A machine learning based stock trading framework using technical and economic analysis

Smarth Behl (smarth), Kiran Tondehal (kirantl), Naveed Zaman (naveedz)

Abstract

The goal of this project is to use a variety of machine learning models to make predictions regarding the stock price movements. Using technical analysis and economic analysis, leveraging various technical and economic indicators, the objective is to identify and optimize the buy and sell triggers to maximize trading profits.

I. INTRODUCTION

Predicting the direction of stock prices is a widely studied subject in many fields including trading, finance, statistics and computer science. Investors in the stock market can maximize their profit by buying or selling their investment if they can determine when to enter and exit a position. Professional traders typically use fundamental and/or technical analysis to analyze stocks in making investment decisions. Fundamental analysis involves a study of company fundamentals such as revenues and profits, market position, growth rates, etc. Technical analysis, on the other hand, is based on the study of historical price fluctuations. Due to the nature of market forces, economies tend to follow a pattern of expansion and contraction, over long periods of time. The stocks trade within an overarching environment where economy moves from one phase of the business cycle to the next.

Compared to the existing work, this project analyzes the stocks trading decisions utilizing the technical behavior of the trading patterns within the context of the fluctuating economic and business environment.

The objective function is to maximize medium to longer term profits based on S&P500 stock market index. The inputs are the technical indicators data and the economic indicators data. Three models (neural network, softmax logistic regression, decision forest) are then used to predict the buy/sell decisions.

II. RELATED WORK

Many techniques to predict the stock market have been developed using the various existing computational techniques, such as LR (Linear Regression) [4], ANN (Artificial Neural Networks) [5], Random Forest [1], SVM (Support Vector Machines) [4], CBR (Case-Based Reasoning) [3]. ANN are non-linear computational models, capable of representation based on market behavior data, without previous knowledge of the relationships among input and output variables. ANN have been widely used and have resulted in satisfactory performance compared to other approaches, which may be seen in the literature [5].

Stock forecasting involves complex interactions between market-influencing factors and unknown random processes. In [3], an integrated system by combining dynamic time windows, case based reasoning, and neural network for stock trading prediction is developed. It includes three different stages that involves screening out potential stocks and their important influential factors; using back propagation network to predict the buy/sell points of stock price and adopting case based dynamic window to further improve the forecasting results from the back-propagation network. The system developed in this research was a motivation for us to predict the sell/buy decision points in our project instead of stock price itself. Other neural network related work in [7] details the clustering power of stock prices and their ability to be utilized in predictive models using temporal recognition and brute force search to make the prediction. In [8], a methodology using trained neural networks and a generic algorithm was developed to determine buying and selling points of products traded in the stock exchange. A set of financial series were used where the price was compared to the returns obtained, by means of random data analysis from each one of the series. The results showed that financial time series are not entirely random.

III. DATASET AND FEATURES

For technical analysis, the focus has been on the most prominent indicators that can be efficiently operationalized and are intuitive in interpretation, including: Moving average convergence & divergence; Stochastic KD; Relative strength index; Larry William's R %; Daily closing volume. For economic analysis, indicators being utilized in terms of their importance and data availability are: Gross Domestic Product; Consumer price Index; Producer Price Index; Employment Index; Fed Funds Rate.

Technical Indicators have been calculated from the downloaded daily closing prices & volume data. The closing prices & volume have been smoothened using Welles Wilder Smoothing without any look-ahead bias, and relative 15-day change is calculated to serve as the price & volume trend indicators. Economic indicators have been extracted from the officially released historical percentage change data. In addition, Fed Funds rate have been smoothened using Welles Wilder Smoothing and relative 15-day change has been applied to capture the trends in the economic cycle.

The historical data for all the above technical and economic indicators since 1990 to today has been compiled and pre-processed. As stated in the project proposal earlier, the stock price prediction model is based on the S&P500 index to keep the scope of this project within manageable schedule. The historical S&P500 trading data since 1990 has also been gathered and processed. A snapshot of the dataset used is shown in the appendix.

