting can be carried on stock price movement pattern. The *ksvm* function takes three parameters. The first parameter is the formula: Open_Perc~., which specifies Open_Perc as the response variable and the remaining variables as the predictors. The second parameter is the R data object containing the training dataset, and the third parameter is optional *kernel* parameter, which we used as kernel = "vanilladot". Table 6 represents the results on the forecasting accuracy of the SVM models.

Table 5 Random Forest classification results

Stock	**Metrics**	**Model I** Case I : Training 2013 Test Data 2013	**Model II** Case II : Training 2014 Test Data 2014	**Model III** Case II : Training 2013 Test Data 2014
Tata Steel	Sensitivity	95.00	95.17	92.75
	Specificity	97.06	99.23	98.46
	PPV	92.23	98.01	96.00
	NPV	98.14	98.09	97.14
Hero Moto	Sensitivity	37.36	46.86	95.00
	Specificity	88.45	88.03	97.06
	PPV	51.13	60.00	92.23
	NPV	81.37	80.00	98.14

Table 6 SVM classification results

Stock	**Metrics**	**Model I** Case I : Training 2013 Test Data 2013	**Model II** Case II : Training 2014 Test Data 2014	**Model III** Case III : Training 2013 Test Data 2014
Tata Steel	Sensitivity	91.26	95.17	85.22
	Specificity	97.77	99.23	97.78
	PPV	94.00	98.01	94.69
	NPV	96.70	98.01	93.44
Hero Moto	Sensitivity	64.71	72.00	71.67
	Specificity	78.53	78.40	75.34
	PPV	18.13	34.78	20.77
	NPV	96.80	94.59	96.72

Analysis of the Results: We make the following brief analysis of the results. For the regression techniques, we find that while multivariate regression is quite effective in modelling and prediction of stock price movements for all the three cases for the Tata Steel stock, it is not at all effective for any of the cases for the Hero Moto stocks. ANN is found to be extremely effective in all the cases for the Tata Steel stock and for the Case I and Case II of the Hero Moto stock. However, for Case III of Hero Moto stock, ANN is found to have performed very poorly. Decision Tree-based regression has produced very high level of prediction accuracy for all the cases for both the stocks, and has been found to be the most effective regression method. It has also been observed that for all cases, the movement patterns in the Hero Moto stock posed greater challenge compared to the Tata Steel stock in modelling and prediction of stock prices.

Among the classification techniques, Logistic Regression method has been found to have produced excellent results for the Tata Steel stock for all the three cases. However, for the Hero Moto stock, the sensitivity and PPV of Logistic Regression are found to be unsatisfactory. Case III for the Hero Moto stock has posed a particular challenge to the Logistic Regression technique. The performance of the Random Forest for classification has been excellent for all the cases for the Tata Steel stock, which was expected. However, quite surprisingly, Random Forest classification has also produced extremely high level of prediction accuracy for the Model III of Hero Moto stock, while it has performed quite poorly in the other two cases for the same stock, even though the other two cases are apparently easier modelling situations. In general, it is also observed that the sensitivity of Random Forest has been poor for both Case I and Case II of the Hero Moto stock. The performance of SVM has been extremely accurate for all the cases of the Tata Steel stock. However, the sensitivity and PPV for all the cases for Hero Moto stock have been found to be very poor for SVM. Considering all these observations, we conclude that Random Forest-based classification produced the most accurate modelling and prediction of the two stocks that we studied.

6 CONCLUSION

In this paper, we have proposed a framework of stock price movement prediction in the short-term time period using three classification and three regression based approaches. We tested our predictive models on two different stocks - Tata Steel and Hero Moto - under three different cases. The raw data on the stock price movements at five minutes interval were collected using the Metastock tool for the period January 2013 till December 2014. The raw data was suitably transformed, and several predictor variables and the response variable were identified for both the regression and classification-based predictive models. Extensive results were presented on the performance of the six different approaches of prediction. It was observed that while among the classification techniques Random Forest yielded the highest level of accuracy in prediction, Decision Tree-based regression models produced the least error in modelling and prediction of stock price movements.

REFERENCES

Adebiyi A., Adewumi A. O. and Ayo C. K. (2014). Stock price prediction using the ARIMA model. *Proceedings of the International Conference on Computer Modelling and Simulation*, Cambridge, UK, pp. 105-111.

Basu S. (1983). The relationship between earnings yield, market value and return for NYSE common stocks: Further Evidence. *Journal of Financial Economics*, 12(1), 129-156.

Chui, A. and Wei, K. (1998). Book-to-market, firm size, and the turn of the year effect: Evidence from Pacific basin emerging markets. *Pacific Basin Finance Journal*, 6(3-4), 275-293.

Dutta, G., Jha, P., Laha, A. and Mohan, N. (2006). Artificial neural network models for forecasting stock price index in the Bombay Stock Exchange. *Journal of Emerging Market Finance*, 5(3), 283-295.

Fama, E. F. and French, K. R. (1995). Size and book-to-market factors in earnings and returns. *Journal of Finance*, 50(1), 131-155.

Jaffe J, Keim D. B. and Westerfield R. (1989). Earnings yields, market values, and stock returns. *Journal of Finance*, 44, 135-148.

Jarrett, J. E. and Kyper, E. (2011). ARIMA modeling with intervention to forecast and analyze Chinese stock prices. *International Journal of Engineering Business Management*, 3(3), 53-58.

Jaruszewicz, M. and Mandziuk, J (2004). One day prediction of Nikkei index considering information from other stock markets. *Proceedings of the International Conference on Artificial Intelligence and Soft Computing*, Japan, 1130–1135.

Metastock Website: https://www.metastock.com.

Mishra, S. (2016). The quantile regression approach to analysis of dynamic interaction between exchange rate and stock returns in emerging markets: Case of BRIC nations. *IUP Journal of Financial Risk Management*, 13(1),7-27.

Mondal, P, Shit, L. and Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4, 13-29.

Mostafa, M. (2010). Forecasting stock exchange movements using neural networks: Empirical evidence from Kuwait. *Expert Systems with Application*, 37, 6302-6309.

Rosenberg, B., Reid, K. and Lanstein, R. (1985). Persuasive evidence of market inefficiency. *Journal of Portfolio Management*, 11, 9-17.

Sen, J. and Datta Chaudhuri, T. (2016a). An alternative framework for time series decomposition and forecasting and its relevance for portfolio choice - A comparative study of the Indian consumer durable and small cap sector. Journal of Economics Library, 3(2), 303 - 326.

Sen, J. and Datta Chaudhuri, T. (2016b). An investigation of the structural characteristics of the Indian IT sector and the capital goods sector - An application of the R programming language in time series decomposition and forecasting. Journal of Insurance and Financial Management, 1(4), 68 - 132.

Sen, J. and Datta Chaudhuri, T. (2017a). A time series analysis-based forecasting framework for the Indian healthcare sector. Journal of Insurance and Financial Management, 3(1), 66 - 94.

Sen, J. and Datta Chaudhuri, T. (2017b). A predictive analysis of the Indian FMCG sector using time series decomposition-based approach. Journal of Economics Library, 4(2), 206 - 226.

Siddiqui, T. A. and Abdullah, Y. (2015). Developing a nonlinear model to predict stock prices in India: An artificial neural networks approach. *IUP Journal of Applied Finance*, 21(3), 36-49.

Wu, Q., Chen, Y. and Liu, Z. (2008). Ensemble model of intelligent paradigms for stock market forecasting. *Proceedings of the IEEE 1st International Workshop on Knowledge Discovery and Data Mining*, Washington DC, USA, pp. 205–208.