CMSC498O - Intermediate Report

Predicting the Directional Movement of Stock Prices

Albert Koy & Nadeem Malik

Introduction

For our final project, we will attempt to predict the directional movement of a stock price given historical stock data. We plan on leveraging historical stock data to compute technical indicators that are commonly used in technical analysis, which is the study of a stock's price action and trading volume for forecasting its future movement. Moreover, we plan on applying machine learning algorithms to our data in order to predict whether a stock will close higher or lower than it did on the previous day.

The inspiration for this project comes from the University of Maryland's Stock Market class (HONR348M), where students learn general technical analysis techniques and apply them to stock selection and trade timing. A plethora of technical analysis indicators and techniques are covered. On a basic chart (*Figure 1*), there are various indicators that a trader might utilize during stock selection and trade timing, such as daily opens, daily closes, daily highs, and daily lows, moving averages, Bollinger Bands, daily trading volume, stochastic indicators, and moving average convergence divergence (MACD) indicators. A trader will use these indicators to determine the general trend of a stock and whether the stock is likely to increase or decrease over time. More sophisticated technical analysis techniques utilize different time frames (i.e. weekly and monthly charts) and even pattern recognition (e.g. cup with handle formations, double bottom formations, etc), but to limit the scope of our project, we will stick to analyzing daily price action, daily volume, and daily technical indicators.

