International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Analysis of the Stock Movement and Prediction of Price of Gold Silver and Crude Oil

Akshay Kumar

Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology, Allahabad

Abstract: ***The stock price varies greatly by the action and the emotion of the people involved in it. They are basically interested in profit. The moods of these people are collected from tweets, articles and blogs. The positive and negative moods affect the stock prices. Using Natural Language Processing and Neural Network we will analyze the blogs and tweets and thus investigate the positive and negative moods and sentiments among the investors and purchasers in stock market. By using this code, SentiWordNet which is a lexical resource for opinion mining. Working of SentiWordNet has an extensive data dictionary.NLTK is a leading platform for building Python programs to work with human language data. It provides user friendly interfaces to about 50 corpora and various resources such as WordNet, along with a suite of text processing libraries for distribution, reasoning, analyzing, tagging, parsing, and semantic reasoning. Neural Network- Neural Network in this method is trained on the basis of features extracted from the Sentiment Classifier.***

Keywords: NLTK, SentiWordNet, Sentiment Classifier

1. Introduction

Emotions in stock market profoundly affect the intellect of the individual, the ability to correlate the stock prices. The companies are greatly affected which has a significant impact on them due to the reaction of people towards a particular stock or management decision about its companies and employees.The moods of the people reacting towards the company (positive and negative) can be collected from various tweets articles published on financial newspaper and other business news channels. The positive and the negative moods can be understood by sentiwordNet. This sentiwordNet predicts the positive and the negative mood and thus can be used to predict the price of the stock few weeks in future. SentiwordNet hence preparesthe company as well as the investors to take best advantage of current market. The country economy is directly proportional to stock market. It's very tough to judge the ups and downs in the stock prices due to the volatile situation created by opinions blogs and forums. These blogs and forums has a great impact on minds of common people, thus determining the stock price commodities such as gold, silver and crude oil.The behavioral finance is directly proportional to the moods of the investors and can be clearly understand that certain movement are driven by moods of investors.The effect of public can be observed in the SentiwordNet score and stock prices of gold silver and crude oil.This is explained in the following figure 1, 2 &3.

2. Literature Survey

Initial thought process

At initial stage of the project we came across various software previously used for opinion mining

1) OPINION FINDER(OF)2 is a publicly available software foranalyzing the sentiments which are used by stock brokers. It can also findout the subjectivity of sentence. To identify the emotional polarity of sentences but what we analyzed id that it focuses mainly on the verb for ex

Sentence	Sentiment
He is a good boy	Positive
He is not a bad boy	Negative

Clearly, this yielded very less efficiency. Also it adheres to a uni dimensional model of mood, making binary distinctions between positive and negative sentiments. This may however ignore the rich, multi-dimensional structure of human mood.

2) **Naive Bayes Classifier:** Then our next approach was to use Naive Bayes Classifier for classifying the stock tweets. The main reason for selecting this was due to its simplicity and the high probability of obtaining good results in most cases. But it assumed class conditional independence and thus suffered from loss of accuracy because practically dependencies exist between variables. Moreover, the training data wasn't vast enough to incorporate all the phrases used in the business world.
3) **Data Dictionary:** Then we made our own data dictionary after manually reading around 900 tweets but soon we realized that they were hardly enough to cover the vast expanse of the English language.
4) **Sentiment Classifier 0.6:** Then we tried sentiment classifier version 0.6.This is an open source package written in python which has been customized to do sentiment analysis for a movie review corpus. While using this software we realized that it gave correct scores for only a few sentences and soon we inferred that its results were not trustworthy. So we had to discontinue its use.

