IJFANS INTERNATIONAL JOURNAL OF FOOD AND NUTRITIONAL SCIENCES
ISSN PRINT 2319 1775 Online 2320 7876
 UGC CARE Listed (Group -I) Journal Volume 11, Iss 8 , 2022

Estimating Stock Market Prices Using a support vector machine and random forest

Dr. Jitin Kumar Gambhir
HOD (Management), Institute of Management Studies, Noida

Abstract

In the past decades, there is an increasing interest in predicting markets among economists, policymakers, academics and market makers. The objective of the proposed work is to study and improve the supervised learning algorithms to predict the stock price. Stock Market Analysis of stocks using data mining will be useful for new investors to invest in stock market based on the various factors considered by the software. Stock market includes daily activities like Sensex calculation, exchange of shares. The exchange provides an efficient and transparent market for trading in equity, debt instruments and derivatives. Our aim is to create software that analyses previous stock data of certain companies, with help of certain parameters that affect stock value. We are going to implement these values in data mining algorithms and we will be able to decide which algorithm gives the best result. This will also help us to determine the values that particular stock will have in near future. We will determine the patterns in data with help of machine learning algorithms.

Introduction

Machine learning is a subfield of computer science, but is often also referred to as predictive analytics, or predictive modeling. Its goal and usage is to build new and/or leverage existing algorithms to learn from data, in order to build generalizable models that give accurate predictions, or to find patterns, particularly with new and unseen similar data.

magine a dataset as a table, where the rows are each observation (aka measurement, data point, etc), and the columns for each observation represent the features of that observation and their values. At the outset of a machine learning proposed system, a dataset is usually split into two or three subsets. The minimum subsets are the training and test datasets, and often an optional third validation dataset is

created as well. Once these data subsets are created from the primary dataset, a predictive model or classifier is trained using the training data, and then the model's predictive accuracy is determined using the test data.

As mentioned, machine learning leverages algorithms to automatically model and find patterns in data, usually with the goal of predicting some target output or response. These algorithms are heavily based on statistics and mathematical optimization. Optimization is the process of finding the smallest or largest value (minima or maxima) of a function, often referred to as a loss, or cost function in the minimization case. One of the most popular optimization algorithms used in machine learning is called gradient descent, and another is known as the normal equation. In a nutshell, machine learning is all about automatically learning a highly accurate predictive or classifier model, or finding unknown patterns in data, by leveraging learning algorithms and optimization techniques.

Literature Survey

Investing money into unpredictable, unstable, and uncontrollable facets can be extremely risky. Like the lottery, the success of stock market trading is partly attributed to luck. Many people have lost vast amounts of money through poor investment decisions that they've made.

Recently, investors with shares in loan-giving companies and American car manufacturers, which were previously a fairly stable investment, have suffered severe losses due to the economic crisis. Investors must understand and accept this risk as an intrinsic part of investing.

There are, however, attractive benefits to successful financial investments. With intelligent decisions, investing can yield significant capital gains, stability, and security. By analyzing the trends of the stock market, the companies one is invested in, and by following an investment strategy, one can be successful in the stock market.

There has been much research into various ways of analyzing the stock market as a means of facilitating intelligent investment decisions. These "intelligent decisions" are paramount to the success of an investment, and will be examined in this experiment.

Proposed system

- Two versions of prediction system will be implemented; one using linear regression and other using Support Vector Machines.
- The experimental objective will be to compare the forecasting ability of machine learning algorithms.
- We will test and evaluate both the systems with same test data to find their prediction accuracy.

ISSN PRINT 2319 1775 Online 2320 7876
Research paper UGC CARE Listed (Group -I) Journal Volume 11, Iss 8 , 2022

System requirements

The software requirements specification is produced at the culmination of the analysis task. The function and performance allocated to software as part of system engineering are refined by establishing a complete information description as functional representation of system behavior, an indication of performance requirements and design constraints, appropriate validation criteria.

System architecture

Design is a multi- step that focuses on data structure software architecture, procedural details, procedure etc… and interface among modules. The design procedure also decode the requirements into presentation of software that can be accessed for excellence before coding begins. Computer software design change continuously as novel methods; improved analysis and border understanding evolved. Software proposal is at relatively primary stage in its revolution.

Therefore, software design methodology lacks the depth, flexibility and quantitative nature that are usually associated with more conventional engineering disciplines. However methods for software designs do exit, criteria for design qualities are existing and design notation can be applied.

