Stock Market Analysis using Supervised Machine Learning

Kunal Pahwa
Dept. of Computer Science & Engineering
Amity University, Noida
Uttar Pradesh, India
kunalpahwa2708@gmail.com

Neha Agarwal
Dept. of Computer Science & Engineering
Amity University, Noida
Uttar Pradesh, India
nagarwal2@amity.edu

Abstract— **Stock market or Share market is one of the most complicated and sophisticated way to do business. Small ownerships, brokerage corporations, banking sector, all depend on this very body to make revenue and divide risks; a very complicated model. However, this paper proposes to use machine learning algorithm to predict the future stock price for exchange by using open source libraries and preexisting algorithms to help make this unpredictable format of business a little more predictable. We shall see how this simple implementation will bring acceptable results. The outcome is completely based on numbers and assumes a lot of axioms that may or may not follow in the real world so as the time of prediction.**

Keywords—Basics, Data Analysis, Fundamental, Implementation, Linear Regression, Stock Market, Supervised Machine Learning

I. INTRODUCTION

STOCK MARKET is one of the oldest methods where a normal person would trade stocks, make investments and earn some money out of companies that sell a part of themselves on this platform. This system proves to be a potential investment schemes if done wisely. However, the prices and the liquidity of this platform is highly unpredictable and this is where we bring technology to help us out. Machine learning is one such tool that helps us achieve what we want. The following 3 paragraphs will briefly explain the key components of this paper:

Stock market as we know, is a very important trading platform which affect everyone at an individual and national level [2]. The basic principle is quite simple, Companies will list their shares in the companies as small commodities called Stocks. They do so in order to raise money for the firm. A company lists its stock at a price called the IPO or initial public offering. This is the offer price at which the company sells the stock and raises money. After which these stock are the property of the owner and he may sell them at any price to a buyer at an Exchange such as BSE or Bombay Stock Exchange. Traders and buyers continue selling these shares at their own price but the company only gets to keep the money made during the IPO. The continue hoping of hares from one party to another in order to make more profits, results in an increase of price of the particular share after every profitable transaction. However, if the company issues more stocks at a lower IPO, then the market price for exchange goes down and traders suffer a loss. This exact phenomenon is the reason for the fear people have in investing in stock markets and the reason for the fall and rise of stock prices in a nutshell.

Now if we try to graph the stock exchange price over the time period (say 6 months), is it really hard to predict the next outcome on the graph?
A human brain is very capable of extending the graph a few coordinates by just simple looking at it for a few minutes. [1]And if we crowd compute i.e. make a group of random people try to extend the graph by a fixed amount of time (say a week), we will get a very reasonable and approximate answer to a real life graph.
Because many brains will try to interpret the pattern and make a guess and this such activity has proven to be a lot more successful in practice than it seems in theory. [5]
Having said that, predicting the true value of the stock is best estimated by the method of crowd computing.
But as it very much evitable that crowd computing is a very slow activity therefore we try to use a computer here to simulate such example with a more scientific and mathematical approach.

In statistics, there is a way where we look at the values and attributes of a problem in a graphs and identify the dependents and independent variables and try to establish or identify an existing relationship amongst them [3][4]. This technique is known as linear regression in statistics and is very commonly used due to its very simple and effective approach. In machine learning we have adapted the same algorithm where we use the features to train the classifier which then predicts the value of the label with certain accuracy which can be checked while training and testing of the classifier. For a classifier to be accurate you must select the right features and have enough data to train your classifier. The accuracy of your classifier is directly proportional to the amount of data provided to the classifier and the attributes selected.

978-1-7281-0211-5/19/$31.00 2019 ©**IEEE**

So with the basic knowledge of stock market, graphs and data analysis coupled with machine learning; we are now prepared to device the program.

II. Prediction Model

A. *Data Analysis Stage*

In this stage, we shall look at the raw data available to us and study it in-order to identify suitable attributes for the prediction of our selected label.

Now the data that we're going to use for our program is taken from www.quandl.com, a premier dataset providing platform.

The dataset taken is for GOOGL by WIKI and can be extracted from quandl using the token "WIKI/GOOGL". We have extracted and used approximately 14 years of data.

