Predicting The Stock Prices Of Tech Companies submitted To International Journal For Science And Advance Research In Technology

Rishabh Upadhyay
Dept of Information Technology
Maharaja Agrasen Institute of Technology

Abstract- *Predicting stock prices has been the dream of many people; however, the unpredictable nature of the world has rendered this as merely a pipe dream. This project focuses on developing an application to determine whether they will rise or fall and tests the overall concept on the stock value of tech companies.*

The dataset used is the stock prices over time of at least 50 tech companies listed in the NASDAQ, taken from a database curated by Quandl. Data mining is applied to see if there is any correlation between the price of any tech company's stock and such things as: cost of materials, fuel prices, etc.

Data that seems to correlate to a change in the price of a tech company's stock is then used to create a "tag" associated with that company. These tags are used to build a decision tree to predict the rise or fall of stock prices for that company. The decision tree is supplied with data from two databases; one containing each company's historical data paired with the historical data of their tags, and one containing up-to-date data for the tags. The graphical user interface component requests a stock symbol, builds and uses the decision tree and outputs the result.

I. INTRODUCTION

The project was chosen because of the potential gains to be had in investing in the stock market, particularly in the tech industry. Start-up tech companies represent a potential windfall in returns; however, they are also very risky. The stock price of Amazon, for example, has experienced an 8000% increase over the past twenty years [1]. On the other hand, Zynga Inc. has dropped from $14.7 to $2.5 per share in this decade [2]. The desire is to develop an application to put peoples' money to work by carefully considering all the factors that affect the stock price of a company.

The process to achieve this is outlined in the following sections. Section ‘Design Consideration’ discusses the various design considerations that went into developing the application, such as what things affect stock prices and how to use the data to predict stock prices. Section 3 gives an overview of the architecture of the MySQL database and how we connect to it. Section 4 describes the implementation of the decision tree, retrieval of historical data, and acquisition of new data from Quandl. Section 5 talks about the lessons learned and challenges faced while completing this project. Finally, Section 6 closes with the current status and possible future work for this project.

II. DESIGN CONSIDERATIONS

At first, multiple elements were considered as possibly affecting the price of a company's stock, this included everything from politics to natural disasters. It was decided that many of these things were beyond the scope of this project or were too difficult to model. Instead, the index values of industries that a company relies upon were chosen.
The idea is that the value of an index reflects the health of its industry. This vitality, in turn, is thought to affect the stock price of any company that relies upon that industry. For each company, indices that are thought to affect its stock were collected as a list of "tags". Indices to back up each tag were found in a NASDAQ database located on Quandl[3].

The data of the companies and their corresponding tags were analysed using R/Rattle. Different modelling tools like decision trees, clustering, missing values, and correlation analysis, were considered as possibilities for accurately predicting the stock price. After comparing the classification error of each model, it was decided to go with decision trees as the means of making predictions. The decision tree model performs reasonably well and creates a set of rules by which new instances can easily be classified.

In the analysis of the data, it was found that the value of a company’s stock did not depend on its associated tags meeting some value threshold rather; the value depends on whether an index rises or falls in value. In this sense, binary

values of true (rise) and false (fall) can be used in place of numeric values. When decision trees were built using this binary form, the jump in performance was impressive. On average, we achieved 70% success using decision trees built using binary values rather than numeric ones.

III. ARCHITECTURE

To build a decision tree, historical information for a company and its associated tags are used. This historical information is stored in a database in its own table. Given the structured nature of the data and its similarity, it was decided that a NoSQL database was not needed. For this reason, MySQL was chosen as the backing database for the project.

The local host of MySQL's server is used to store the tables for every company. This allows the JSP application to connect, query information, and use that information to build the decision tree for a target company. The connection is made possible by a JDBC driver downloaded from the MySQL website. This driver provides the application with a Java library to connect to and query the database from within a Java program.

To build the decision tree, the application also gets data from Quandl's website [3]. It was decided to not store this data locally in the MySQL database since only the most recent data is needed. Conveniently, there is a JDBC driver available to connect to Quandl from Java.

IV. IMPLEMENTATION

The application is implemented in the Java programming language and is hosted online using Java Server Pages (JSP). Users choose a stock symbol in an HTML document, this is then passed to a Java class and used to retrieve historical data from the database and the associated tags from a text file(profile.txt). The Java class then uses the tags to extract the correct historical data and uses the result to builda decision tree for the company represented by the stock symbol.

This decision tree is created using Weka's machine learning Java library. Although the choice to use decision trees was based upon the performance of R's decision tree algorithm, R's decision tree is a purely statistical model; it cannot be used to classify new instances, nor can it be used within object oriented programs. Weka is built using Java and its machine learning library for Java provides decision trees that can be used to classify new instances in an object oriented context.

One consequence of using Weka is that data read in from MySQL must be converted to a string representation of an ARFF file. Weka's decision tree performs at its best when supplied with these ARFF files. When CSV files are used instead, there is a drop in performance; indeed, Weka's algorithm for decision tree building is unable to determine proper target values without the metadata provided by ARFF files.

To create new instances to classify, a JDBC connection with Quandl's database is used to acquire the latest data for a particular company's tags. This data is then processed into Boolean values, packaged into an instance with ARFF metadata, and passed into the decision tree for classification. A JDBC connection with Quandl is a desirable method of acquiring new data. Possible alternatives of connecting to Quandl would have involved downloading les to a client's machine; obviously this is a poor choice. The output of the decision tree is then returned to the JSP application and displayed to the user.