ARCHITECTURE DIAGRAM

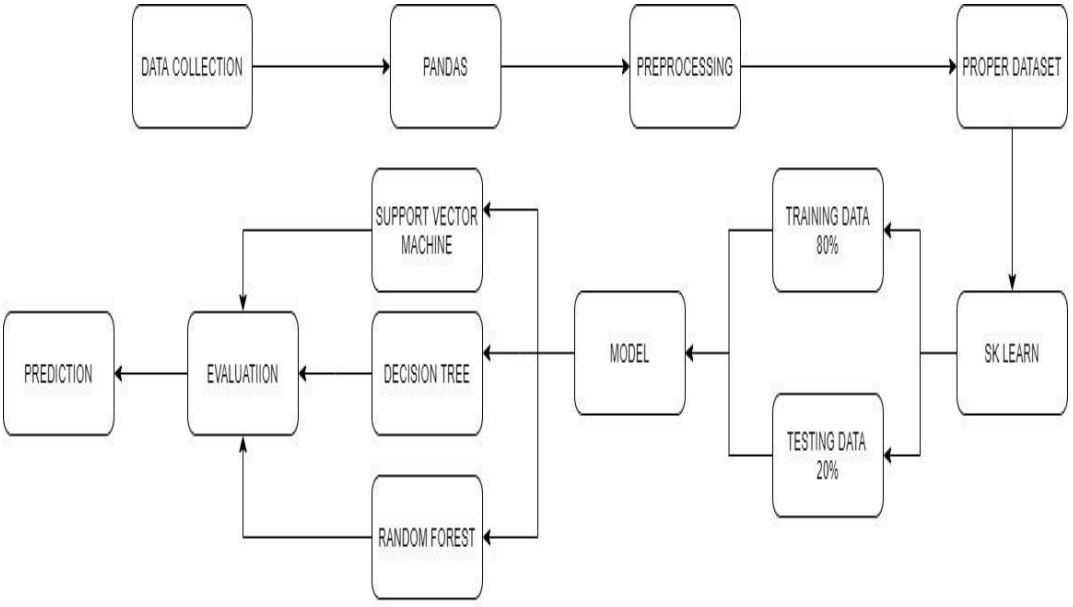

Fig 1: system design

Module description

Collecting Dataset

- Data Collection is one of the most important tasks in building a machine learning model. We collect the specific dataset based on requirements from internet. The dataset contains some unwanted data also. So first we need to pre-process the data and obtain perfect data set for algorithm.

Pre-processing

- It is the gathering of task related information based on some targeted variables to analyse and produce some valuable outcome. However, some of the data may be noisy, i.e. may contain inaccurate values, incomplete values or incorrect values. Hence, it is must to process the data before analysing it and coming to the results. Data pre-processing can be done by data cleaning, data transformation, data selection. Data cleaning includes Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies. Data transformation may include smoothing, aggregation, generalization, transformation which improves the quality of the data. Data selection includes some methods or functions which allow us to select the useful data for our system.

Data input

- Dataset values converted into array values which is going to given to the algorithm to find accuracy. Select the algorithm based on the accuracy and analyse the data by using the algorithm.

Algorithm

Support Vector Machine

"Support Vector Machine" (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

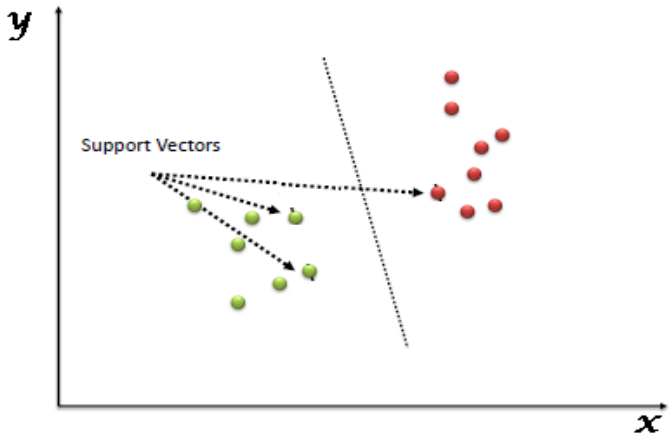

Fig 2: SVM Graph

Support Vectors are simply the co-ordinates of individual observation. The SVM classifier is a frontier which best segregates the two classes (hyper-plane/ line).