Significant time and effort was involved in understanding the different indicators and how they need to be processed to be used as indicators effectively. For example, the raw fed funds rate in themselves are not a good indicator of whether the economy is expanding or contracting. These needed to be looked at relative to their trend over the past weeks and months. To address that, Welles Wilder smoothing is applied to the fed funds rate and the slope of the smoothened data, that represents both the magnitude and direction of the change in fed funds rate, is used as models' input indicator.

IV. METHODS AND MODELS

Several supervised learning methods were considered before deciding on the three models used for the project.

1. Neural Network: The Neural Network architecture consists of 15 hidden nodes. Sigmoid activation is used for the hidden nodes activation, softmax activation is used for the 4-class output layer.

$$z = w^T x + b$$

$$a = \frac{1}{1 + e^{-z}}$$

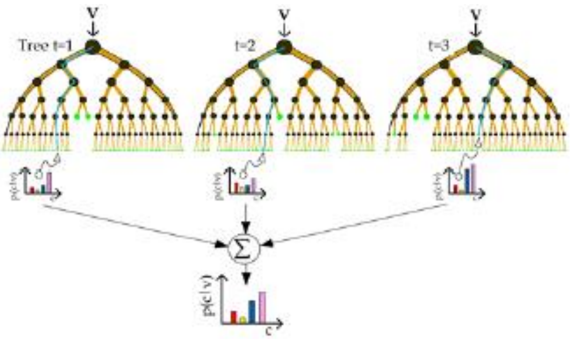
$$\hat{y} = \frac{\exp(z)}{\Sigma_k \exp(z)}$$

2. Softmax Logistic Regression: A 4-class logistic regression model is used. The classes represent strong buy, buy, strong sell and sell labels.

$$p(y = \mathrm{i} | x; \theta) = \frac{e^{\theta_i^T x}}{\sum_j^k e^{\theta_j^T x}}$$

3. Decision Forest: It is one of the models quite commonly used in a lot of

reviewed literature corresponding to stock trading predictions. The intent has been to analyze its performance compared to the other two models used.



For the ground truth labels, the 15-day future returns, calculated as a percentage change on the Welles Wider smoothened closing prices, are used. The historical economic cycles are analyzed and identified to classify labels as strong-buy or sell in an economic uptrend, and as strong-sell or buy in an economic slowdown. A profit/Loss calculation algorithm has been created and is used to automate the evaluation of the target labels performance. It is also used to compute the performance of the dev and test data.

The performance of the models is measured relative to the S&P500 index over the time-period being analyzed.

V. EXPERIMENTS & RESULTS

For trading, the metric is profitability. The models' predictions for the buy and sell classes are used to generate medium/long term trading signals on the dev and test data. The resulting profit (or Loss) multipliers are tabulated in comparison to the S&P500 index performance over the same period.

Experiments:

As explained above we conducted experiments with the goal to maximize the profit over the test period. We conducted various experiments using different indicators and built up the indicators set by adding one indicator at a time to analyze the corresponding effect. Different modeling strategies were employed, Decision forests and Neural networks with Wells Wilder smoothing gave better results. Started off with a target of getting better results for train data first to solve bias problem and then tuned the models to avoid overfitting. Some important experiments done to reach this conclusion using daily data of closing price from 1990-2017 are described next.

We initially decided to do the experiments with 2 labels as Buy and Sell. We made a 'Buy' decision if next day price was higher than the next day's price and Sell otherwise. This did not give us good profit because decisions were hard to predict and given the uncertainty of stock market, it is difficult to find the feature set which captures it correctly. Also, all misclassifications are not equal as some decisions lead to more loss than others.

The labels were then refined to 5 labels - 'Buy', 'Sell', 'StrongBuy', 'StrongSell', 'Hold'. Instead of defining labels based on next day's selling price, we defined labels on 15 days sliding window price, (which means seeing price 15 days ahead while defining the label) and based on the market trend. The market trend was captured well by the economic indicators in our feature set. This approach gave us much better results. But we were encountering a class imbalance problem as a lot of our labels were classified as 'Hold'.