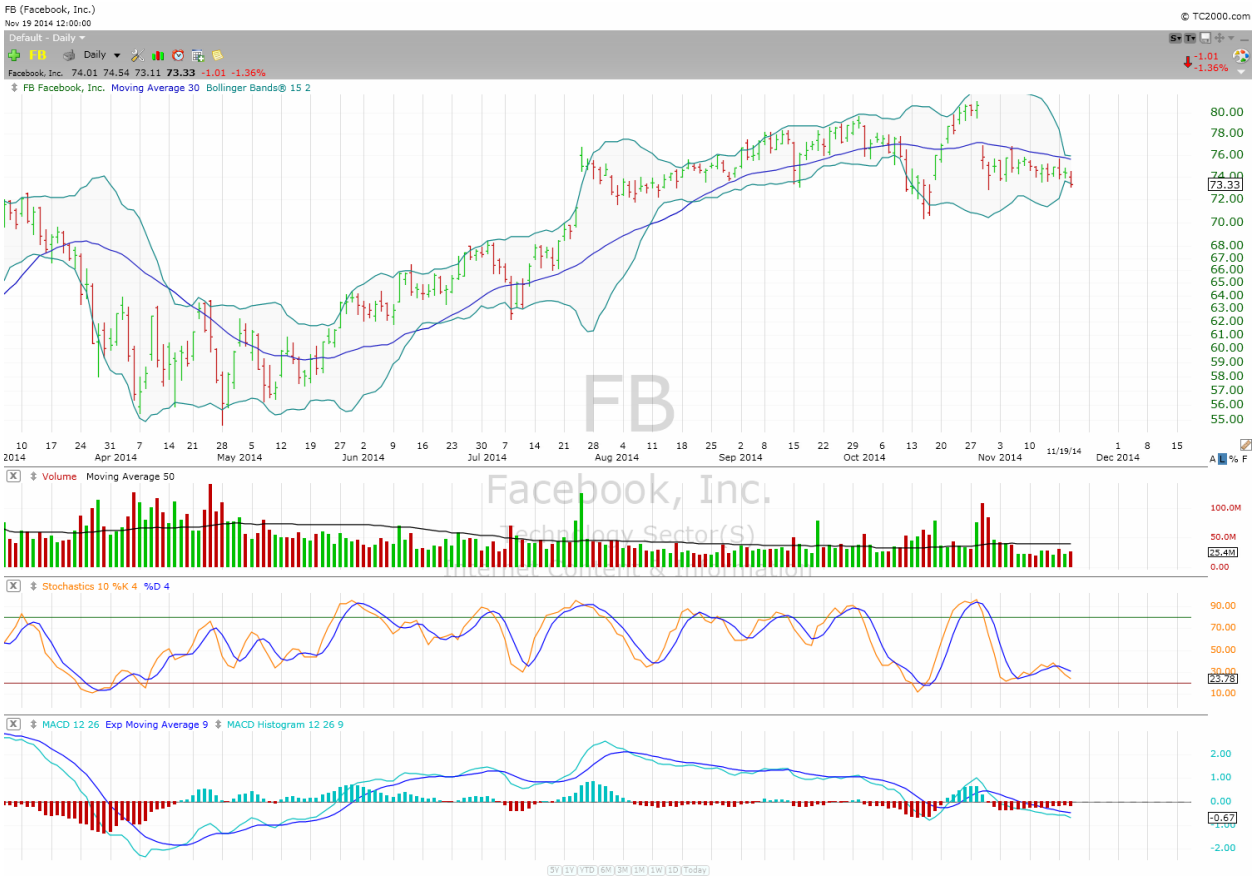

Figure 1

Data Sources and Data Wrangling

The first challenge in taking on this project was to find a source for historical stock market data. Unfortunately, we couldn't find many large, free, downloadable sets of historical stock market data on the Internet, so we had to write a web scraper to gather this data. We did this by writing a Ruby script to scrape the Yahoo! Finance page to grab data for all U.S. stocks on a daily timeframe over the past 5 years. We now have cleanly formatted CSV files for each U.S. Stock symbol (*Figure 2*) containing the Open, High, Low, Close, and Adjusted Close prices for each stock along with the daily volume (*Figure 3*).