SentiWordNet

After much research we came across SentiWordNet which is a lexical resource for opinion mining. Working of SentiWordNet: SentiwordNet has an extensive data dictionary. For each word, its creators have meticulously assigned positive and negative scores based upon the word's meanings. So whenever some sentence needs to be analyzed, it is fed into a python program which uses SentiWordNet dictionary. The program compares each and every word of the sentence with SentiWordNet dictionary and assigns the designated score. The total score of the sentence is the summation of scores of individual words. SentiwordNet needs input in the form of "the word which needs to be analyzed" followed by Part of speech tag(POS tag) i.e. whether is it a verb, noun, adverb, adjective, conjunction etc. So in order to assign each word a POS tag, we need to use NLTK(the natural language toolkit for python)

NLTK

NLTK is mainly used for making Python software which work with human lingua franca. It is easy to analyzevarious resources such as wordnet, along with the suite of text processing libraries for distribution, reasoning, analyzing, stemming, tagging, parsing, and semantic reasoning.

3. Problem Definition

The error percentage present in the previous software regarding the moods of the stock market was quite high which used to challenge the emotions of stock brokers as a result the software present was not able to give the expected result in this research by applying various methodology and various software like NLTK, SentiWordNet, Neural Network has brought down the error percentage of prediction and real stock market value of the day

4. Methodology

We collected the tweets from various websites like stock tweets,gold digger and various other stock review websites.We use natural Language processing to tokenize, tagging and extracting the positive and the negative sentiments from the collected tweets.We trained the neural network using the results of past two weeks.We then compared the actual price and the predicted price of the stock to find the error percentage.

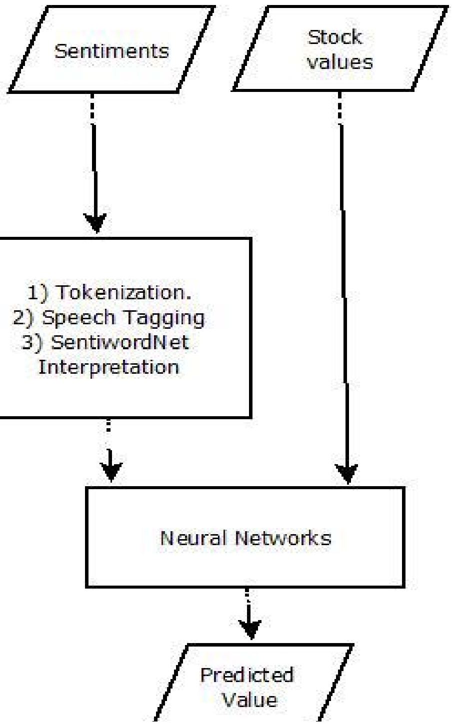

Figure 1: Methodology for our research

The Code:

```
%for prediction
[Nind Nvar]=size(GlobalParams);
trin=[9.5    6.75   7.875  5.625  16.27  5      4.25   4.125  3.625  15.79;
5      4.25   4.125  3.625  15.79  8.625  5      5.625  4.25   16.67;
8.625  5      5.625  4.25   16.67  6.375  5.75   4.625  4.75   16.44;
6.375  5.75   4.625  4.75   16.44  5.875  2.75   4.75   1.5    17.54;
5.875  2.75   4.75   1.5    17.54  6.625  4.25   4.5    3.75   17.03;
6.625  4.25   4.5    3.75   17.03  5.25   3.875  4.75   2.375  16.68

];
inp=size(trin,2);
out=1;
hidden=5;
```

```
[var_x var_y]=size(trin);
for ii=1:var_x
  for jj=1:var_y
    if (mod(jj,5)~= 0)
      trin(ii,jj)=trin(ii,jj)/20;
    else
      trin(ii,jj)=(trin(ii,jj)-10)/20;
    end
  end
end
for i=1:Nind

x=GlobalParams(i,:);
  iw = reshape(x(1:hidden*inp),hidden,inp);
  b1 = reshape(x(hidden*inp+1:hidden*inp+hidden),hidden,1);
  lw = reshape(x(hidden*inp+hidden+1:hidden*inp+hidden+hidden*out),out,hidden);
  b2 = reshape(x(hidden*inp+hidden+hidden*out+1:hidden*inp+hidden+hidden*out+out),out,1);

  y = logsig(logsig(trin*iw'+repmat(b1',size(trin,1),1))*lw'+repmat(b2',size(trin,1),1));
end;

ii=size(y);
for jj=1:ii
  fprintf('%g\n',y(jj)*20+10);
end
```

