
“Evolutionary Hybrid Approach for Stock market Prediction for Trading ”

Mr.Prabhat Sharma

¹(PG Student,VITNagpur)

Abstract- *The application will contain data mining from previous record of the stock market, which will be useful to get exact market value of each share from share market. In this application final user will be able to get idea of increasing decreasing ratio of the stock share on particular month or year of particular share. Data mining will be done in every way, also the results will get displayed on flowchart. k-means clustering is a method of vector quantization, originally from signal processing, that is popular for cluster analysis in data mining. k-means clustering aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean, serving as a prototype of the cluster. This results in a partitioning of the data space into Voronoi cells. The problem is computationally difficult (NP-hard); however, there are efficient heuristic algorithms that are commonly employed and converge quickly to a local optimum. These are usually similar to the expectation-maximization algorithm for mixtures of Gaussian distributions via an iterative refinement approach employed by both algorithms. Additionally, they both use cluster centers to model the data; however, k-means clustering tends to find clusters of comparable spatial extent, while the expectation-maximization mechanism allows clusters to have different shapes.*

Keywords—

(Stock Market Prediction, Existing System, Propose System, Technology, Result)

1. INTRODUCTION

The project is mobile based application where user will be able to see the existing stock market data, which will be use full for making guess about next condition of future market. A functionally accented, diverse, efficient and flexible financial system is vital to the national objective of creating a market-driven, productive and competitive economy. Today markets of varying maturity exist in

equity, debt, commodities and foreign exchange. Of the 25 stock markets in the country, the most important is Bombay Stock Exchange. In this work we attempt to generate prediction rules scheme for stock price movement at Bombay Stock Exchange using an important Soft Computing paradigm viz., Rough Fuzzy Multi Layer Perception. The use of Computational Intelligence Systems such as Neural Networks, Fuzzy Sets, Genetic Algorithms, etc. for Stock Market Predictions has been widely established. The process is to extract knowledge in the form of rules from daily stock movements. These rules can then be used to guide investors. To increase the efficiency of the prediction process, Rough Sets is used to discretize the data. The methodology uses a Genetic Algorithm to obtain a structured network suitable for both classification and rule extraction. The modular concept, based on divide and conquer strategy, provides accelerated training and a compact network suitable for generating a minimum number of rules with high certainty values. The concept of variable mutation operator is introduced for preserving the localized structure of the constituting Knowledge Based sub-networks, while they are integrated and evolved. Rough Set Dependency Rules are generated directly from the real valued attribute table containing Fuzzy membership values. The paradigm is thus used to develop a rule extraction algorithm. The extracted rules are compared with some of the related rule extraction techniques on the basis of some quantitative performance indices. The proposed methodology extracts rules which are less in number, are accurate, have high certainty factor and have low confusion with less computation time. To the best of our knowledge, this is the first Rough Fuzzy Multi Layer Perception method for discovering stock prediction rules of Bombay Stock Exchange. The major prima face of this Paper is to present a generic Stock Price Prediction Model using a Modular Evolutionary Approach for designing a Hybrid Connectionist System in Soft Computing framework for both Classification and Rule Generation. The basic building block used is the Rough Fuzzy Multi Layer