We discovered that it was better to define labels based on four classes without the 'Hold' class. We used this to define our baseline with multi class logistic regression model which gave us a profit multiplier of 7.47 (results available in table) on training data.

We used neural network and decision forest model as the next steps. We experimented with 15, 30 ,50 and 100 nodes in hidden layer of neural network and got better results with 15 hidden nodes, it is closer to number of features we have. We used Gaussian normalization on data which performed better than min-max normalization as stocks data is closer to normal distribution.

Before going into workflow of our evaluation it is important to introduce the metric we used to judge performance of the models.

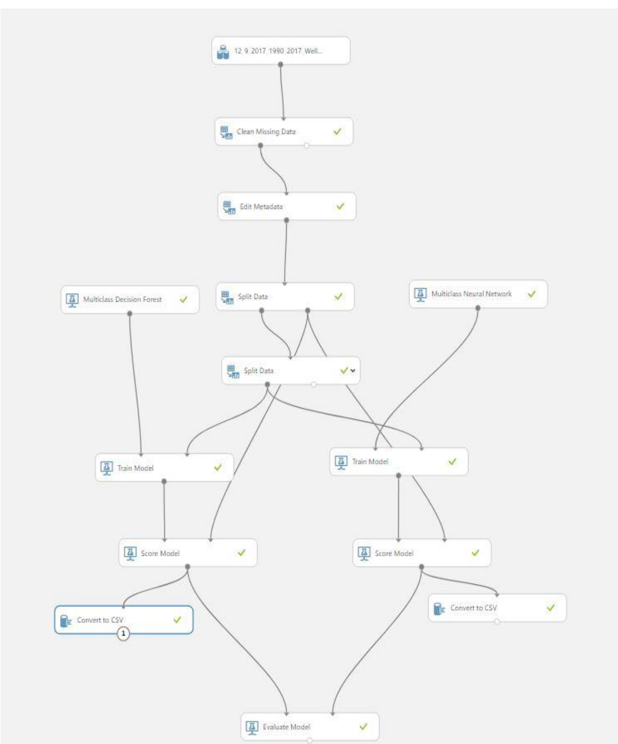
Profit Multiplier Metric:

We defined a single number target approach to judge performance of the model. We coded an application which takes our scored labels as input and gave a profit multiplier. For example, a profit multiplier of 1.2 over a period of one year will mean a profit of 20%. Our ground truth labels when given as input to the Profit Calculator for entire dataset (1990-2017), gave us a profit multiplier of 277. This gave us the motivation that a reasonably well performing model should be able to beat the S&P 500 index returns.

Tools Used:

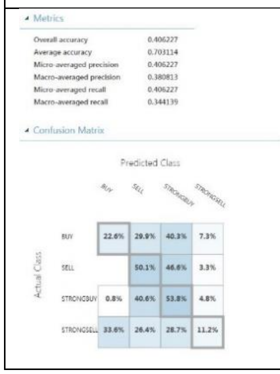
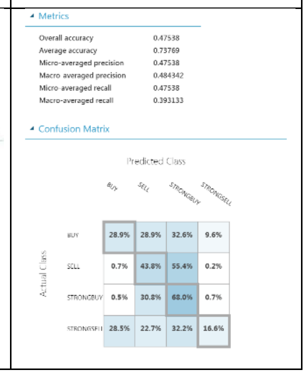
We used Microsoft's Azure Machine Learning Studio for our experiments. A workflow explaining comparison of neural network and decision forest is described in the diagram.

This allowed quick experimental runs on pre-defined models and experimental iterations using different variations (e.g. type of regularizations, learning rates, etc.). It enabled efficient results analysis, debug and learning.



Results:

The classification labels from above experiments with accuracies and confusion matrices are described in the table below, Decision Forest (left) and neural network (right).