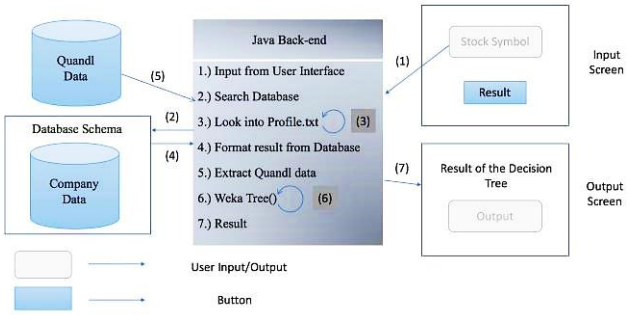

Figure 1: Final Application Design

V. LESSONS LEARNED

The initial results of analysing the data in Rattle were surprising. When viewing correlations between different attributes and relationships that were expected and were not seen. For example, in viewing the correlation between Apple's attributes, there was very little correlation between the closing price of Apple's stock and the value of the software index.

Considering that Apple is heavily involved in the software industry, this result seems to make little or no sense. However, Apple's closing price did have a correlation to the consumer electronics index. Conversely, Autodesk showed a strong correlation between its closing price and the value of the software index. This makes sense, since Autodesk is involved with software and cloud computing.

Because of these unexpected results, particularly Apple's lack of correlation with the software index, it would seem that the index chosen for the software tag is not specific enough, it could be that too many things are included under the software tag. For example, cloud computing and software in general are both part of the current software tag. Autodesk's correlation to this index makes sense because it deals with software and cloud computing. Apple, however, does not focus on cloud computing, so of course its stock prices would not correlate to cloud computing. This problem was ameliorated by finding more specific indices to address the specialties of different tech companies.

Although finding more specific indices revealed more relationships between the data, the performance of the decision trees was still poor. Indeed, R's decision tree algorithm was creating trees that were enormous even after pruning; yet, despite their size they had a high rate of error.

Eventually, it was realized that the rise and fall in price of a company's stock is not dependent on the magnitude of its tags values. No matter the time or context, the price of a company's stock is dependent on the rise or fall of its dependencies. After replacing the numerical values of the data with Boolean values that represent this rise and fall, the performance of the decision trees improved dramatically. In some cases, it rose more than 20%.

The original intention of this project had been to develop a master algorithm that would have been used to determine the rise or fall of a company's stock. When analysing the data generated by the decision tree model, it was observed that each company's decision tree was unique. This led to the realization that this uniqueness is a more powerful determination of a company's stock price than a universal, one-size- fits-all, master algorithm.

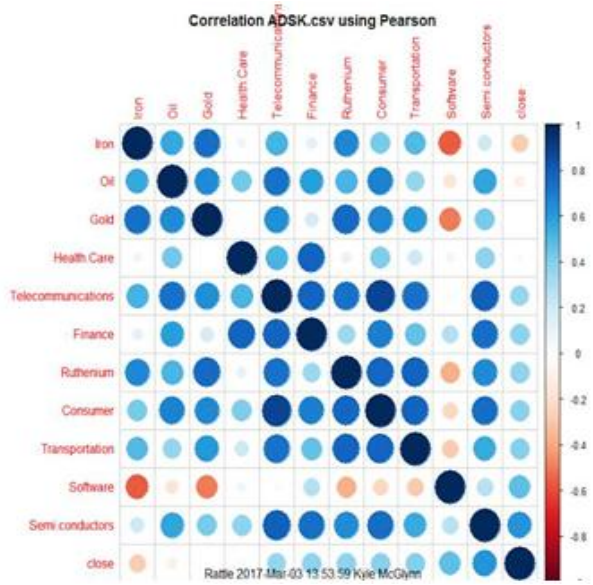

figure 2: Graph for Autodesk (expected correlation)

VI.CURRENT STATUS AND FUTURE WORK

Currently, the application is able to take in a stock symbol, read in the appropriate information from MySQL, retrieve information from Quandl, build the decision tree, and classify the new instance. Unfortunately, there is an issue reading new information from Quandl for some of the tags. It was found that some tables in Quandl have null values. The JDBC connection to Quandl is unable to skip these entries despite query constraints because this data lacks types that would normally be informed by a schema.

So far, no work around has been found for this issue. It was thought that a schema could be enforced upon the data using the JDBC connection; however, no alternative schema is available to be called, nor can a new schema be inserted and enforced upon the data. While it is true that the information could be downloaded as les, cleaned, and processed on the client machine, as said above this is a truly undesirable situation. No client would want to download les onto their machine and then be left with the task of deleting them. So it is that a work around shall be left for future work.

As of now, the application is only able to determine if the price of a company's stock will go up or down. It is not able to determine the magnitude of that change. There is the possibility that this situation can be remedied by altering the numeric data into numeric categories of delta values. These delta values are calculated by finding the difference between consecutive records' closing values. For example,determining the delta value of the i^{th} record is calculated by taking the difference between the closing values of the $(i+1)^{th}$ and i^{th} records. The idea is that by using multiple, numeric categories,

both high performing decision trees and numeric predictions for the increase or decrease of a company's stock value can be acquired.

Unfortunately, initial experiments with this form of the data did not perform well. It is possible that too many numeric categories are created, or that the process is somehow awed; however, the potential of this approach is tantalizing, so further e orts shall be made to improve upon it.

REFERENCES

[1] Amazon. https://finance.yahoo.com/quote/AMZN?p=AMZN.
[2] Zynga. https://finance.yahoo.com/quote/ZNGA/?p=ZNGA
[3] Quandl-indices. https://www.quandl.com/data/ NASDAQOMX-NASDAQ-OMX-Global-Index-Data.