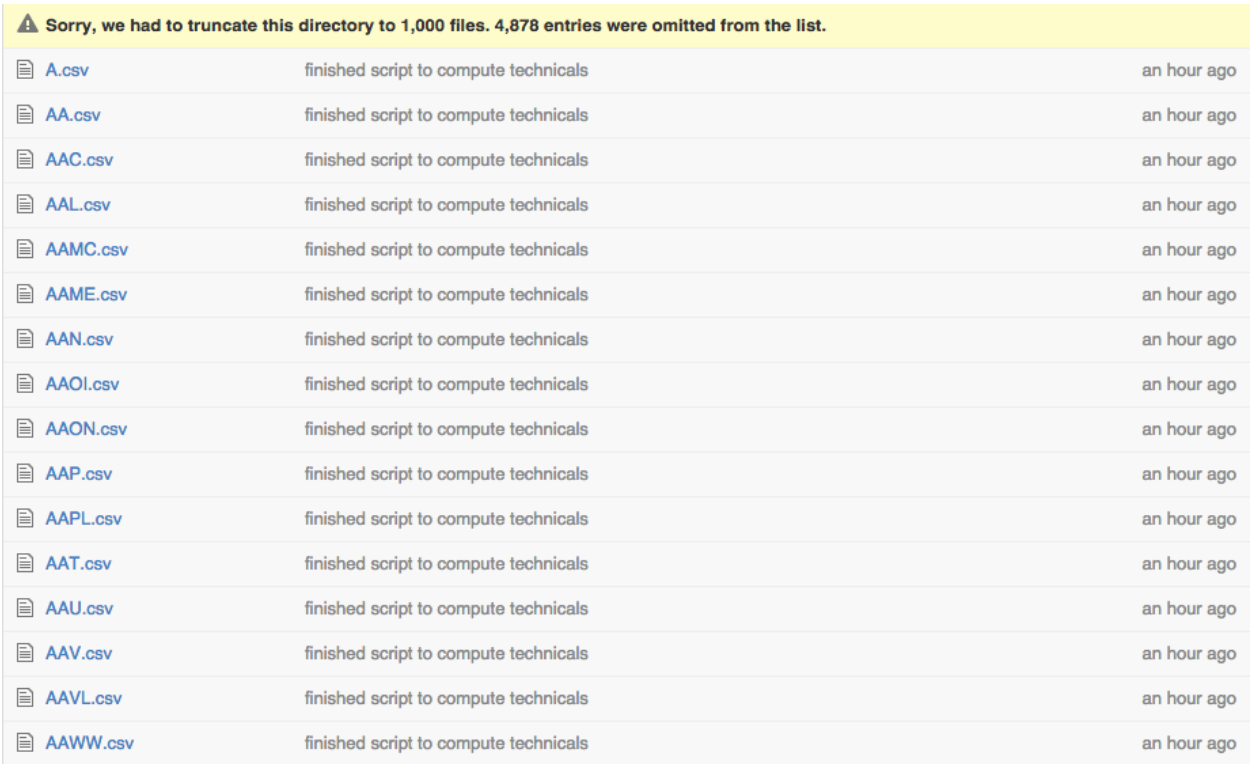

Figure 2

Date	Open	High	Low	Close	Volume	Adj Close
2014-11-18	40.25	41.16	40.05	40.80	5891400	40.80
2014-11-17	41.57	41.70	41.21	41.24	2696100	41.24
2014-11-14	41.36	41.65	41.29	41.57	1331300	41.57
2014-11-13	41.55	41.74	41.26	41.45	1805500	41.45
2014-11-12	41.30	41.54	41.02	41.45	2504300	41.45
2014-11-11	41.52	42.11	41.34	41.66	2163800	41.66
2014-11-10	40.98	41.55	40.88	41.53	2583900	41.53
2014-11-07	41.39	41.48	40.86	40.93	2458700	40.93
2014-11-06	40.39	41.41	40.26	41.37	2968300	41.37
2014-11-05	40.61	40.67	39.98	40.13	2341400	40.13
2014-11-04	40.26	40.83	39.84	40.18	5765200	40.18
2014-11-03	39.90	41.24	39.49	40.84	4559800	40.84

Figure 3

After retrieving and formatting the raw stock data, the next step was to compute technical analysis indicators for each stock. This meant writing a script to transform each of the 5000+ CSV files into a file that would give us some more information to work with for each stock. We used python to write this script and relied heavily on the NumPy, Pandas, and

TA-Lib libraries. TA-Lib is an industry standard library for performing technical analysis, with many functions to help compute indicators. To augment the raw data that we retrieved from Yahoo! Finance, we computed the 50 Day Average Volume, 30 Day, 4 Week, 10 Week, and 30 Week Simple Moving Averages, the Daily 20.2 Upper, Middle, and Lower Bollinger Bands, the Daily 12.26.9 MACD, 12.26.9 MACD Signal, and 12.26.9 MACD Histogram values, the Relative Strength Index with 14 bars as the timeframe, and the Daily 10.4 and 10.4.4 Stochastic values. For now, we believe that this should be enough data to work with since we have included every essential indicator that is used in most manual technical analysis.

	Date	Open	High	Low	Close	Volume	Adj Close	50DayAvgVol	30DaySMA
0	2014-11-18	40.25	41.16	40.05	40.8	5891400	40.8	3203320.0	48.618
1	2014-11-17	41.57	41.7	41.21	41.24	2696100	41.24	3121452.0	49.0913333333
2	2014-11-14	41.36	41.65	41.29	41.57	1331300	41.57	3105158.0	49.6023333333
3	2014-11-13	41.55	41.74	41.26	41.45	1805500	41.45	3103900.0	50.1136666667
4	2014-11-12	41.3	41.54	41.02	41.45	2504300	41.45	3089896.0	50.593
5	2014-11-11	41.52	42.11	41.34	41.66	2163800	41.66	3081134.0	51.0846666667
6	2014-11-10	40.98	41.55	40.88	41.53	2583900	41.53	3094214.0	51.5953333333
7	2014-11-07	41.39	41.48	40.86	40.93	2458700	40.93	3097894.0	52.116
8	2014-11-06	40.39	41.41	40.26	41.37	2968300	41.37	3087736.0	52.635
9	2014-11-05	40.61	40.67	39.98	40.13	2341400	40.13	3061726.0	53.1523333333
10	2014-11-04	40.26	40.83	39.84	40.18	5765200	40.18	3039650.0	53.7466666667