Figure 2: The code used

Table 1: Table of Silver

Week 1	2.5	8	1.375	6.375	100.83
Week 2	2.75	5	2.375	3.125	103.13
Week 3	3.125	5.75	1.5	2.75	102.09
Week 4	2.375	4.75	1.5	3.5	97.88
Week 5	1.375	3.375	0.625	1.5	97.65
Week 6	1.75	3.75	1.125	2.375	97.35
Week 7	1.75	4.25	1.25	2.2	93.65
Week 8	1.5	5	1	3.125	95.96
Week 9	1.25	1.875	0.875	0.5	92.27
Week 10	1.125	3.125	0.625	0.625	92.42
Week 11	1.875	4.875	0.875	3.375	93.54
Week 12	1.625	3.375	1.25	1.875	89.74
Week 13	4.25	7.125	3.25	4.25	85.82
Week 14	2	4.125	1.125	1.625	82.75
Week 15	1.875	3.25	1	0.875	81.01
Week 16	3.25	6.25	1.875	4.125	80.54
Week 17	2.375	5.125	1.625	3.25	78.65
Week 18	3.875	5.5	2.75	3.75	75.82
Week 19	4	7.5	3.25	4	76.51
Week 20	1.75	4.375	1.375	2.125	66.15
Week 21	2.75	4.625	2	3.125	57.81
Week 22	4.75	6.625	3.25	4.375	56.52
Week 23	2.5	6	1.5	2.75	54.73
Week 24	5.75	3.875	3.875	1.375	52.69
Week 25	5.5	4.5	4	1.375	48.36
Week 26	3.5	5.125	2.625	2.875	48.69
Week 27	4.25	3.5	3.375	1.875	45.59
Week 28	4.25	7.125	2.75	4.75	48.24
Week 29	4	5.25	2.75	2.5	51.69
Week 30	5.5	7.25	3.75	4.875	52.78
Week 31	5.625	4.625	4.75	2.75	50.34
Week 32	7.75	6.5	6.125	3.25	49.76
Week 33	4	5.75	3	3.125	44.84
Week 34	4.25	5	3.25	3.125	45.72
Week 35	4.875	5.125	3.25	2.75	48.87
Week 36	5.5	6.875	4	4.125	49.14
Week 37	6	4.5	3.25	2.5	51.64
Week 38	3	6.25	2.25	5.125	55.74
Week 39	6.75	3.75	4.75	2.625	57.15
Week 40	3.25	4.625	1.75	2.125	59.15
Week 41	4.875	3.625	3.25	1.75	59.39

Table 2: Table of gold

Week 1	5	3.25	3.625	2.375	18.99
Week 2	5.875	6.125	5.125	4.625	19.72
Week 3	5.125	5	3.25	3.375	20.95
Week 4	8	5	5.75	3.25	21.01
Week 5	3.75	4.75	2.5	2.625	20.97
Week 6	8.125	4.75	6.125	3.75	20.84
Week 7	5.25	3.25	4.625	2.5	20.96
Week 8	4	2.75	3.5	2.5	20.55
Week 9	4.875	3.75	3.5	1.75	19.9
Week 10	5	6.625	3.375	4.75	19.49
Week 11	12.125	8.5	9.375	6.75	19.36
Week 12	6.125	3.625	4.375	3.125	19.4
Week 13	5.875	2.625	4.875	1.625	18.89
Week 14	4.5	5.625	4.125	4	18.56
Week 15	4.5	2.625	3	1.875	17.7
Week 16	5.5	4.625	4.25	2.375	17.52
Week 17	6.75	2.875	5.5	2.125	17.02
Week 18	4	3.5	2.875	2.5	17.42
Week 19	4.375	4	3.875	2.25	17.19
Week 20	2.25	2.5	2.125	2	17.22
Week 21	6.75	4	4.375	3	15.66
Week 22	6.25	3.375	3.625	2.375	16.05
Week 23	5.75	4.625	4	3.125	16.38
Week 24	6	3.375	5.375	1.75	16.65
Week 25	3.375	2.625	2.625	1.375	16.22
Week 26	5.875	6	4.625	4.125	16.53

