Stock Market Analysis: Algorithmic Trading

Shivam Sharma¹, Shrey Kaushik², Shubham Srivastava³, Vikrant Jatain⁴, Nagarathna N⁵

¹,²,³,⁴Student, Department of Computer Science and Engineering, B.M.S College of Engineering, Bangalore-India
³Assistant Professor, Department of Computer Science and Engineering, B.M.S College of Engineering, Bangalore-India

Abstract - Stock Market Analysis is a platform that is used in the evaluation National Stock Exchange of India with over 5000 Indian companies. Algorithmic trading gives us greater profits with a hybrid model on LS-SVM (Least square support vector machine) to avoid overfitting and combining various modules like analysis, prediction, automated trading, stock guru analysis, search, developer, trend analysis, financial reports, etc.

Algorithmic trading drives about 30% of the trading volumes in Indian equity markets and the percentage is on the rise every day. In the west this percentage is somewhere around 70-80 percent. This gap combined with the low investments from the Indian households is a serious indicator in our economy.

Hence the shift to algorithmic trading with the utilization of Artificial Neural Network for price prediction helps us in dealing with optimizing large sets with multiple parameters. A stochastic model is used like that of Markov model where randomness is a key factor in consideration while making predictions.

The use of machine learning algorithms or machine learning model are proving to be quite useful for stock price prediction, trend analysis, charting, technical analysis while dealing with real time data.

2. RELATED WORK

There are several existing systems like quantitative trading or high frequency trading or arbitrage, that are available to large multi corporations, investment banks and hedge fund managers. The systems have been personally tailored to meet the investors requirements and all the fundamental and technical analysis done by the high computing servers which is available to only a few people[11]. Whereas the larger number of retail or common investors with a small portfolio, like a common Indian household investor with little capital must rely on traditional means of accessing stock market data with limited analysis to be performed[7].

The lack of mathematical and technical knowledge of a common man also is an inhibiting factor to algorithmic trading or stock market analysis[1]. The existing third party stockbroker’s websites, google finance or yahoo finance provide various technical charts and live stock data with no access to perform user algorithms or analysis on the given data[13]. There is just a one-way communication from the websites to users. The ability to perform statistical analysis on the database isn’t provided[2]. Access to a company’s financial reports must be manual either by visiting the company website or viewing the company profile in the website[34]. Newspapers, financial journals and online websites like economic times provide recommendations from various
stock experts or analysts and other trend details in the current market[9].

Identifying social emotions and detecting the trend through the web data. News & blogs also give us a great insight on current trends[13]. The semantics and affective resources for opinion mining becomes crucial and the structure of data and the way user classifies it[15].

When user obtains such varied forms of data from different sources, we start looking for new financial stock market forecast measures using current data mining technique[8]. Simple LSTM neural network with character level embeddings for stock market forecasting, using only financial news as predictor[29]. We have also come across the usage of optimal Long Short-Term Memory (0- LSTM) deep learning and adaptive Stock Technical Indicators (STIs)[8]. They’ve also evaluated the model for taking buy sell decisions at the end of the day at around 3:30pm IST for Indian Markets[10].

The several advantages with these implementations are that algorithms like PSO algorithm helps in selecting the best free parameters combination for LS-SVM (Least square support vector machine) to avoid overfitting[18], also they help in selecting the best free parameters combination for LS-SVM (Least square support vector machine) to avoid overfitting[18]. The use of machine learning algorithms or machine learning models are proving to be quite useful for stock price prediction, trend analysis, charting, technical analysis while dealing with real time data[20]. Also the current financial report formats obtained for every company are standardized and easy to extract, compute and analyze[23].

Various disadvantages are that there is no information on the success ratio or the trust factor on the analysts that provide such recommendations or predictions[25] and the information provided in a one-way communication and the user cannot statistically operate on the data[28]. The main problem faced is the uncertain trend in the stock market, the uncertain trends justify that there might be risk involved with stock for various time periods[4]. The twitter sentiment analysis of majority of tweets often is an outcome of the media presence of the company and sentiments don’t factor based on company portfolio [32].

3. PROPOSED FRAMEWORK

In the framework, we use a web data scraper with python to fetch, analyse and classify the datasets using algorithms which increases the possibilities of profits. The raw data from various resources are stored in a MySQL Database to be later accessed by the Django Framework based website with templates to give users a great visual experience and also provide necessary tools. The stock data present in the database are also to be organized based on gainers and losers in various periods of time like daily, weekly, monthly, quarterly, etc. This way we can perform simple analysis that can be easily understood even by a beginner.

