www.ijecs.in
International Journal Of Engineering And Computer Science
Volume 8 Issue 9 September 2019, Page No. 24847-24850
ISSN: 2319-7242 DOI: 10.18535/ijecs/v8i09.4361

Stock Market Prediction using Machine Learning and Cloud Computing

Nirbhay Narkhede

Lakshmi Narain College of Technology Bhopal(M.P),India

Abstract

In the world with increasing globalization, where money places a crucial role in determining the expansion and earnings of a company trading places a very crucial role. Multiple companies invest millions and billions of dollars in other countries with an expectation to make profits. In such a risky business Predicting the movement of the market can help companies or individual in making good decisions and can prevent severe loses. In this research paper we will discuss how we can use the computational power of the computer on cloud along with the machine learning algorithms to predict the closing values of the stocks which is a big challenge otherwise. For this purpose, we will use Python as our programming language which supports a lot of ML based Libraries. The models we will be using are SVM (Support Vector Machine), Linear Regression, Random Forest, XG Boost, LSTM for deep learning

Key Words: Stock Market, Machine Learning, Predictions, Cloud Computing.

1. Introduction

The system of stock trading is fairly simple, someone with excessive cash decides to invest it in the market by buying shares of different companies based on their market performance and market capital with a hope to make good profits in future. This is done by buying the share at lower rate and selling it at a high value. The prediction of 'when' the stock will hit its low and high can make a huge difference in the earnings.

There are already a lot of indicators in the market which somehow helps in market prediction but their accuracy is low, also it requires a lot of knowledge from the user end to analyse these indicators. These indicators also use the past data of the markets like high, lows, volume, P/E ratio etc, to give meaningful insights to the users.

In this research paper we will be using some calculators already in the market like gann calculator, tenkan sen for comparing the accuracy of our model.

A stock prediction largely depends on its previous data and most importantly the last day data. Using the data scrapping the whole data of a company can be extracted from the stock exchange. In order to predict the new price of the day we basically needs 3 parameters i.e. open,high,low to predict the closing price. The closing price will be the value our model will predict based on its past data and the Final result will be in the form of whether to buy or sell the stock based on its opening price. The models we are using focuses mostly on intra day trading. This project will be using a lot of data which may take days to process and hence it will not be efficient to run it on our regular PC. That's where cloud computing comes into play. The advance high computing GPU's can run our model in minutes and can get us the results as soon as possible. There are two types of Data we can use to train our model

1) Data from stock exchange
2) Tick by Tick data

It should be kept in kind how much of this data has to be used to prevent overfitting or over training of the models. The tick by tick data can be extracted using API's Provided by your stock broker.

2. Methedology

This basic rule of this project is to use the data from previous day to predict the closing value of the day. Though there are many other factors which effects the price of a stock which are not in our hands like rumours, government policies etc. But working with numbers we should be able to make best model for prediction.

The Training data should be first normalised and database we will be using is mongo DB for its fast and efficient processing.

The data we have will be in the form of 4 columns O1,H1,L1,C1 i.e. open(for day 1),high(for day 1),Low(for day 1),close(for day 1).The last column is our label we need to predict using our models. Therefore, to predict the value for tomorrow we should be having the data till today.

While preparing the data set for training we have to shift the fourth column one block up. So that the data of today has a label of closing value of tomorrow i.e. O1,H1,L1,C2.

We are here using multiple models to predict their accuracy and based on this we can define which to use for the predictions.

3.Models Used

1) SVM (Support Vector Machine)

Support Vector Machine(SVM) is a supervised machine learning algorithm which is widely used for both classification and regression problems. However, it is mostly used in classification problems. But in our case, we will be using it for regression purpose because we need a numerical value as our output rather than a category.

It uses a plane on a 2D surface along with a plane to divide the plane in two parts where each class resides on the either sides.

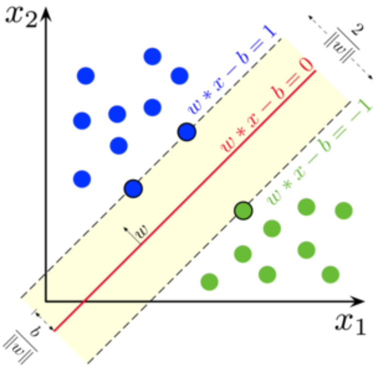


Fig 1. Support Vector Machine

2)Linear Regression

Basically, the most widely used and simple regression model is Linear Regression. The representation is a linear equation that use or combines a particular set of input values i.e. x the solution will be the predicted output for that set of input values i.e. (y). Both the input values (x) and the output value(y) are numeric (i.e. the closing value of the stock).

