George Mason University

Spring 2013

CS 687 – Advanced AI

Professor: Dr. Harry Wechsler

Lawrence W. Stark

Project Report :

Predicting Stock Market Movement Using Artificial Intelligence Algorithms.

May 5th 2013

Introduction:

The purpose of this project was to investigate the utility of using historical stock data to predict future stock market movement. The first phase included modeling past behavior using the stocks from the Standard & Poor's (S&P) 500 and using several classifiers to determine if the next day's market close would be greater or less than the previous day's close value. During the first phase, market volatility revealed itself to be a significant factor in the ability of a classifier to correctly predict the next day's close value. Thus, phase two incorporated market volatility as defined by the Chicago Board Options Exchange (CBOE) Market Volatility Index, VIX. Regression analysis was performed to attempt to predict the next day's market volatility.

Being able to accurately predict this value could greatly increase the Return On Investment (ROI) from phase one. One possibility would be to avoid buying stocks on days when the market was predicted to be volatile thus decreasing the likelihood of taking the wrong action. On the other hand, if the model indicated strong movement (either up or down), the volatility index could be used to predict the amount of money at stake to be gained or lost. Thus, riskier trades may be a worthwhile gamble for the potential larger rewards.

Methodology:

The strategy for this project is founded in ideas dating back to the 1700's that originated from a Japanese rice trader known as Munehisa Homma. He was the first to define the candlestick pattern and investigate complex relationships between multiple candlesticks and how that correlates to the value of a commodity. Homma's ideas were rediscovered in the early twenty-first century by a financial analyst, Steve Nison, who applied the candlestick pattern to contemporary stock market analysis. Numerous publications from both finance and academia prove that the candlestick pattern analysis techniques are still used today even by some of the largest volume traders in the market.

For this project, the S&P 500 was used as a microcosm of the entire stock market. This was a deliberate choice due to the composition of the stocks that comprise the S&P 500 as well as the index Electronically Traded Fund (ETF), SPY. The spider ETF is an aggregate of the entirety of the S&P 500 and is used in both finance and academia as an index. Its sole purpose is to model the market as a whole. Thus, the 500 individual stocks could be used for training and testing while leaving the SPY values as sequestered data for validation purposes.

The selection of the S&P 500 lends itself well to comparison with other state-of-the-art research in stock market prediction. There are too many papers attempting the same goal as this project to list them all. Instead, the various algorithms and techniques being employed can be clustered into the following categories: Support Vector Machines; Neural Networks; Twitter; Autoregressive Integrated Moving Average; and Echostate Networks. There are more papers that fall outside of Artificial Intelligence and Machine Learning that would be clustered into additional categories. One theme that the majority of these papers have in common with this project is that they also used the S&P 500 and the SPY index.

Although its impossible to prove, it is likely that even the state-of-the-art algorithms in today's financial market incorporate the S&P 500. Due to the proprietary nature of their algorithms and trading strategies, big investment firms don't reveal specifics. However, it is possible to say with confidence that the S&P 500 in addition to whatever additional factors are included in the model are able to yield impressive real-world results. JP Morgan Chase recently published that for the entire 2013 calendar year, they had exactly 0 days of loss while trading. Somewhat lesser known investment conglomerate, Virtu, similarly revealed that from 2009 through 2013, they have had exactly 1 day of trading loss. While this project cannot hope to achieve the same level of success as a team of sophisticated researchers, those firms have established a goal to achieve and a degree of success to be measured against.

Data Sets:

Most of the data for this project came from TradeStation (http://www.tradestation.com) which is a trading platform and online financial brokerage. In order to entice stock traders to use their platform, TradeStation, provides a wealth of historical data to enable back-testing of investment strategies. It is this data that was downloaded and exported into .CSV format.