The implementation is divided into modules. The developer module is for the traders with a little technical SQL knowledge to perform their own statistical analysis on the database of stock data. A search module to get all the stock data with financial reports and technical charts presented in an interactive dynamic web page. The analysis module is to provide simple analysis on the tables present in the database. Prediction module is to combine all the raw data we have obtained and predict the probability of bullish or bearish movement of any stock. Stock guru analysis to determine the success ratio of the analysts who provide recommendations or predictions. Trend analysis and automated buy and sell modules would help users place orders in the same platform with a link to his or her bank and demat account. The integration of all these modules in our python based framework will help us provide a great platform for an investor.

4. IMPLEMENTATION

The project has been implemented in python based Django Framework which consists of a model view template. The model consists of our processed raw data in the form of a structured SQL database, template deals with the UI with html files, view is the link in between the model and template where we send the required data to the template for the user to interact with.

A local Xampp server is used to organise our database to store raw data. This project uses BeautifulSoup, Selenium with Drivers, etc to automate the live extraction and live trading. It takes three hours to fetch all historical data of stock market 5275 companies from Year 2000 to present.

A total of six modules are created and its implementation in detail is mentioned below.

1. Developer Module.
2. Search Module.
3. Analysis Module.
5. Stock Guru Module.

In the project a tables page has also been implemented to create a simple UI and UX for users to view and interact with the backend database as a two way communication in the project. We are trying to automate trading, give normal users a one stop solution and financial reports need not be fetched manually from company websites.

**DEVELOPER MODULE**

The developers who have knowledge of sql or even a basic programmer can use this. He/She will be able to execute his own queries on our database to obtain desired results. On google when we search for some company we only get the data, chart, etc. Here we get the raw data and also enable user to communicate with data and process it so that the user can also apply the algorithms.

The user has been provided with input text to insert the sql query in string format. The string is then extracted on submit via POST method to strip it and remove unwanted characters like ';' if present. The query is then made to run on the database present with the help of mysql.connector. We connect to the database and with the cursor run the query that the user has entered.

The results are fetched using cursor.fetchall() which will return records of the entered analysis query if it is present. In case the query entered by user is wrong or doesn’t return any records, the corner error scenarios are handled and a placeholder message is sent to convey the error message to the user.

Example: -

Developer runs query “select * from d1g” or “select * from d1g where Current>200;”

The code strips the string and removes unwanted characters. Here d1g represents day 1 gainers, and the query runs on the database to provide users the results of the query. Complex queries are also handled by our module.

**SEARCH MODULE**

Search module is a one place platform to view financial statements and stock historical data all in one place. The user is supposed to enter the name of the company to search and find some details and analyse them. A "POST" request is sent and we then match the entered string as a substring of all the company names present in our database, using the query “Select Cticker from Companies where Cname LIKE '%{companyName}%'”. This query with string pattern matching returns the ticker for the company that is used to identify the specific company. The Ticker is then used to extract stock records and present it to the user in descending order of the date using “Select * from Cticker ORDER BY Date DESC”. The result given has High, Low, Open, Close, Adjusted Close prices for the mentioned Date and also the volume of shares bought & sold. In case multiple companies are present or if incorrect names are entered by the user, then the error handling cases will return placeholder message to the user.

This module also fetches live data of its financial yahoo finance website. As soon as user enters the company and then we find ticker we replace the ticker in the link: https://in.finance.yahoo.com/quote/RELIANCE.NS/financials

The ticker here is “RELIANCE.NS” the python drivers will run selenium code to open the mentioned link and fetch the last three years financial record statements from the website

**ANALYSIS MODULE**

This is our analysis module where we now perform preliminary analysis process on our database to generate significant potential companies for users to invest in.

The entire structure of historic stock data have been divided in the time frames as mentioned below:

tablelist=["d1g","d2g","d3g","d1l","d2l","d3l","w1g","w2g","w1l","w2l","m1g","m1l","q1g","q2g","q3g","q4g","q1l","q2g"]

FIG 2: Yahoo financial statements for Reliance Infrastructure Limited – ‘RELIANCE.NS’
The above python code line is to create tables and populate them in the structure. Each table gives significant values on time and gainers or losers. The time frames are daily, weekly, monthly, quarterly, half yearly, yearly. The numbers mention the timeline in past. The end notation ‘g’ or ‘l’ is used to represent gainers or losers.