The linear equation in this model assigns one scaling factor to each value of x, called a coefficient which is represented by the capital Greek letter Beta (B). One additional coefficient called intercept or the bias coefficient is also added which gives the additional degree of freedom i.e. Moving up and down on a 2D plane.

For example, in a simple regression problem (a single x and a single y), the form of the model would be:

y = B0 + B1*x

But in our case, we have multiple values of x (Open, high, low) and a single value for y (close value). So, we can use either Gradient Descent or Ordinary least Squares which supports multiple inputs and provides a single output.

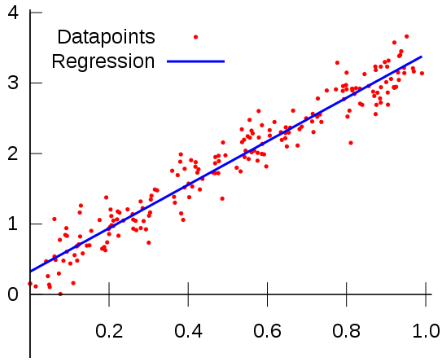


_Fig 2. Linear Regression

3)Random Forests

Another supervised machine learning model is Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by using multiple decision trees at the training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression in our case) of the individual trees. It is an extension of decision tree and keeps track of overfitting of training data.

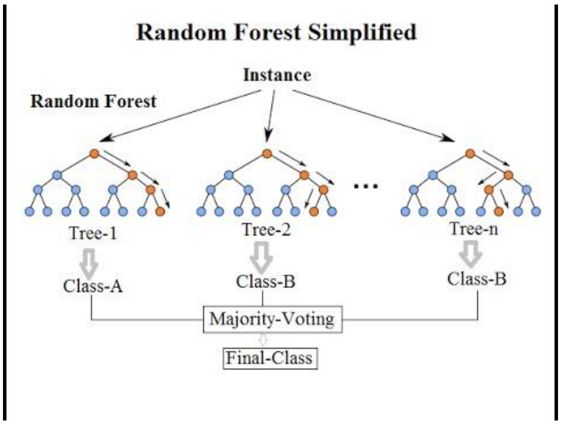


Fig 3. Random Forest

4)XG Boost

XG Boost is an implementation of gradient boosted decision trees designed for speed and performance.

It has a perfect combination and balance of software and hardware optimization techniques to provide superior/best results using fewer computing resources in the shortest amount of time.

The two reasons to use XG Boost are also the two goals of the project:

1. Execution Speed.
2. Model Performance.

1. XG Boost Execution Speed

Generally, XG Boost is fast. Really fast when compared to other implementations of gradient boosting.

Szilard Pafka performed some objective benchmarks comparing the performance of XG Boost to other implementations of gradient boosting and bagged decision trees.

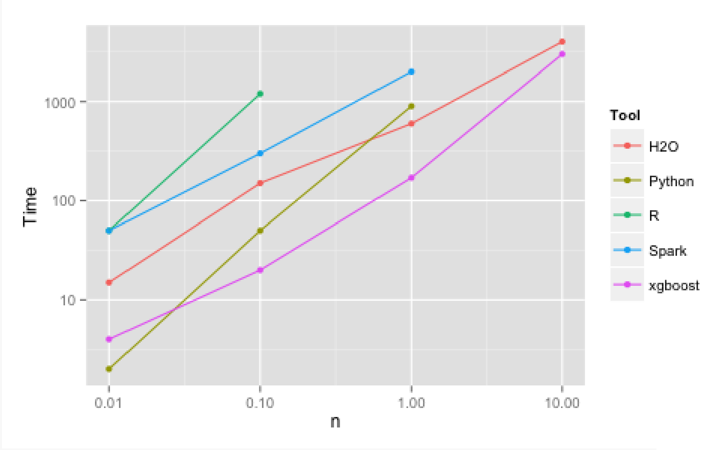


Fig 4. Benchmark Performance of XG Boost

His results showed that XG Boost was almost always faster than the other type of implementation models from R, Spark, Python etc. From this experiment we can conclude that this should be the best choice for machine learning model for our project.

2. XG Boost Model Performance

XG Boost dominates structured or tabular datasets on classification and regression predictive modelling problems.

It has been proved that this model is far better than its other counter parts when it comes to speed and accuracy. When the founder of this model won a ML Competition.

4)LSTM deep learning neural network

Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Till now we have used only regression models which are fast but compromises on the accuracy, the deep learning models provides high accuracy but also takes a lot of time and computational resources. The deep learning algorithms uses multiple players with multi neuron architecture which is what makes it highly computational and time consuming.

The deep learning also has many types of networks and LSTM is one of the feedback neural networks where results backtracks to previous layers unlike feed forward networks. It's hard to implement and use, as compared to single node regressions.