Perception (RFMLP) Networks [4]. The model is expected to extract knowledge in the form of rules from daily stock movements of the Bombay Stock Exchange (BSE) that would guide investors whether to buy, sell or hold a stock. To increase the efficiency of the Prediction process, Rough Set with Boolean Reasoning (RSBR) Discretization Algorithm is used to discretize the data. The original classification includes splitting the tasks into several subtasks and a number of RFMLP Networks are obtained for each subtask. The sub-network modules are integrated in a particular manner so as to preserve the crude domain knowledge which was encoded in them using Rough Sets. The pool of integrated networks is then evolved using a Genetic Algorithms with a restricted adaptive or variable mutation operator that utilizes the domain knowledge to accelerate training and preserves the localized rule structure as potential solutions. The parameters for input and output fuzzy membership functions of the network are also tuned using Genetic Algorithms together with the link weights. The existing procedure has been modified for the generation of Rough Set Dependency Rules for handling directly the real valued attribute table containing fuzzy membership values. This helps in preserving all the class representative points in the Dependency Rules by adaptively applying a threshold that automatically takes care of the shape of membership functions. In this Knowledge Based Network design all possible inference rules contribute to the final solution. The use of Genetic Algorithms in this context is beneficial for modeling Multimodal distributions, since all major representatives in the population are given fair chance during Network synthesis. Next the Rule Extraction Algorithm is presented. The performance of the generated rules is evaluated quantitatively. Two new measures are accordingly defined indicating the certainty and confusion in the decision. These new indices are used along with some existing measures to evaluate the quality of the rules. A quantitative comparison of the Rule Extraction Algorithm is made with some existing ones like the Subset Method [9], M of N Method [35], X2R Method [20], etc. on datasets of the BSE.

2. Existing System:

Existing system for calculating the market review is like everyone has to maintain the record by excel sheet and using those excel sheet we can get the idea what will be the rates in future. This is not accurate method, this may

lead to cause problems in prediction. Prediction itself mean that it may happen, that is not confirmed, but the probability to happen the things matters. If the success rate of of prediction is great, we can call it as perfect prediction. We can not do perfect prediction of share market using the manual record maintain using the excel sheets, hard papers, or some basic software's. We need some strong software which, which will help to generate good prediction.

3. Proposed System.

The application is mobile application, That one is created on native android platform. The application shows existing records of stock market. We can synchronise the data runtime, we can get saved stock data, we can check all of these in all currencies. The application shows all of these things in pi chart and also in different formats of charts.

Methodology Used

1) Anroid :

The Android software development kit (SDK) includes a comprehensive set of development tools.[4] These include a debugger, libraries, a handset emulator based on QEMU, documentation, sample code, and tutorials. Currently supported development platforms include computers running Linux (any modern desktop Linux distribution), Mac OS X 10.5.8 or later, and Windows 7 or later. As of March 2015, the SDK is not available on Android itself, but software development is possible by using specialized Android applications.[5][6][7]

Until around the end of 2014, the officially supported integrated development environment (IDE) was Eclipse using the Android Development Tools (ADT) Plugin, though IntelliJ IDEA IDE (all editions) fully supports Android development out of the box,[8] and NetBeans IDE also supports Android development via a plugin.[9] As of 2015, Android Studio,[10] made by Google and powered by IntelliJ, is the official IDE; however, developers are free to use others, but Google made it clear ztto focus on Android Studio as the official Android IDE.[11] Additionally, developers may use any text editor to edit Java and XML files, then use command line tools (Java Development Kit and Apache Ant are required) to

create, build and debug Android applications as well as control attached Android devices (e.g., triggering a reboot, installing software package(s) remotely).[12]

Enhancements to Android's SDK go hand in hand with the overall Android platform development. The SDK also supports older versions of the Android platform in case developers wish to target their applications at older devices. Development tools are downloadable components, so after one has downloaded the latest version and platform, older platforms and tools can also be downloaded for compatibility testing.[13]

Android applications are packaged in .apk format and stored under /data/app folder on the Android OS (the folder is accessible only to the root user for security reasons). APK package contains .dexfiles[14] (compiled byte code files called Dalvik executables), resource files, etc.