4WeekSMA	10WeekSMA	30WeekSMA	20.2UpperBB	20.2MiddleBB
46.3355	52.0862	55.1544666667	59.2177753813	46.3355
46.9955	52.4192	55.2403333333	60.0274292125	46.9955
47.551	52.7474	55.3194666667	60.5011288024	47.551
48.086	53.0744	55.3933333333	60.8867912255	48.086
48.599	53.3992	55.4688	61.1140691568	48.599
49.116	53.7314	55.5528	61.2557139999	49.116
49.6225	54.0596	55.6455333333	61.3125878953	49.6225
50.1855	54.3722	55.7330666667	61.3344541662	50.1855
50.821	54.6966	55.8225333333	61.2103019977	50.821
51.504	55.0168	55.9172	61.0826479213	51.504

20.2LowerBB	12.26.9MACD	12.26.9MACDSignal	12.26.9MACDHist	RSI14	10.4STOC	10.4.4STOC
33.4532246187	-3.54979768637	-3.3683180917	-0.181479594666	24.9545947239	47.9681330397	32.8918681589
33.9635707875	-3.61532868011	-3.32294819304	-0.292380487076	25.8518585235	41.4216093414	23.88742912
34.6008711976	-3.6911834586	-3.24985307127	-0.441330387331	26.5158331494	29.4107704553	16.4882139706
35.2852087745	-3.76791179829	-3.13952047444	-0.628391323851	25.8729693976	12.766959799	11.491048995
36.0839308432	-3.79882088933	-2.98242264347	-0.816398245859	25.8729693976	11.9503768844	10.1248429648
36.9762860001	-3.7830165526	-2.77832308201	-1.00469347059	26.2190770856	11.8247487437	8.92744974874
37.9324121047	-3.72920059478	-2.52714971436	-1.20205088042	25.6473340401	9.42211055276	13.0948268126
39.0365458338	-3.59303258177	-2.22663699426	-1.36639558751	23.0931934421	7.30213567839	23.2726019073
40.4316980023	-3.30587948035	-1.88503809738	-1.42084138297	23.6463252806	7.1608040201	39.6870879476
41.9253520787	-2.9360821839	-1.52982775164	-1.40625443227	18.5404400456	28.4942569993	61.8440795085

Figure 4

Finally, in terms of gathering data, the last task that we need to complete is normalizing the stock data. Depending on the machine learning algorithms that we end up using, normalization may take different forms. For instance, if we were using a decision tree algorithm or another binary classifier, we may decide to replace stock data with a binary value indicating the relation between the current day's data and the previous day's. In particular, if the closing price increased, we could normalize the value to 1; if it decreased, we could normalize it to -1. Another technique we may employ is to use the percentage change between days instead of the actual price values. These are simply some ideas to consider, but when we actually start training our models, normalization should add some consistency between our datasets for each stock, allowing us to produce a more robust predictive model.

Research and Findings

To gain an understanding of the current methods used in algorithmic technical analysis, we decided to look at past research and the methods of analysis used. One of the simplest techniques we found for stock prediction was using decision trees. This involves constructing a tree using the basic attributes of stock (e.g. min, max, opening and closing prices), and labeling each stock either "buy" or "sell." This would also involve discretization as the stock values for each attribute would need to be binned to a smaller set of classifications. Though the decision tree approach is very simple, we found it to be inaccurate based on prior research, where it was found to be only 42-54% for predicting stocks based on historical data from 3 companies obtained from Amman Stock Exchange (ASE) over a 2 year time period [1]. Though we opted to not use decision trees, we did learn from this research that it was based on the Cross Industry Standard Process for Data Mining (CRISP-DM) model, a model used by data science experts as it provides a blueprint for analysis. The CRISP-DM model has six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. We will apply this knowledge to our final project, allowing us to formulate a structured analysis.