Decision Forest	Neural Network
	

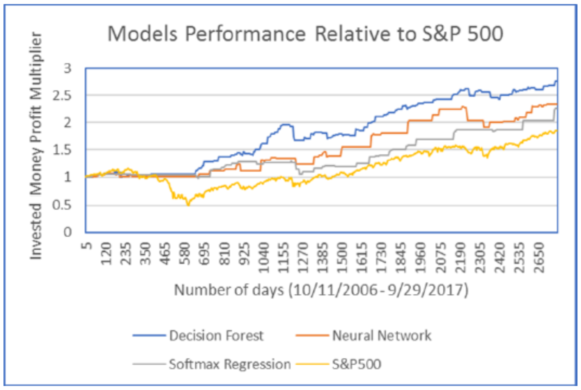
Based on the scored labels from different models, two trading decision strategies are used.

In the first strategy, the trading is triggered for all the four classes. In the second strategy, trading is triggered for the three classes - Strong Buy, Strong Sell and Sell.

Model And Decision Strategy	Train Profit Multiplier	Dev Profit Multiplier	Test Profit Multiplier
Strong Buy, Strong Sell, Buy, Sell			
SoftMax Regression	7.47	6.22	1.55
Neural Network	44.52	44.35	1.9
Decision Forest	105.21	65.03	2.15
Strong Buy, Strong Sell, Sell			
SoftMax Regression	5.98	6.35	2.26
Neural Network	18.24	19.95	2.44
Decision Forest	40.5	24.79	2.77
S&P 500	3.95	3.82	1.86

The table above and the graph below show the results on the test period from Oct. 2006 to Sept. 2017. Training and Dev used data from Jan. 1990 to Sept. 2006. The 80%-20% split was used between Training and Dev data.

The results look quite promising over the analyzed period. As noticed in the above table, models perform especially well during the recession of 2008 which confirms the idea of using economic indicators in ML for analyzing stocks. This can be extended to the time frame encompassing all the available historical data, to further analyze the models' predictive quality.



After getting decent results on a longer period, we decided to run the model for a shorter period. We choose test data from Jan 2015 to September 2017. The profit multiplier results for better performing models, neural network and decision forest models on this shorter-term test data are described in the following table. The models in the shorter-term beat S&P 500 for strategy 1 and Neural network beats slightly for strategy 2. Also, since the market is in an uptrend during the test period we are not significantly

Model And Decision Strategy	Test Profit Multiplier
Strong Buy, Strong Sell, Buy, Sell	
Neural Network	1.33
Decision Forest	1.28
Strong Buy, Strong Sell, Sell	
Neural Network	1.26
Decision Forest	1.17
S&P 500	1.25

beating the market. Economic indicators seem to play a more important role during recession / down trend. This also validated the idea of introducing economic indicators to avoid steep losses during an economic downturn.

Decision forest performs well on training and dev data as compared to neural network but results on test data are comparable. Decision forest model is possibly overfitting the training and dev data and optimally choosing number of decision trees and depth might give better results. A random forest model which is an extension of decision forest can be used for future work.

VI. DISCUSSION

Predicting stocks is a tough problem and is fun to work with because of its challenging nature. The economic indicators helped in identifying the beginning and end of recession periods. Different labeling techniques were tried, starting with two labels - buy and sell - on a daily basis. Such labels become hard for model to understand and correlate based on the feature set used. The data was then labelled based on prices and the overarching business cycle trends. As the objective function has been to optimize stock returns, hence not all mistakes made by the model are equal. One misclassification can be costlier than another misclassification, Paint the target approach was applied in implementing a profit calculator to understand the model's performance in terms of returns target at the end of each run.

VI. FUTURE WORK

The data set can be extended to start from 1900, to increase the train, dev, test data. The k-fold technique can be used to walk the test data over a broad range of periods to validate the robustness of the models' prediction. Translating the objective function of maximizing the S&P500 profits into a cost function that is directly applied to the models should further improve the targeted results. Additional optimization can be done by improving the cost function that penalizes the more severe misclassifications, and by using random forest for feature selection. This predictive framework can be extended to other stock market indexes, ETF's and individual stocks.