Week 27	6.625	4.375	5.125	2.75	15.65
Week 28	3.75	1	2.75	1	15.74
Week 29	8.375	4.25	5.75	2.375	16.35
Week 30	4	3.75	3.5	1.625	17.07
Week 31	3.75	4.5	2	3.125	18.35
Week 32	6.625	3.625	4.5	2.625	16.76
Week 33	4	2.625	3.75	1.25	17.05
Week 34	3.375	3.25	1.75	1.625	17.28
Week 35	6.25	2.875	4	1.375	16.25
Week 36	3.625	3.625	2.75	3	16.41
Week 37	6.5	5.125	4.125	3.375	15.76
Week 38	5	2	3	1.25	15.6
Week 39	5.875	2.625	3.875	2	16.87
Week 40	7.375	3.75	6.375	2.5	16.66
Week 41	7.5	3.25	5.125	1.75	16.44

Table 3: Table of crude oil

Week 1	5	4	2.25	3	1287.7
Week 2	6.125	5.5	1.825	2.125	1293.4
Week 3	5.75	3.625	1.75	3.125	1295
Week 4	11.375	11.625	6.625	9.125	1245.6
Week 5	8.5	3.5	5.625	3	1253.5
Week 6	7	5.875	4.125	4.625	1271.7
Week 7	6.681	2.694	3.181	2.194	1317.6
Week 8	9	5.875	4.25	4.875	1326.4
Week 9	2.5	0.875	1.25	0.875	1323.8
Week 10	7.625	5.5	5.25	3.875	1296.9
Week 11	5	3.625	3.125	2.5	1313.7
Week 12	5.5	4.125	2.25	2.375	1298.3
Week 13	9.75	7.875	3.25	6.25	1308.9
Week 14	7.125	6.5	5.125	6.25	1313.9
Week 15	8	4.25	4.625	2.875	1278.6
Week 16	8.125	6.75	4.75	4.75	1288.7
Week 17	7.375	5.75	4.125	4.625	1246.8
Week 18	16.25	11.125	8.625	9	1233.6
Week 19	7.125	5.375	3	4.5	1221
Week 20	14.125	10	8.125	8	1217.5
Week 21	4.75	3.125	2	1.625	1205.3
Week 22	5.625	6.375	3.75	5	1244.1
Week 23	3.625	4.375	2.125	3.375	1244.8
Week 24	4	2	2.125	1.5	1244.3
Week 25	7.375	6.375	4.625	5.875	1169.6
Week 26	8	5.5	4	3.5	1185
Week 27	11.75	8.25	7.875	6.875	1195.5
Week 28	5.125	4.75	3	3.875	1218
Week 29	4.875	5	3	4.125	1194.7
Week 30	4.875	5.25	2.75	4	1207.2
Week 31	5.375	6.25	4.625	5.25	1179.7
Week 32	3.625	2.875	2.25	1.875	1200.2
Week 33	4.125	3.5	3.25	2.375	1208.4
Week 34	0.625	0.875	0.625	0.5	1264.7
Week 35	6	6.125	4.75	4.625	1300.7
Week 36	2.5	2.375	2.125	1.75	1254.6
Week 37	10.75	5.125	6.375	3.625	1240.8
Week 38	8.25	4.75	5.25	3.875	1226.5
Week 39	9.125	7.375	5.25	5.625	1200.3
Week 40	11.25	12	8.375	10	1207.7
Week 41	3.75	3.625	3.25	3	1166.4

5. Result and Discussions

Training Using Backpropagation

Neural Network in this method is trained on the basis of features extracted from the Sentiment Classifier. We used Levenberg Marquardt as the training function in neural network as this is the most widely used optimization algorithm. The advantage of this function is that this function converges very fast. We have compared the dependency of future stock prices on previous one, two and three weeks' sentiments and as a result stock prices were more dependent on previous two weeks' sentiments. So we have used previous two weeks' sentiments as input to the Neural Network. In this training total ten inputs were given to the neural network, eight inputs were the features extracted from the Sentiment Classifier and the other two were the price of the stock in past two weeks.