For example w1g & w2g - It represents the stocks that are weekly gainers in the week 1, current date-5days, and weekly gainers 2 week before- current date -5 to current date -10. All the data are for the trading days and each trading week is generally having 5 trading days.

The list of all stock tickers are obtained in an excel format. The excel format is worked on using pandas library and we run the api to fetch yahoo financial records of each and every company. Try and except statements are present wherever error scenarios could arise. The records are received in descending order from the current date. The following checks are made and based on

```python
currentTime = df['Close'].iloc[total-1]
previousTime = df['Close'].iloc[total-6]
```

Current and previous closing price. We insert the companies into the mentioned tables. Every company runs its code for all the time frames mentioned and sorts it to place it in those tables.

The analysis module also has implementations to fetch only buyers and only sellers data live from the website, so further analysis can be done. Which companies had only buyers of shares or only sellers with volume data. These are highly demanded (only buyers) and highly avoidable (only sellers) companies.

The assortment is done and now the analysis is done on these tables. Each table mentions its title and its code word.

**STOCK GURU MODULE**

A lot of stock guru or analysts come up with predictions like buy shares of a company or sell it. Usually such analysts don’t have any success ratios giving us confidence about them. So from economic times we extract all the recommendations on the particular day and analyse their statement by fetching.

Example : “Buy Maruti Suzuki, target price Rs 6,230: Emkay Global”

The string information is used to extract info like Which company they recommend, Who is recommending, When, Whether to buy or sell. Only Buy and Sell are considered in our implementation.

A selenium web browser opens up economic times website : https://economictimes.indiatimes.com/markets/stocks/
We then start obtaining the block of each recommendation and extract our data present in the block with string processing functions in python and applying our logic to segregate recommendations. The data once extracted is again processed one by one to figure on the day of prediction what price it was and how many days it took to achieve the target. The least number of days it took to reach the target price for buy or sell is searched for in the table and calculated. Companies that don't achieve target have 0 as Achieved.

Some recommendations of companies fetched might not be present in our Database so we filter it out by running a query "select * from stockguru where Achieved IS NOT NULL"

### TRADE MODULE

Automated trade execution platform is present where user can enter company name and stock quantity. It is used to execute our trade on HDFC Securities where we have a trading account to place our orders.

Once the user enters the data, the automated selenium driver will open the website and enter the security credentials which are stored in a secret file within the system. The selenium web driver executes the script which then has a sleep time of 2 seconds and then executes the trade of buy or sell.

### PREDICTION MODULE

The main prediction module is run by simulating our trade or backtesting it with training data and test data. The backtesting gives us the profitability of the strategy that we use with a graph. The simulation is run such that orders of buy or sell placed on the particular day is printed as an output and the calculation of the initial fund that we started at (Here we put 1000000 Rupees) and then after the simulation is done. The final fund is checked for the percentage gain and loss. All such gains and loss are collectively calculated for either a set of top 100 companies or all companies.

The python code uses talib library to get all the parameters we need like RSI, Bollinger Bangs, SMA, etc

<table>
<thead>
<tr>
<th>Indicator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBANDS</td>
<td>Bollinger Bands</td>
</tr>
<tr>
<td>DEMA</td>
<td>Double Exponential Moving Average</td>
</tr>
<tr>
<td>EMA</td>
<td>Exponential Moving Average</td>
</tr>
<tr>
<td>HT_TRENDLINE</td>
<td>Hilbert Transform - Instantaneous Trendline</td>
</tr>
<tr>
<td>KAMA</td>
<td>Kaufman Adaptive Moving Average</td>
</tr>
<tr>
<td>MA</td>
<td>Moving average</td>
</tr>
<tr>
<td>MAMA</td>
<td>MESA Adaptive Moving Average</td>
</tr>
<tr>
<td>MAVP</td>
<td>Moving average with variable period</td>
</tr>
<tr>
<td>MIDPOINT</td>
<td>MidPoint over period</td>
</tr>
<tr>
<td>MIDPRICE</td>
<td>Midpoint Price over period</td>
</tr>
<tr>
<td>SAR</td>
<td>Parabolic SAR</td>
</tr>
<tr>
<td>SAREXT</td>
<td>Parabolic SAR - Extended</td>
</tr>
<tr>
<td>SMA</td>
<td>Simple Moving Average</td>
</tr>
<tr>
<td>T3</td>
<td>Triple Exponential Moving Average (T3)</td>
</tr>
<tr>
<td>TEMA</td>
<td>Triple Exponential Moving Average</td>
</tr>
<tr>
<td>TRIMA</td>
<td>Triangular Moving Average</td>
</tr>
<tr>
<td>WMA</td>
<td>Weighted Moving Average</td>
</tr>
</tbody>
</table>