The Android 3.1 platform (also backported to Android 2.3.4) introduces Android Open Accessory support, which allows external USB hardware (an Android USB accessory) to interact with an Android-powered device in a special "accessory" mode. When an Android-powered device is in accessory mode, the connected accessory acts as the USB host (powers the bus and enumerates devices) and the Android-powered device acts as the USB device. Android USB accessories are specifically designed to attach to Android-powered devices and adhere to a simple protocol (Android accessory protocol) that allows them to detect Android-powered devices that support accessory mode.[29]

2) Java

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented,[14] and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),[15] meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[16] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming

languages in use,[17][18][19][20] particularly for client-server web applications, with a reported 9 million developers.[21] Java was originally developed by James Gosling at Sun Microsystems(which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses. As of May 2007, in compliance with the specifications of the Java Community Process, Sun relicensed most of its Java technologies under the GNU General Public License. Others have also developed alternative implementations of these Sun technologies, such as the GNU Compiler for Java (bytecode compiler), GNU Classpath (standard libraries), and IcedTea-Web (browser plugin for applets).

The latest version is Java 8 which is the only version currently supported for free by Oracle, although earlier versions are supported both by Oracle and other companies on a commercial basis.

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991.[22] Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time.[23] The language was initially called Oak after an oaktree that stood outside Gosling's office. Later the project went by the name Green and was finally renamed Java, from Java coffee.[24] Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[25]

Sun Microsystems released the first public implementation as Java 1.0 in 1995.[26] It promised "Write Once, Run Anywhere" (WORA), providing no-cost run-times on popular platforms. Fairly secure and featuring configurable security, it allowed network- and file-access restrictions. Major web browsers soon incorporated the ability to run Java applets within web pages, and Java quickly became popular. The Java 1.0 compiler was re-written in Java by Arthur van Hoff to comply strictly with the Java 1.0 language specification.[27] With the advent of Java 2 (released initially as J2SE 1.2 in December 1998 – 1999), new versions had multiple configurations built for different

types of platforms. J2EE included technologies and APIs for enterprise applications typically run in server environments, while J2ME featured APIs optimized for mobile applications. The desktop version was renamed J2SE. In 2006, for marketing purposes, Sun renamed new J2 versions as Java EE, Java ME, and Java SE, respectively.

In 1997, Sun Microsystems approached the ISO/IEC JTC 1 standards body and later the Ecma International to formalize Java, but it soon withdrew from the process.[28][29][30] Java remains a de facto standard, controlled through the Java Community Process.[31] At one time, Sun made most of its Java implementations available without charge, despite their proprietary software status. Sun generated revenue from Java through the selling of licenses for specialized products such as the Java Enterprise System.

On November 13, 2006, Sun released much of its Java virtual machine (JVM) as free and open-source software, (FOSS), under the terms of the GNU General Public License (GPL). On May 8, 2007, Sun finished the process, making all of its JVM's core code available under free software/open-source distribution terms, aside from a small portion of code to which Sun did not hold the copyright.[32]

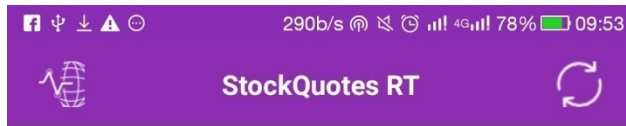
Sun's vice-president Rich Green said that Sun's ideal role with regard to Java was as an "evangelist".[33] Following Oracle Corporation's acquisition of Sun Microsystems in 2009–10, Oracle has described itself as the "steward of Java technology with a relentless commitment to fostering a community of participation and transparency".[34] This did not prevent Oracle from filing a lawsuit against Google shortly after that for using Java inside the Android SDK (see Google section below). Java software runs on everything from laptops to data centers, game consoles to scientific supercomputers.[35] On April 2, 2010, James Gosling resigned from Oracle.