The attributes of the dataset include:

Open (Opening price of Stock)
High (Highest price possible at an instance of time)
Low (Lowest price possible at an instance of time)
Close (Closing price of stock)
Volume (Total times traded during a day)
Split ratio

$$\left.\begin{array}{l}\text{Adj. Open}\\ \text{Adj. High}\\ \text{Adj. Low}\\ \text{Adj. Close}\\ \text{Adj. Volume}\end{array}\right\}\text{Adjusted values of above attributes}$$

We select the attribute "*Close*" to be our label (The variable which we shall be predicting) and use "Adj. Open, Adj. High, Adj. Close, Adj. Low and Adj. Volume" to extract the features that will help us predict the outcome better.

It is to be noted that we use adjusted values over raw as these values are already processed and free from common data gathering errors.

Now we aware that the graphs made for stock analysis use the above attributes to plot them. Such graphs are called OHLCV graphs [11] and are very informative about the status of the stocks. Now we use the same graphing parameters to decide out features for the classifier.

Let's define the set of features which we shall be using:

- *Adj. Close*: This is an important source of information as this decides market opening price for the next day and volume expectancy for the day.
- *HL_PCT*: This is a derived feature which is defined by:

$$HL\_PCT = \frac{\text{Adj. High} - \text{Adj. Low}}{Adj.Close} \times 100$$

We use percentage change as this helps us reduce the number of features but retain the net information involved. High-Low is a relevant feature because this helps us formulate the shape of the OHLCV graph.

- *PCT_change*: This is also a derived feature, defined by:

$$PCT\_Change = \frac{Adj.Close - Adj.Open}{Adj.Open} \times 100$$

We do the same treatment with Open and Close as High and Low, since they both are very relevant in our prediction model and helps us reduce number of redundant features as well.

- *Adj. Volume*: This is a very important decision parameter as the volume traded has the most direct impact on future stock price than any other feature. Therefore we shall use this as it is in our case.

We have successfully analyzed the data and extracted the useful information that we shall be needing for the classifier. This is a very crucial step and shall be treated with extreme care. A miss of information or small error in deriving useful information will lead to a fail prediction model and a very inefficient classifier.

Also, the features extracted are very specific to the subject used and will definitely vary from subject to subject. Generalization is possible if and only if, the data of the other subject is collected with the same coherence as the earlier subject.

B. *Training and Testing Stage*

At this stage we shall be using what we extracted from our data and implement in our machine learning model.

We will be using SciPy, Scikit-learn and Matplolib libraries in python to program our model, train them with the features and label which we extracted and then test them with the same data.

First we shall preprocess the data to make the data which includes:

- Shifted values of the label attribute by the percentage you want to predict.
- Dataframe format is converted to Numpy array format.
- All NaN data values removed before feeding it to the classifier.
- The data is scaled such that for any value X, $X \in [-1,1]$
- The data is split into test data and train data respective to its type i.e. label and feature.

Now the data is ready for us to input in a classifier. We will be using the simplest classifier i.e. Linear Regression, which is defined in Sklearn library of the Scikit-learn package. We choose this classifier because of its simplicity and because it serves our purpose just right. Linear regression is a very commonly used technique for data analysis and forecasting. It essentially uses the key features to predict relations between variables based on their dependencies on other features. [9] This form of prediction is known as Supervised Machine learning.

Supervised learning is a method where we input labelled data i.e. the features are paired with their labels. Here we train the classifier such that it learns the patterns of which combination of features result in which label.

Here in our case, the classifier sees the features and simply looks at its label and remembers it. It remembers the combination of features and its respective label which in our case is the stock price a few days later. Then it moves on and learns what pattern is being followed by the features to produce their respective label. This is how supervised machine learning works [10].

For testing in supervised machine learning, we input some combination of features into the trained classifier and cross check the output of the classifier with the actual label. This helps us determine the accuracy of our classifier. Which is very crucial for our model. A classifier with an accuracy less than 95% is practically useless.

Accuracy is a very crucial factor in a machine learning model. You must understand what accuracy means and how to increase your accuracy on the next subtopic.

C. *Results*

Once the model is ready, we use the model to obtain the desired results in any form we want. In our case, we shall be plotting a graph of our results (fig. 1) as per our requirements which we have discussed earlier in this paper.