Number of Inputs - Number of inputs are 10. As the input contains data of previous 2 weeks and each week's data contains 5 attributes i.e. Pos Score, Neg Score, Pos Tre Score, Neg Tre Score and the value of the stock for that week.

Number of Hidden layers - Number of hidden layer is only 2 with different number of neurons for each commodity.

Number of outputs - Number of outputs are only 1 i.e. the value of the stock to be predicted.

Results using Backpropagation

To validate our results, we have used collection of sentiments and stock values of the duration April 2014 to May 2015.

After training the neural network we have predicted the price of Gold, Silver and Crude Oil.

Gold

Neural network used is with 2 hidden layers and each layer contain 12 neurons.

Table 4: Table showing actual and predicted price of gold

Actual	Predicted	Absolute Error
1183.1	1199	15.9
1203.1	1267.81	64.71
1201.5	1179.05	22.45
1186.9	1175.38	11.52

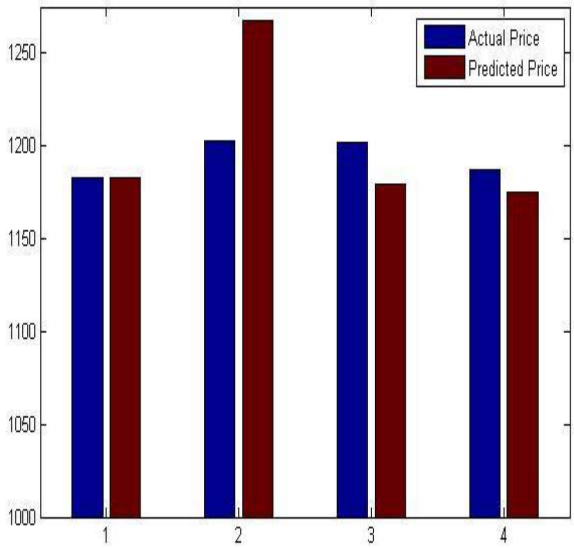

Figure 3: Graph between actual and predicted prices of gold using data of last 2 weeks

Silver

Table 5: Neural network used is with 2 hidden layers and each layer contains 12 neurons.

Actual	Predicted	Absolute Error
16.67	17.23	0.56
16.44	16.33	0.11
17.54	16.94	0.6
17.03	17.19	0.16
16.68	17.14	0.46

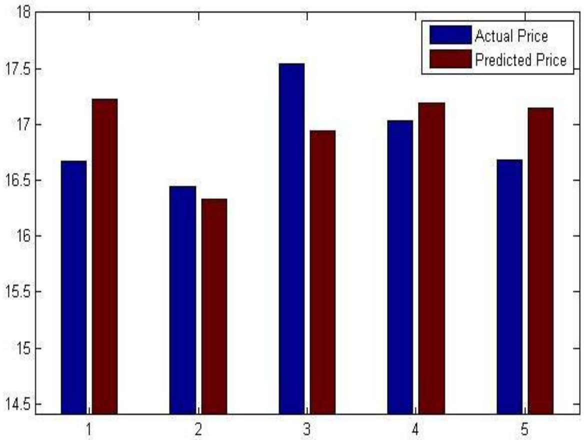

Figure 4: Graph between actual and predicted prices of silver using data of last 2 weeks

Crude Oil

Table 6: Neural network used is with 2 hidden layers and each layer contain 15 neurons

Actual	Predicted	Absolute Error
59.96	62.48	2.52
59.61	60.96	1.35
59.63	61.74	2.11
56.93	55.26	1.67

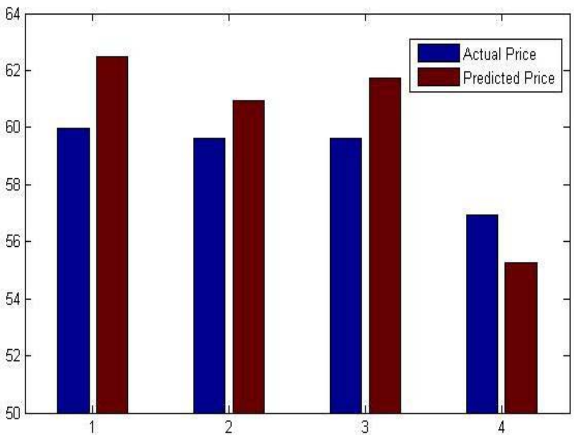

Figure 5: Graph between actual and predicted prices of crude oil using data of last 2 weeks