A common LSTM unit is composed of a cell, an input gate, an output gate and a forget gate. The cell is responsible for remembering the values over a particular time interval and the three *gates* regulate the flow of information into and out of the cell.

This networks are well-suited for classifying, processing and making predictions based on time series data just like we have in our project. LSTMs is a extended version of a traditional Recurrent Neural Network developed to deal with the exploding and vanishing gradient problems.

This model is specifically used when we want our model to deal with long-term dependencies like in our case the closing value of our stock depends on its all previous values. LSTM is smart model which determine how long to hold onto old information, when to remember and forget, and how to make connections between old memory with the new input.

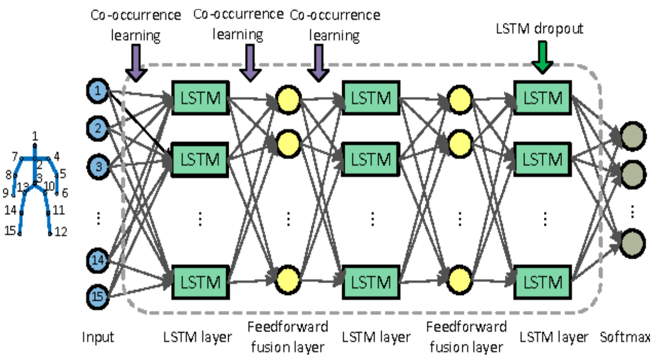


Fig 5 LSTM Network

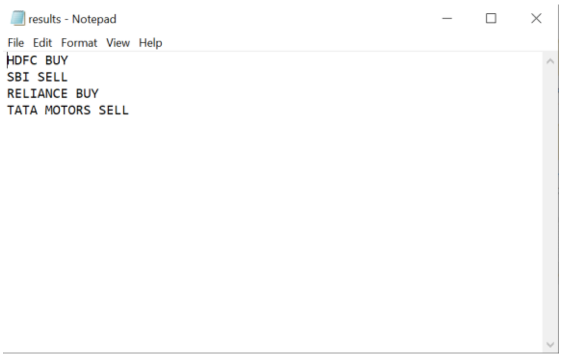
The best way to implement LSTM in our project is to use TensorFlow by google, this library does most of the work for us but still designing this network specific to our project is a big and time-consuming task.

Results

Open	High	Low	Close
2076.9	2077.95	2049	2051.800049
2053.8	2060.2	2035.55	2045.849976
2049	2059	2032.6	2052.199951
2051	2060	2040.75	2056.949951
2055	2055	2025.1	2041.150024
2046.3	2048	1994.15	2000.400024
2001.9	2021.35	1989.55	2011.849976
2019.85	2037.4	2018.95	2029.599976
2021.3	2022.95	1988	1992.199951
1994.9	2009.85	1980.95	1989.199951
1994.85	1998	1956.5	1961.349976
1973.9	1999.85	1910.2	1970.25
1967.25	1972.2	1915	1925.699951
1925.7	1962	1915	1952.400024
1962.95	1978	1955.3	1968.199951
1969.4	1986	1955.15	(predicted value)

Fig 6. Result Output

The final result will be in a text file with the company name along with whether to buy or sell the stock for the given day.



```
HDFC BUY
SBI SELL
RELIANCE BUY
TATA MOTORS SELL
```

Fig 7. Result in a Text file

Conclusion

The expected scrapped data size should be more than 300GB including historical and tick by tick data. The type of models we are using are expected to take a lot of time to process and predict the new value and hence a powerful GPU on cloud has to be used in order to boost our processing time. Using the predicted value, it will be easy to calculate whether to buy or sell the stock for the given day.

References

[1] Analytic Vidhya” [Online] Available: https://www.analyticsvidhya.com/

[2] Machine Learning Mastery” [Online] Available:https://machinelearningmastery.com/linear-regression-for-machine-learning/

[3] Medium” [Online] Available: https://medium.com/simple-ai/linear-regression-intro-to-machine-learning-6-6e320dbdaf06

[4] Ho TK (1998). "The Random Subspace Method for Constructing Decision Forests" (PDF). IEEE Transactions on Pattern Analysis and Machine Intelligence. 20 (8): 832–844. doi:10.1109/34.709601

[5] Long-Short-Term-Memory (LSTM) https://en.m.wikipedia.org/wiki/Long_short-term_memory

[6] *Graves, Alex; Fernández, Santiago; Gomez, Faustino (2006). "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks"*

[7] Semantic Scholar https://www.semanticscholar.org/paper/Co-occurrence-Feature-Learning-for-Skeleton-based-Zhu-Lan/4eb753322f13443d0e463e6c7123088734b3583a/figure/0