Decision tree

Decision tree learning is one of the most successful techniques for supervised classification learning.

Decision tree or a classification tree is a tree in which each internal (non-leaf) node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

A decision tree is a simple representation for classifying examples. A decision tree or a classification tree is a tree in which each internal node is labeled with an input feature. The arcs coming from a node labeled with a feature are labeled with each of the possible values of the feature. Each leaf of the tree is labeled with a class or a probability distribution over the classes.

In this method a set of training examples is broken down into smaller and smaller subsets while at the same time an associated decision tree get incrementally developed. At the end of the learning process, a decision tree covering the training set is returned.

The key idea is to use a decision tree to partition the data space into cluster (or dense) regions and empty (or sparse) regions.

Random Forest

The below diagram explains the working of the Random Forest algorithm:

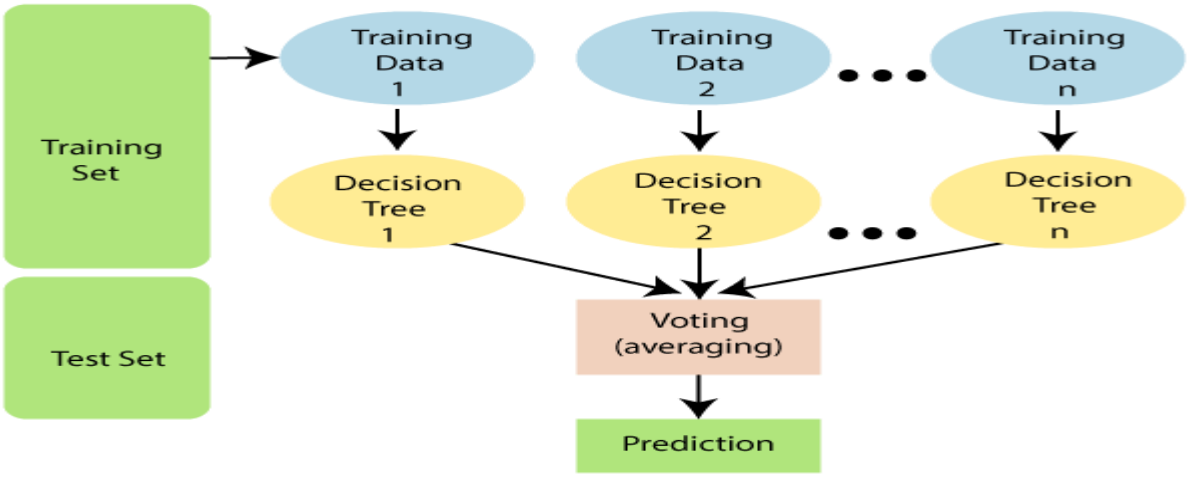

Fig 3: random forest proces**s**

Step-1: Select random K data points from the training set.

Step-2: Build the decision trees associated with the selected data points (Subsets).

Step-3: Choose the number N for decision trees that you want to build.

Step-4: Repeat Step 1 & 2.

Step-5: For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.Random forest, like its name implies, consists of a large number of individual decision trees thatoperate as an ensemble. Each individual tree in the random forest

spits out a class prediction and the class with the most votes becomes our model's prediction (see figure below).