The accuracy of the trained neural network is gauged on the basis of Mean. Absolute Error and Mean Absolute Percentage Error(MAPE). The results pertaining to these errors are shown table 4 A mean absolute error of 28.64, 0.37 and 1.91 and mean absolute percentage error of 2.38, 2.22 and 3.23 is observed for Gold, Silver and Crude oil respectively which is quite good.

Table 7: Mean Absolute Error and MAPE of Gold, Silver and Crude oil

Commodity	Mean Absolute Error	MAPE
Gold	28.64	2.38
Silver	0.37	2.22
Crude Oil	1.91	3.23

Training Using Artificial Bee Colony (ABC) Algorithm

ABC is an optimization algorithm which is based on the intelligent foraging behavior of the honey bee swarm. It algorithm discovered by Kharaboga is commonly used robust algorithm. It is very simple, robust and population based stochastic optimization algorithm. So instead of training the neural networks using the basic back propagation we have used the help of an evolutionary algorithm which is good in its exploration and exploitation capabilities i.e. Artificial Bee Colony (ABC) algorithm. The generic back propagation technique may have some drawbacks such as initialization of weights, parameters. Also it is possible that sometimes saturation is reached if the derivative is too small to make any significant changes in the weights causing the network to settle in an incorrect local minimum. To overcome the disadvantage of gradient descent based algorithms we have used ABC algorithm to evolve weights. ABC can be used to find the optimal set of weights in training neural networks and can prove better than the gradient descent algorithm. We have used the algorithm in a similar manner to optimize the weights in neural networks. Again in this neural networks also we have used the dependency of sentiments of last weeks to predict the future stock prices. The trained neural network adheres to the following specifications

Number of Inputs - Number of inputs are 10. As the input contains data of previous 2 weeks and each week's data contains 5 attributes i.e. Pos Score, Neg Score, Pos Tre Score, Neg Tre Score and the value of the stock for that week.

Number of Hidden layers- Number of hidden layer is only 1 with 5 neurons excluding the bias one in it.

Number of outputs - Number of outputs are only 1 i.e. the value of the stock to be predicted.

Results using ABC

Gold

Table 8: Table showing actual and predicted price of gold using ABC

Actual	Predicted	Absolute Error
1183.1	1185.59	2.49
1203.1	1208.48	5.38
1201.5	1193.6	7.9
1186.9	1209.63	22.63

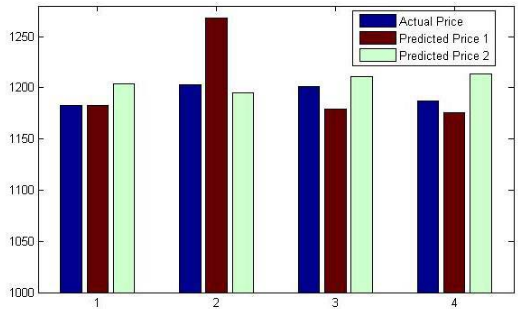

Figure No. 6: Graph between actual and predicted prices of gold where predicted price 1 represents price of crude oil using simple neural network and predicted price 2 represents price of crude oil using ABC algorithm in neural network

Silver

Table 9: Table showing actual and predicted price of silver

Actual	Predicted	Absolute Error
16.67	16.84	0.17
16.44	16.93	0.49
17.54	17.12	0.41
17.03	16.42	0.6
16.68	17.55	0.87