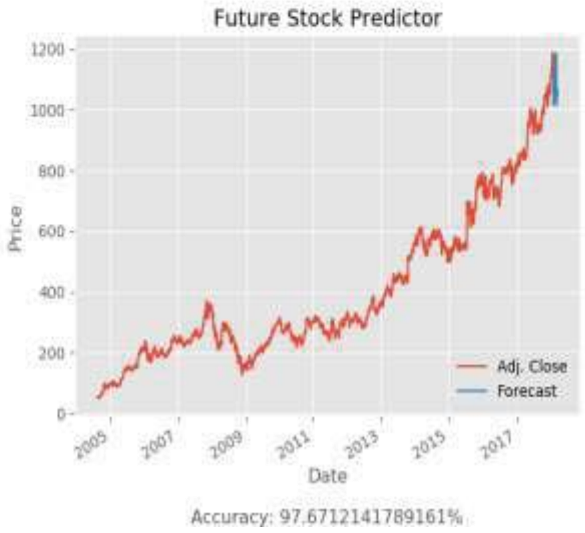

Fig. 1. Graph showing stock price of GOOGL from year 2005 till July 2018. Red is the line representing given data and blue is representing the forecasted or the predicted value of stock.

The key component of every result is the accuracy it delivers. It should be according to our needs and as stated earlier, a model with accuracy less than 95% is practically useless. There are some standard methods to calculate accuracy in machine learning, some are as follows:

- R^2 value of the model.
- Adjusted R^2 value
- RMSE Value
- Confusion matrix for classification problems.
- And many more

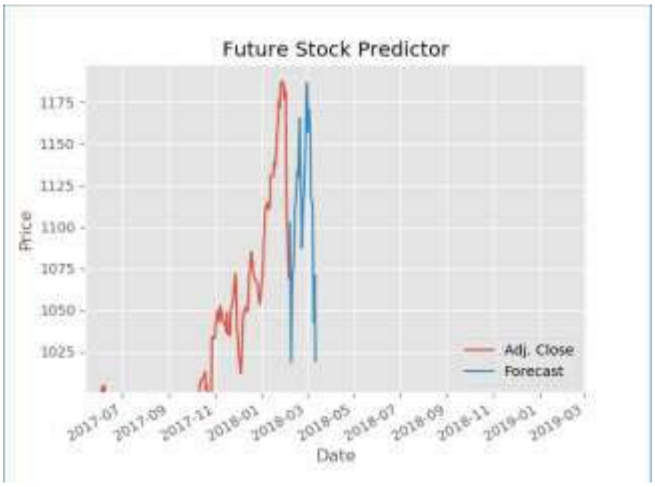

Fig. 2. Graph showing the exact amounts of predicted values.

Accuracy is the component which every machine learning developer is always committed to contribute towards. After the model is developed, there is infinite effort towards optimizing the model to get more and more accurate results. There are some very common and simple ways to boost the efficiency of your model, and have been discussed above.

However let us look at some of the standard ways to optimize a machine learning algorithm:

- Unconstrained Optimization
  - Gradient Decent
  - Newton's Method
  - Batch Learning
  - Stochastic Gradient Decent
- Constrained Optimization
  - Lagrange Duality
  - SVM in primal and Dual forms
  - Constrained Methods

Most of the machine learning problems are, in the end, optimization problems, where we minimize a function subject to some constraints.

III. Helpful Hints

A. *Requirements and Specification*

You must know the exact problem requirements and the machine and throughput specification thoroughly as the very first stage. Do not rush this step as this step is very crucial in deciding the overall plan for the development of the program.

Study case carefully, do a little background check, collect ample of knowledge of the subject in hand, and identify what you actually want and set it as your goal.

B. *Careful function Analysis*

You must be very careful while deriving the features from the data as they play a direct role in the prediction model. They all must make direct sense in conjunction with the labels. Minimizing the functions subject to the requirement constraints, as much as possible is highly recommended as well.

C. Implementation

You must select the appropriate model in which you will implement your math to produce results.

The model selected or designed must be in conjunction with the input data. A wrong model designed or selected for an inappropriate data or vice-versa, will result in a garbage model which is completely useless. You must see for compatible SVM or some other available methods to process your data. Trying out different models simultaneously to check which works the most effectively is also a good practice.