4. Future work

In future we can integrate many other things in this application. We can show runtime changing rates of the stock market, we can show live stock market analysis and progress going on live basis. The algorithm has converged when the assignments no longer change. Since both steps optimize the WCSS objective, and there only exists a

finite number of such partitioning, the algorithm must converge to a (local) optimum. There is no guarantee that the global optimum is found using this algorithm. The algorithm is often presented as assigning objects to the nearest cluster by distance. The standard algorithm aims at minimizing the WCSS objective, and thus assigns by "least sum of squares", which is exactly equivalent to assigning by the smallest Euclidean distance. Using a different distance function other than (squared) Euclidean distance may stop the algorithm from converging. Various modifications of k-means such as spherical k-means and k-medoids have been proposed to allow using other distance measures. One of the other areas of further research is to evolve simple trading strategies after the model predicts the stock trend. It states that if the model predicts as up or down, which actions to take and how much stock to buy or sell or when to buy or sell. This is actually a complementary for the prediction model as after predictions it gives the afterward instructions to keep the investors in the best financial position.

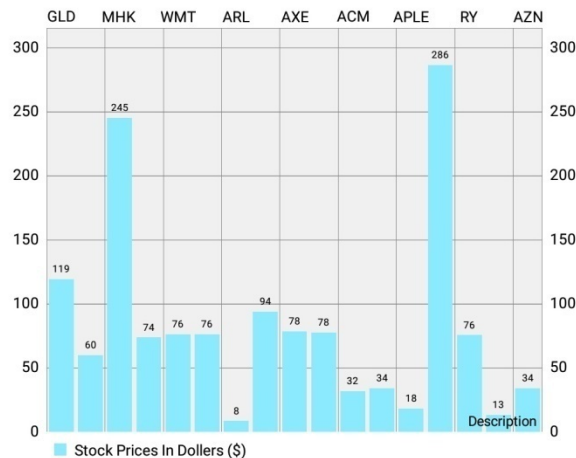
5. .Result:



Main Menu



International Market



sts	Name	Last	Chng %
	GLD	119.26	1.02

S

References

[1] R. P. Schumaker and H. Chen, "A discrete stock price prediction engine based on financial news," *Computer*, vol. 43, no. 1, pp. 51–56, 2010.

[2]..akov Amihud, Haim Mendelson, and Lasse Heje Pedersen. *Liquidity and Asset Prices. Foundations and Trends in Finance* , 1(4):269{364, August 2007.

[3] T. H. Nguyen and K. Shirai, "Topic modeling based sentiment analysis on social media for stock market prediction," in *Proceedings of the 53rd Annual Meeting of the Association for Computational*

Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, 2015, pp.1354–1364.

[4] Y. Tsuji, K. Komiya, and Y. Kotani, "Stock price prediction corresponding to more than one companies on the web news," in *IEICE technical report SIG-IBISML SIG-NLC (in Japanese)*, vol. 110, no. 476, 2011, pp. 109–113.

[5] K. Ichinose and K. Shimada, "Stock market prediction from web news using filtering and machine learning," in *IEICE technical report SIGNALC (in Japanese)*, vol. 115, no. 70, 2015, pp. 9196.

[6] K. Izumi, T. Goto, and T. Matsui, "Trading tests of long-term market forecast by text mining," in *Proceedings of 2010 IEEE International Conference on Data Mining Workshops (ICDMW)*, 2010, pp. 935–942.

[7] J. Bollen, H. Mao, and X.-J. Zeng, "Twitter mood predicts the stockmarket," *Journal of Computational Science*, vol. 2, no. 1, pp. 1–8, 2011.

[8] V. N. Vapnik, *Statistical Learning Theory*. Wiley, 1999.

[9] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–408, 1958.

[10] T. Joachims, *Learning to Classify Text Using Support Vector Machines – Methods, Theory, and Algorithms*. Kluwer/Springer, 2002.

[11] H. Lee, M. Surdeanu, B. MacCartney, and D. Jurafsky, "On the importance of text analysis for stock price prediction," in *Proceedings of the 9th edition of the Language Resources and Evaluation Conference (LREC)*, 2014, pp. 1170–1175.

[12]. J. Bollen, H. Mao, and X. Zeng. *Twitter mood predicts the stock market. Journal of Computational Science* , 2(1):1{8, 2011.

[13]. D. Harvey, S. Leybourne, and P. Newbold. *Testing the equality of prediction mean*