CONTRIBUTIONS

It has been a joint teamwork, the team members have contributed to all aspects of the project, including the overall strategy; algorithm; objective function definition; model selection & refinement; features selection; model creation, debug & execution; performance analysis. The deliberate intent has been for everyone to get hands-on exposure to all areas to maximize learning.

REFERENCES

[1] Khaidem, L., Saha, S. and Keevil, S. (2016). Predicting the direction of stock market prices using random forest. arXiv:1605.00003v1 [math.FA]

[2] Armstrong, P. and Keevil, S. (2016). A hybrid stock trading framework integrating technical analysis with machine learning techniques. The Journal of Finance and Data Science,4(1), 42-57

[3] Chang, P et al. (2009). A neural network with a case based dynamic window for stock trading prediction. Expert Systems with Applications, ISSN: 0957-4174, 36(3), 6889-6898

[4] Luo, L., Chen, X. (2013). Integrating piecewise linear representation and weighted support vector machine for stock trading signal prediction. Applied Soft Computing, ISSN: 1568-4946, 13(2), 806-816

[5] Oliveira, F., Nobre, C., and Zarate, L. (2013). Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index – Case study of PETR4, Petrobras, BrazilExpert Systems with Applications, ISSN: 0957-4174, 40(18), 7596-7606

[6] Ferreira, T., Vasconcelos, G.C, Adeodato, P. (2005). A new evolutionary method for time series forecasting, ACM proceedings of genetic evolutionary computation conference-GECCO, ACM, Washington, DC , 2221-2222

[7] Nayak, R., Braak, P. (2007). Temporal pattern matching for the prediction of stock prices, Proceedings of the second international workshop on Integrating artificial intelligence and data mining, Australian Computer Society, Inc, Darlinghurst, Austrália, 95-103

[8] A. Skabar, I. Cloete. (2002) Neural networks, financial trading and the efficient markets hypothesis. ACSC '02: Proceedings of the twenty-fifth Australasian conference on computer science, Australian Computer Society, Inc., Darlinghurst, Austrália, 241-249

[9] Introduction to Welles Wilder Smoothing. Available at: https://www.tradingtechnologies.com/help/x-study/technical-indicator-definitions/welles-wilders-smoothing-average-wws/ [Accessed November,2017].

[10] Economic Indicator Data. Available at: https://fred.stlouisfed.org/series/ [Accessed November,2017].

[11] Economic Research, Federal Reserve Bank of St Louis. U.S Regional and National Economic data (1940-2017). [Data file]. Available from https://fred.stlouisfed.org/series/

[12] Yahoo Finance, U.S S&P 500 Index Historical Data. [Data file]. Available from https://ca.finance.yahoo.com/quote/%5EGSPC/history/

[13] Investopedia, by IAC Publishing. Technical Indicator descriptions. Available from https://www.investopedia.com/terms/w/williamsr.asp?lgl=rira-layout-cpa-bsln

[14] Microsoft Azure Machine Learning Studio. Introduction to ML Studio. Available at: https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio [Accessed October 15,2017].

[15] Microsoft Azure Machine Learning Models. Neural Network. Available at: https://msdn.microsoft.com/library/azure/e8b401fb-230a-4b21-bd11-d1fda0d57c1f [Accessed October 15,2017].

[16] Microsoft Azure Machine Learning Models. Decision Forest. Available at: https://msdn.microsoft.com/library/azure/e8b401fb-230a-4b21-bd11-d1fda0d57c1f [Accessed October 15,2017].

[17] Microsoft Azure Machine Learning Models. Multiclass logistic. Available at: https://msdn.microsoft.com/en-us/library/azure/dn905853.aspx [Accessed October 15,2017].