Furthermore, implementation is the simplest step of them all and should take the least amount of time so as to save us some time from the total time cost which could be utilized in some other important steps.

D. Training & Testing

Training of a model is very straightforward. You only need to make sure that the data is consistent, coherent and is available in great abundance. A large set of training data contributes to a stronger and more accurate classifier which ultimately increases the overall accuracy.

Testing is also a very straightforward process. Make sure your test data is at least 20% of the size of your training data. It is important to understand that testing is the test of you classifiers accuracy and is sometimes observed to be inversely proportional to a classifiers score. However, the accuracy of the classifier has no dependency or correlation with testing. It sometimes seem so, but testing does not have any relationship with the classifier.

E. Optimization

It is almost impossible to create a versatile classifier in a single go, therefore we must always continue to optimize. There is always some room for improvements. When optimizing, keep in mind the standard methods and basic requirements.

Shifting to SVM, trying and testing different models, looking for new and improved features, changing the entire data model to suit the model entirely etc. are some very fundamental ways to optimize your classifier.

IV. Some Common Mistakes

Let us mention some of the common mistakes made by practitioners in this field, which you are required to avoid [12]:

- Bad annotation of training and testing datasets
- Poor understanding of algorithms' assumptions
- Poor understanding of algorithms' parameters
- Failure to understand objective
- Not understanding the data
- Avoid leakage (Features, information)
- Not enough data to train the classifier
- Using machine learning where it is not necessary

V. ConclusionS

Machine learning as we have seen till now, is a very powerful tool and as evitable, it has some great application. We have seen till now that machine learning is very much dependent upon data. Thus it is important understand that data is quite invaluable and as simple is it may sound, data analysis is not an easy task.

Machine learning have found tremendous application and has evolved further into deep learning and neural networks, but the core idea is more or less the same for all of them.

This paper delivers a smooth insight of how to implement machine learning. There are various ways, methods and techniques available to handle and solve various problems, in different situations imaginable. This paper is limited to only supervised machine learning, and tries to explain only the fundamentals of this complex process.

Acknowledgment

This paper is written based upon the summer project of the author during his degree programme in the year 2018 after the invaluable guidance by Neha Agarwal (Asst. Prof.) AUUP, Noida.

The author of this paper, does not claim rights to any of the algorithm, code, data, formula used, definitions, problem solving approach, as his property. He has only used his brain in compiling it all together and made efforts in obtaining results and putting it together in the format of an IEEE paper.

References

[1] Andrew McCallum, Kamal Nigam, Jason Rennie, Kristie Seymore "A Machine learning approach to Building domain-specific Search engine", IJCAI, 1999 - Citeseer

[2] Yadav, Sameer. (2017). STOCK MARKET VOLATILITY - A STUDY OF INDIAN STOCK MARKET. Global Journal for Research Analysis. 6. 629-632.

[3] Montgomery, D.C., Peck, E.A. and Vining, G.G., 2012. Introduction to linear regression analysis (Vol. 821). John Wiley & Sons.

[4] Draper, N.R.; Smith, H. (1998). Applied Regression Analysis (3rd ed.). John Wiley. ISBN 0-471-17082-8.

[5] Robert S. Pindyck and Daniel L. Rubinfeld (1998, 4h ed.). Econometric Models and Economic Forecasts

[6] "Linear Regression", 1997-1998, Yale University http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

[7] Agarwal (July 14, 2017). "Introduction to the Stock Market". Intelligent Economist. Retrieved December 18, 2017.

[8] Jason Brownlee, March 2016, "Linear Regression for machine learning", Machine learning mastery, viewed on December 2018, https://machinelearningmastery.com/linear-regression-for-machine-learning/

[9] Google Developers, Oct 2018, "Decending into ML: Linear Regression", Google LLC, https://developers.google.com/machine-learning/crash-course/descending-into-ml/linear-regression

[10] Fiess, N.M. and MacDonald, R., 2002. Towards the fundamentals of technical analysis: analysing the information content of High, Low and Close prices. Economic Modelling, 19(3), pp.353-374.

[11] Hurwitz, E. and Marwala, T., 2012. Common mistakes when applying computational intelligence and machine learning to stock market modelling. arXiv preprint arXiv:1208.4429.