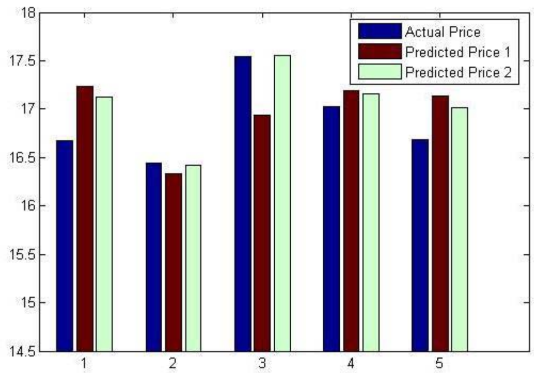

Figure 7: Graph between actual and predicted prices of silver where predicted

price 1 represents price of crude oil using simple neural network and predicted price 2 represents price of crude oil using ABC algorithm in neural network

Crude Oil

Table 10: Table showing actual and predicted price of Crude Oil

Actual	Predicted	Absolute Error
59.96	58.12	1.83
59.61	58.47	1.13
59.63	55.82	3.8
56.93	55.64	1.28

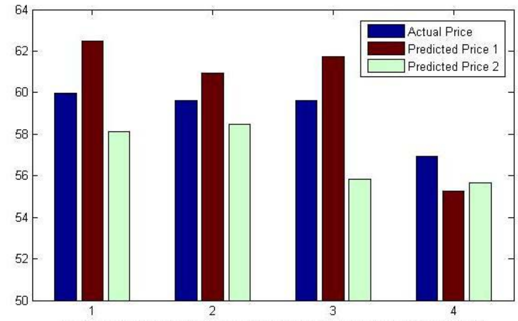

Figure 8: Graph between actual and predicted prices of crude oil where predicted

price 1 represents price of crude oil using simple neural network and predicted price 2 represents price of crude oil using ABC algorithm in neural network

Table 11: Mean Absolute Error and MAPE of Gold, Silver and Crude oil

Commodity	Mean Absolute Error	MAPE
Gold	9.6	0.8
Silver	0.5	3.01
Crude Oil	2.01	3.38

The accuracy of the trained neural network using ABC algorithm is gauged on the basis of Mean Absolute Error and Mean Absolute Percentage Error (MAPE). The results pertaining to these errors are shown table A mean absolute error of 9.6,0.50 and 2.01 and mean absolute percentage error (MAPE) of 0.8 , 3.01 and 3.38 is observed for Gold, Silver and Crude oil respectively which is good as compared to the training using simple neural network.

6. Conclusions

In our research work we got a positive code which significantly reduce the error percentage. This code was able to predict the market value of gold, silver and crude oil approximately to the day's value in the stock exchange. However by increasing sample size the scope of narrowing down the predicted value and the market value is still there.The average price was predicted not the opening price of each day. If it would have been at professional level it would have been able to the same

7. Future Scope

Our work was on gold, silver and crude oil. The above code can be extended to various other commodities like automobiles pharmaceutical and various industrial product

References

[1] Bollen, J., Mao, H., and Zeng, X. Twitter mood predicts the stock market. Journal of Computational Science 2, 1 (2011)
[2] Ray Chen, Marius Lazer. Sentiment Analysis of Twitter Feeds for the Prediction of Stock Market Movement(2011)
[3] Kouloumpis, E., Wilson, T., and Moore, J. Twitter sentiment analysis: The good the bad and the omg!

ICWSM 11 (2011), 538-541.
[4] Medhat, W., Hassan, A., and Korashy, H. Sentiment analysis algorithms and applications: A survey. Ain Shams Engineering Journal 5, 4 (2014), 1093-1113.
[5] Karaboga, Dervis, and BahriyeBasturk. "A powerful and e_cient algorithm for numerical function optimization: arti_cial bee colony (ABC) algorithm." Journal of global optimization 39.3 (2007): 459-471.
[6] http://www.investing.com/commodities
[7] https://www.python.org/download/releases/2.6
[8] SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining
[9] http://www.nltk.org
[10] http://www.silver-phoenix500.com/

Author Profile

Akshay Kumar is presently working as a Software Engineer in Comviva Technology. He pursued Under Graduate Degree from Department of Computer Science and Engineering of Motilal Nehru National Institute of Technology, Allahabad.