APPENDIX (Data Snapshot)

	ECONOMIC INDICATORS					TECHNICAL INDICATORS								OUTPUT
DATE	Daily_Fed_Rate	relChange_afterSmoothing_15days_fedrate	CPI_%Change	GDP_%Change	PPI_%Change	Delta_Volume	RSI	MACD	Williams %R	Stochastic %K	Stochastic %D	EMPLOYEE_RATE	Change_AfterSmoothing_15days	DECISION_SLIDING15days
5/8/1990	8.2	-0.298708452	0.2	1.42306	0.43821	-2.135252457	52.1768559	-0.225665	-0.1400772	99.8599228	77.6805408	79.9	-0.153387852	STRONGBUY
5/9/1990	8.18	-0.32412259	0.2	1.42306	0.43821	3.218382809	58.5636939	0.274089	-1.4360465	98.5639535	90.3682049	79.9	-0.121980557	STRONGBUY
5/10/1990	8.23	-0.320085124	0.2	1.42306	0.43821	7.289790677	66.8343312	0.739093	-6.7363759	93.2636241	97.2291668	79.9	-0.067880824	STRONGBUY
5/11/1990	8.24	-0.316120066	0.2	1.42306	0.43821	56.63190582	84.9748919	1.747524	-1.2627216	98.7372784	96.854952	79.9	0.057924495	STRONGBUY
5/14/1990	8.27	-0.295344531	0.2	1.42306	0.43821	49.3373077	88.0974085	2.737065	-11.941288	88.0587122	93.3532049	79.9	0.201749389	STRONGBUY
5/15/1990	8.34	-0.258055979	0.2	1.42306	0.43821	9.583714611	86.1083384	3.443661	-13.474733	86.5252665	91.1070857	79.9	0.330045163	STRONGBUY
5/16/1990	7.53	-0.417344869	0.2	1.42306	0.43821	5.54962352	84.900591	3.935682	-14.38827	85.6117297	86.7319028	79.9	0.44885982	STRONGBUY
5/17/1990	8.2	-0.404252201	0.2	1.42306	0.43821	8.633801344	97.2077455	4.31381	-12.854825	87.1451753	86.4273905	79.9	0.590935588	STRONGBUY
5/18/1990	8.22	-0.398624161	0.2	1.42306	0.43821	6.997349485	97.0402557	4.574467	-13.654425	86.3455747	86.3674932	79.9	0.721169921	STRONGBUY
5/21/1990	8.24	-0.3593485	0.2	1.42306	0.43821	9.265788332	97.2477064	4.994588	-3.9747641	96.025236	89.838662	79.9	0.860175947	STRONGBUY
5/22/1990	8.22	-0.233908535	0.2	1.42306	0.43821	32.732567	97.0530431	5.301125	-7.9523899	92.0476101	91.4728069	79.9	0.985616279	STRONGBUY
5/23/1990	8.24	-0.217151458	0.2	1.42306	0.43821	12.20472158	97.0261699	5.549483	-4.7769113	95.2230887	94.4319783	79.9	1.107129053	STRONGBUY
5/24/1990	8.29	-0.193444731	0.2	1.42306	0.43821	0.991801134	92.99826	5.610623	-9.3345002	90.6654998	92.6453996	79.9	1.203989121	STRONGBUY
5/25/1990	8.27	-0.179892734	0.2	1.42306	0.43821	-21.37907415	78.1337004	5.289058	-29.11961	70.8803896	85.5896594	79.9	1.263195745	STRONGBUY
5/29/1990	8.26	-0.161774914	0.2	1.42306	0.43821	-9.976734427	81.5290364	5.461063	0	100	87.1819631	79.9	1.348373796	SELL
5/30/1990	7.69	-0.277351327	0.2	1.42306	0.43821	29.92885661	81.1202846	5.550342	-7.1832914	92.8167086	87.8990328	79.9	1.427912647	SELL
5/31/1990	8.23	-0.27181764	0.2	1.42306	0.43821	7.717773769	80.7270994	5.586556	-5.585677	94.414323	95.7436772	79.9	1.502207455	SELL
6/1/1990	8.28	-0.256690252	0.8	1.42306	-0.26178	21.59266287	75.2716933	5.705223	-3.1113718	96.8886282	94.7065533	79.9	1.537075928	SELL