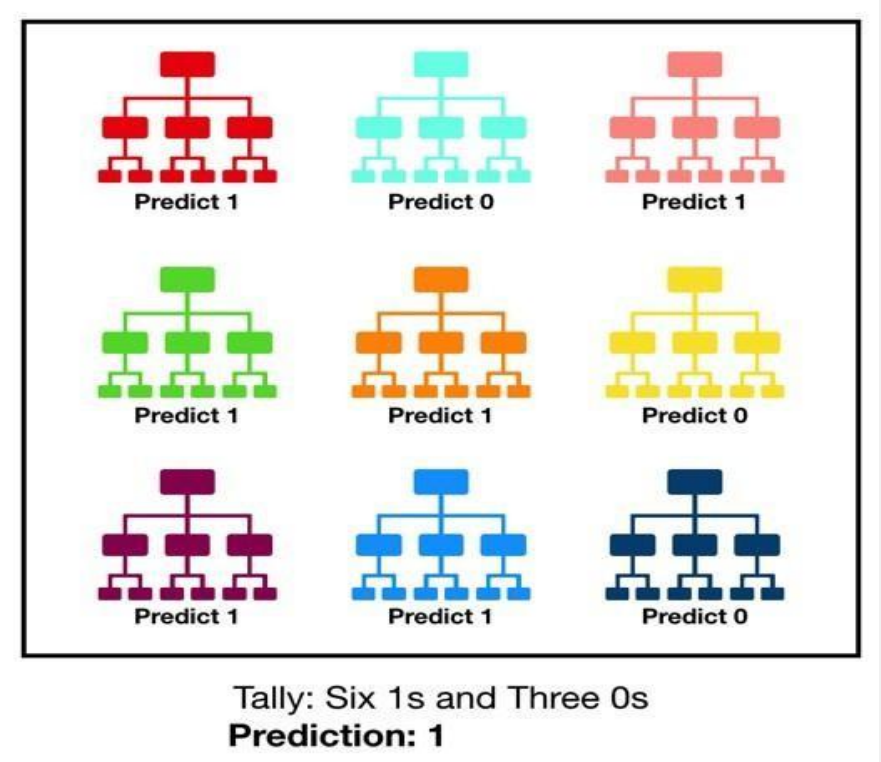

Fig 4: Visualization of a Random Forest Model Making a Prediction

The fundamental concept behind random forest is a simple but powerful one — the wisdom of crowds. In data science speak, the reason that the random forest model works so well is:

A large number of relatively uncorrelated models (trees) operating as a committee will outperform any of the individual constituent models.

The low correlation between models is the key. Just like how investments with low correlations (like stocks and bonds) come together to form a portfolio that is greater than the sum of its parts, uncorrelated models can produce ensemble predictions that are more accurate than any of the individual predictions. The reason for this wonderful effect is that the trees protect each other from their individual errors (as long as they don't constantly all err in the same direction). While some trees may be wrong, many other trees will be right, so as a group the trees are able to move in the correct direction. So the prerequisites for random forest to perform well are:

1. There needs to be some actual signal in our features so that models built using those features do better than random guessing.
2. The predictions (and therefore the errors) made by the individual trees need to have low correlations with each other.

Result

- Based on that dataset we can get the result used our machine learning algorithm to predict the result.
- It will show the future values of particular stock.

Python platform

Apart from Windows, Linux and MacOS, C Python implementation runs on 21 different platforms. Iron Python is a .NET framework based Python implementation and it is cabable of running in both Windows, Linux and in other environments where .NET framework is available.

References

[1] PhishMe Q1 2016 Malware Review. [Online]. Available: https://phishme.com/proposed system/phishme-q1-2016-malware-review/

[2] A. Belabed, E. Aimeur, and A. Chikh, ‘‘A personalized whitelist approach for phishing webpage detection,’’ in Proc. 7th Int. Conf. Availability, Rel. Security (ARES), Aug. 2012, pp. 249–254.

[3] Y. Cao, W. Han, and Y. Le, ‘‘Anti-phishing based on automated individual white-list,’’ in Proc. 4th ACM Workshop Digit. Identity Manage., 2008, pp. 51–60.

[4] T.-C. Chen, S. Dick, and J. Miller, ‘‘Detecting visually similar Web pages: Application to phishing detection,’’ ACM Trans. Internet Technol., vol. 10, no. 2, pp. 1–38, May 2010.

[5] N. Chou, R. Ledesma, Y. Teraguchi, D. Boneh, and J. C. Mitchell, ‘‘Clientside defense against Web-based identity theft,’’ in Proc. 11th Annu. Netw. Distrib. Syst. Security Symp. (NDSS), 2004, pp. 1–16

[6] C. Inc. (Aug. 2016). Couldmark Toolbar. [Online]. Available: http://www.cloudmark.com/desktop/ie-toolbar

[7] J. Corbetta, L. Invernizzi, C. Kruegel, and G. Vigna, ‘‘Eyes of a human, eyes of a program: Leveraging different views of the Web for analysis and detection,’’ in Proceedings of Research in Attacks, Intrusions and Defenses (RAID). Gothenburg, Sweden: Springer, 2014.

[8] X. Deng, G. Huang, and A. Y. Fu, ‘‘An antiphishing strategy based on visual similarity assessment,’’ Internet Comput., vol. 10, no. 2, pp. 58–65, 2006.

[9] Z. Dong, K. Kane, and L. J. Camp, ‘‘Phishing in smooth waters: The state of banking certificates in the US,’’ in Proc. Res. Conf. Commun., Inf. Internet Policy (TPRC), 2014, p. 16.