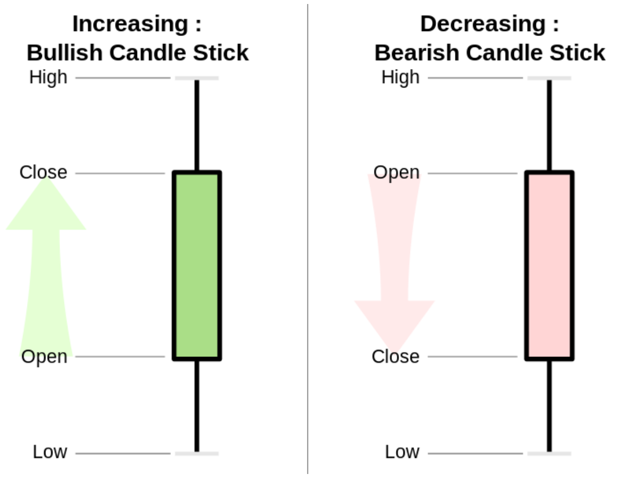
Details:

- Time span: January 1993 to October 2013 (~ 20 years)
- S&P 500 + SPY = 501 stocks
- Training & Test Set: ~ 2.2 million samples
- Validation Set: ~5,200 samples

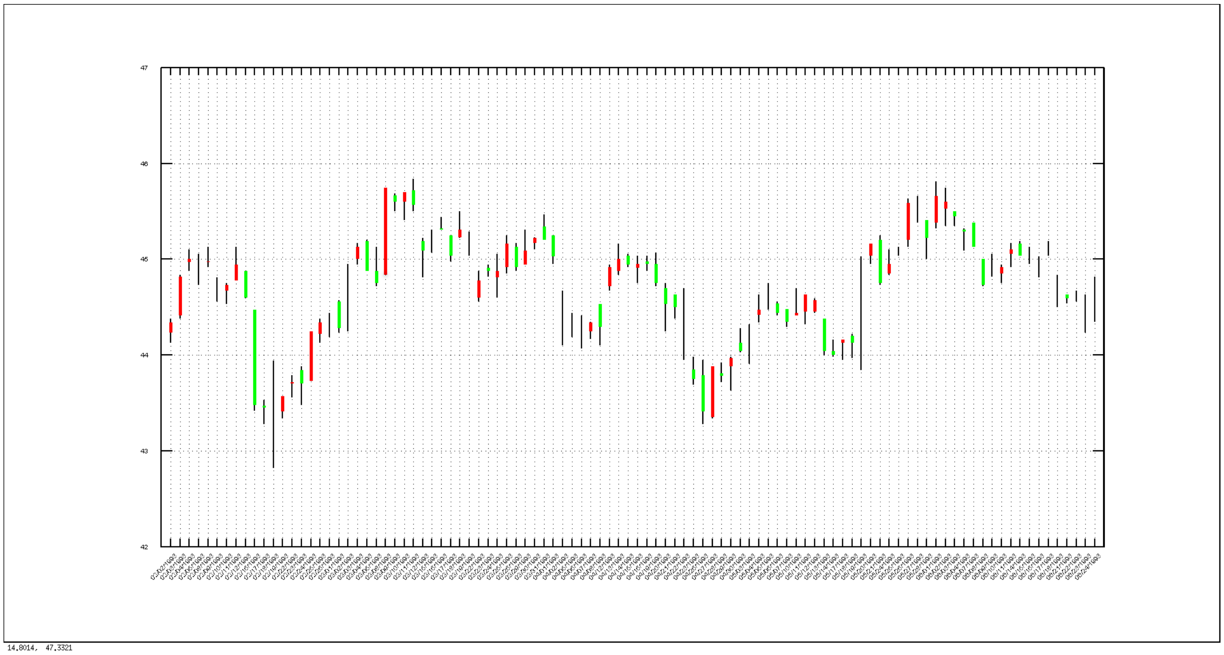
Candlestick Pattern Features:

Feature	**Day 1**	**Day 2**	**Day 3**
Open	Open 1	Open 2	Open 3
High	High 1	High 2	High 3
Low	Low 1	Low 2	Low 3
Close	Close 1	Close 2	Close 3
Close Change %	(C2 – C1) / C1	(C3 – C2) / C2	

Sample Candlesticks:



Subset of Candlesticks over time:



The remaining data, used in phase two, was obtained directly from the creator, CBOE (http://www.cboe.com/micro/vix/historical.aspx). A subset of the VIX data spanning the same time frame as above (January 1993 – October 2013) was selected. This corresponded to the same approximately 5,200 samples.