6/4/1990	8.28	-0.249119804	0.8	1.42306	-0.26178	13.29728452	76.8349037	6.071412	-2.8302652	97.1697348	96.157562	79.9	1.579766515	SELL
6/5/1990	8.27	-0.261050078	0.8	1.42306	-0.26178	28.17693294	75.9011059	6.228498	-12.715304	87.2846961	93.7810197	79.9	1.619842179	SELL
6/6/1990	8.22	-0.088749093	0.8	1.42306	-0.26178	5.16081687	71.6943254	6.146571	-23.49327	76.5067303	86.9870537	79.9	1.650777315	SELL
6/7/1990	8.25	-0.074841174	0.8	1.42306	-0.26178	2.750249798	66.3157536	5.86795	-34.624856	65.3751443	76.3888569	79.9	1.667504772	SELL
6/8/1990	8.26	-0.063634255	0.8	1.42306	-0.26178	-8.471455972	56.5921163	5.228598	-67.133387	32.8666133	58.249496	79.9	1.656612072	SELL
6/11/1990	8.26	-0.057504525	0.8	1.42306	-0.26178	-22.90748686	55.9645061	4.901031	-49.446736	50.5532645	49.5983407	79.9	1.643034945	SELL
6/12/1990	8.23	-0.053928338	0.8	1.42306	-0.26178	1.280377507	61.2940412	4.957085	-17.496545	82.5034551	55.3077776	79.9	1.654416686	SELL
6/13/1990	8.61	0.036998494	0.8	1.42306	-0.26178	2.397143022	57.9891448	4.836818	-26.832691	73.1673088	68.7413428	79.9	1.652449214	SELL
6/14/1990	8.29	0.036253784	0.8	1.42306	-0.26178	-12.29411973	56.1965012	4.527927	-40.663957	59.3360425	71.6689355	79.9	1.644024116	SELL
6/15/1990	8.29	0.040381496	0.8	1.42306	-0.26178	31.65558553	62.8509843	4.235117	-41.250811	58.7491886	63.7508466	79.9	1.658718258	SELL
6/18/1990	8.28	0.044427095	0.8	1.42306	-0.26178	-14.09056838	44.1767287	3.476418	-100	0	39.3617437	79.9	1.601481267	STRONGBUY
6/19/1990	8.25	0.179727707	0.8	1.42306	-0.26178	-12.9217933	46.4592863	2.969216	-81.825342	18.174658	25.6412822	79.9	1.553592039	SELL
6/20/1990	8.24	0.178562921	0.8	1.42306	-0.26178	-11.11370392	46.8685617	2.588255	-76.825305	23.1746953	13.7831178	79.9	1.508259565	STRONGBUY
6/21/1990	8.25	0.167677537	0.8	1.42306	-0.26178	-10.18358176	45.9790703	2.369573	-65.952334	34.0476663	25.1323399	79.9	1.460496884	STRONGBUY
6/22/1990	8.24	0.154582277	0.8	1.42306	-0.26178	11.58951777	32.5255618	1.769185	-99.109169	0.89083141	19.3710644	79.9	1.35926767	STRONGBUY
6/25/1990	8.3	0.15876585	0.8	1.42306	-0.26178	-13.69259475	30.4288465	1.029745	-97.39587	2.60413015	12.5142093	79.9	1.24664838	STRONGBUY
6/26/1990	8.33	0.182325573	0.8	1.42306	-0.26178	-8.145136909	31.6657411	0.418735	-98.638183	1.36181733	1.61892629	79.9	1.145027675	STRONGBUY
6/27/1990	8.39	0.21269023	0.8	1.42306	-0.26178	-4.676748866	39.0123875	0.18095	-75.623383	24.376617	9.44752148	79.9	1.074246351	STRONGBUY
6/28/1990	8.36	0.232719605	0.8	1.42306	-0.26178	-11.29913037	48.4348043	0.191221	-60.099733	39.9002667	21.879567	79.9	1.045573823	STRONGBUY
6/29/1990	8.32	0.242627042	0.8	1.42306	-0.26178	-5.081916104	44.3540569	0.228199	-57.668408	42.3315917	35.5361585	79.9	1.002660993	STRONGBUY