Another research paper analyzed ten different data mining techniques relevant to technical analysis and found that each technique was quite accurate and had a hit rate of over 80% [2]. This analysis looked at data from Hong Kong's stock market using Hang Seng Index (HSI). In addition to the standard indicators (min, max, opening price, closing price), they used the USD to HKD exchange rate. The authors of this paper found that Support Vector Machines (SVM) and Least Squares Support Vector Machines (LS-SVM) were the best techniques for predicting the directional movement of stocks. LS-SVMs were preferred to SVMs when predicting outside the sample, with a hit rate of 86.4%. The worst prediction techniques were k-nearest neighbors and tree classifications. Support Vector Machines are supervised learning models used for classification. They allow for both linear and nonlinear classification, where the latter is done by what is known as a "kernel trick," transforming linear input into higher dimensions.

Table 4. Summary result of prediction performances by ten different approaches

	In-sample		Out-of-sample		
Predictors	Hit Rate	Error Rate	Hit Rate	Error Rate	Rank
LDA	0.8393	0.1607	0.8440	0.1560	6
QDA	0.8305	0.1695	0.8480	0.1520	5
K-NN	0.8312	0.1688	0.7960	0.2040	9
Naïve Bayes	0.8386	0.1614	0.8280	0.1720	7
Logit model	0.8474	0.1526	0.8560	0.1440	3
Tree classification	0.8717	0.1283	0.8000	0.2000	8
Neural network	0.8481	0.1519	0.8520	0.1480	4
Gaussian process	0.8595	0.1405	0.8520	0.1480	4
SVM	1.0000	0.0000	0.8600	0.1400	2
LS-SVM	0.8528	0.1472	0.8640	0.1360	1

Another popular technique for technical analysis is Artificial Neural Networks (ANN). One research paper on the subject found that using Neural Networks could allow for prediction with a best case accuracy of 96%, with an average case accuracy of 88% [4]. Neural networks, as the name suggests, are modeled after human brain, and function through a learning process with interconnected elements. The network functions through the transmission of signals between each element and each "neuron" or element of the network fires when a certain threshold has been reached. ANNs are useful because they have a parallel adaptive learning process that is very unique in the field of machine learning. Though Artificial Neural Networks seem to be one of the best techniques for technical analysis, we decided not to proceed with them due to the scope of our project and their relative complexity.

Finally, we found a project that is very similar to ours, where the team analyzed 94 stocks from S&P 500 over 90 days [3]. This project used a variety of models including Logistic Regression, Random Forests, and Blended Models. Logistic regression is another

classification model that uses probabilities to predict labeling outcomes by looking at the relationship between the independent and dependent variables, and has the benefit of not having to discretize the independent variables. Some important lessons from that project were that the magnitude of stock movement is important and may not even be predictable, and you have to keep transaction prices in mind, so you cannot be overly frequent with trading. The team was able to achieve high prediction accuracy with an area under the ROC curve (AUC) of over 90%. Another interesting finding was that the AUC varied by industry, and the healthcare industry was able to yield the greatest AUC.

We have decided to proceed with a variety of models, as some models work better than others depending on the category of stock. For our purposes, we will primarily focus on SVMs, Logistic Regression, and Random Forests as our research has found that these techniques have resulted in models with good prediction accuracy. In addition, we hope to explore the effect of applying Platt scaling on our models, which is a way of transforming the outputs of classification models into a probability distribution over classes.

Communicating Our Results

Once we have developed our predictive models, we need to choose the right medium through which to communicate our results. As of right now, we believe that an interactive visualization would be the best way to do this. There are several libraries that we have found that we think would help us with this. We plan on using d3, cubism [5], and cross-filter [6]. The envisioned product is one where the user may provide a stock ticker symbol, prompting our application to pull real-time stock data for that symbol and show how the predictions from our models change in real time in response to the changing stock data. As an add-on to this idea, we could even display a list of all U.S. stocks sorted by their probability of increasing the next day.

Resources:

1. http://www.acit2k.org/ACIT/2013Proceedings/163.pdf
2. http://www.ccsenet.org/journal/index.php/mas/article/view/4586/3925
3. https://sites.google.com/site/predictingstockmovement/
4. http://www.cs.berkeley.edu/~akar/EE671/report_stock.pdf
5. https://square.github.io/cubism/
6. http://square.github.io/crossfilter/