The code snippet to obtain the indicator values are:

```python
close = price['Adj Close'].values
up, mid, low = talib.BBANDS(close, timeperiod=20, nbdevup=2, nbdevdn=2, matype=0)
rsi = talib.RSI(close, timeperiod=14)
sma = talib.SMA(close)
```

Various such indicators are combined in a hybrid model to run the simulation on the stock data for a single company to calculate the difference of final fund and initial fund. The loss making companies are put in one set and the profitable ones. The profits and losses of each company are considered to measure overall profit or loss and gain insights on our strategy.

### RESULTS

The implementations of all these modules which are integrated together gives us a one stop platform for all the
The results of the investor. We have provided a two-way communication platform where users can interact with the system and directly query the database to obtain the results. This platform has also increased the transparency into the stock analysts by calculating the prediction success ratio. Preliminary analysis gives a user an easy explanation of the trend in the stock market with the appropriate code words. The user can also view the entire financial records of the companies for the past 3 years and also go through the historic stock prices of about 5275 companies in the Indian stock market.

The user also has access to the pre-processed database with all the gainers and losers of all time frames from daily, weekly, monthly, quarterly, half yearly and yearly.

The model created by combining various parameters as mentioned in this strategy when run on a range of companies gives us an estimate of profit to loss ratio. The profitability which is measured by the difference in the final funds to the initial funds gives us data representing about 2/3rd of profitable companies and 1/3rd of loss making companies. When cumulative is considered the strategy is manipulated again and applied to specific top 100 companies to give us a higher number of profit making companies and hence the strategy can be applied by the user to trade automatically.

The testing results are:

![Figure 6: Trade simulation for 'Bajaj Finserv Ltd' (Loss Rs 80 in 8 trades) and '3M India Ltd' (Profit Rs 662 in 4 trades)](image)

**Figure 7: Vwap vs stock price trading**
- x-axis: days
- y-axis: price in Rs
- black line: Vwap
- blue line: live stock price

**Figure 8: Bollinger bands vs stock price.**

**Figure 9: Vwap vs stock price.**

**Figure 10: RSI (Relative Strength Index 0-100)**

### 6. CONCLUSIONS

The proposed framework has generated greater cumulative overall profits when traded with multiple companies using the proposed algorithms. The final funds after simulation gives us a profit margin of greater than 7 percent with respect to initial funds of Rs 10,00,000.

The importance of stock market analysis through the use of computational power will give us better insights into the financial world.
Having a user-friendly interface is one of the important key aspects. We are aiming at providing one platform for all the queries, a new user could have while going through the stock market. In this project, our main aim is to provide a one stop solution for the users of the stock market of India or any country in the world. The more the people invest the more the country progresses.

Users can make a handsome amount of return on investing in stocks or trading in the stock market. All they need is some patience and have to delve into and adapt to algorithmic trading so that they can gain some experience and invest smartly.

REFERENCES

[1] Rajesh V. Argiddi 1, Dr.Mrs.S.S.Apte2 , Bhagyashri U. Kale3"An Analysis on Stock Market Intelligence and Research Approaches",2014 Walchand Institute of Technology, Solapur, India


[14] Price Prediction of Share Market using Artificial Neural Network (ANN),Zabir Haider Khan Department of CSE, SUST, Sylhet, Bangladesh.


[22] The Trend Analysis of China’s Stock Market Based on Fractal Method and BP Neural Network Model WU Bing-hui HE Jian-min School of Economics and Management, Southeast University, Nanjing 211189, P.R.China. B. Chauhan, U. Bidave, A. Gangathade, S. Kale, Stock Market Prediction Using Artificial Neural Networks.