Tools:

The Machine Learning toolkit, WEKA (Waikato Environment for Knowledge Analysis) was used for this project. The workbench provides state-of-the-art implementations for numerous AI algorithms including classifiers like Naive Bayes and Radial Basis Functions; clustering like K-Means and preprocessors like Principal Component Analysis for attribute selection. Each of the algorithms implemented in WEKA provide several algorithm-specific customization that allow the performance of each to be tuned for optimal performance on the specific data being used.

Before the data could be processed by WEKA, it was first transformed by a custom Java program. This software was written specifically to read in the .CSV files from TradeStation and to output an .ARFF file that is compatible with WEKA. The transformation process also provided the necessary hook to format single-day values into the 3-day and N-day sliding window representation. In addition, the otherwise unlabeled data could at that point be associated with a class label for use by a classification algorithm within WEKA.

Procedure:

1. Create a Training and Test Set
   - Use S& P 500
   - Create 3/N Day sliding window representation

2. Create a Validation Set
    - Use SPY
    - Create 3/N Day sliding window representation

3. Format all data files into .arff format

4. Optionally Apply Principal Component Analysis for Attribute Selection
    - linear feature extraction (y = Wx):

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_n \end{bmatrix} \rightarrow \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m1} & w_{m2} & & w_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \\ x_n \end{bmatrix}$$

    - Mean Square Error:

$$\bar{\varepsilon}^2(M) = \sum_{i=M+1}^{N} \varphi_i^T \Sigma_x \varphi_i$$

where Σ_x is the covariance matrix of x

    - Minimize:

$$\frac{\partial}{\partial \varphi_i} \bar{\varepsilon}^2(M) \;=\; 2(\Sigma_x \varphi_i - \lambda_i \varphi_i) \;=\; 0 \;\Rightarrow\; \Sigma_x \varphi_i = \lambda_i \varphi_i$$

Note: where $\frac{d}{dx}(x^T A x) = (A + A^T)x \;=\; 2Ax$

to make the problem computationally solvable for high dimensions

5. Classify Using Naive Bayes
    - Likelihood Ratio Test:

$$\Lambda(x) \;=\; \frac{P(x \mid \omega_1)}{P(x \mid \omega_2)} \overset{\omega_1}{\underset{\omega_2}{\gtrless}} \frac{P(x \mid \omega_2)}{P(x \mid \omega_1)}$$

6. Classify using Radial Basis Function Network Classifier
    - Compute Mixture Model coefficients:

$$h(x)^{(i} \;=\; \sum_{k=1}^{N} w_k \varphi(\|x^{(i} - x^{(n}\|) \;=\; t^{(i} \;\Leftrightarrow\; w \;=\; \Phi^{-1} t$$

    - Activation of hidden unit is determined by distance between input x and prototype µ:

$$\varphi_j(x) \;=\; f(|x - \mu_j|)$$

    - Radial Basis with Gaussian kernels:

$$\varphi_j(x) \;=\; e^{[-\frac{1}{2}(x-\mu_j)' \sum^{-1} (x-\mu_j)]}$$

- Hidden-to-output mapping:

$$y_k = \sum_{j=1}^{N_H} W_{jk} \varphi_j (|x - \mu_j|) + w_{0k}$$

- Select Hidden-to-Output weights to minimize Mean Square Error at the output:

$$W = argmin_W \{ \sum_{n=1}^{N} \sum_{k=1}^{N_0} (t_k^{(n} - \sum_{j=1}^{N_H} w_{jk} \varphi_j (x^{(n)})) \} = argmin_W \|T - \Phi W\|$$

7. Classify Using Linear Regression:

$$y = X\beta + \varepsilon$$

8. Classify Using Support Vector Machines for Regression:

$$\tilde{L}(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j k(x_i, x_j)$$

subject to (for any i = 1,...,n): $0 \leq \alpha_i \leq C$

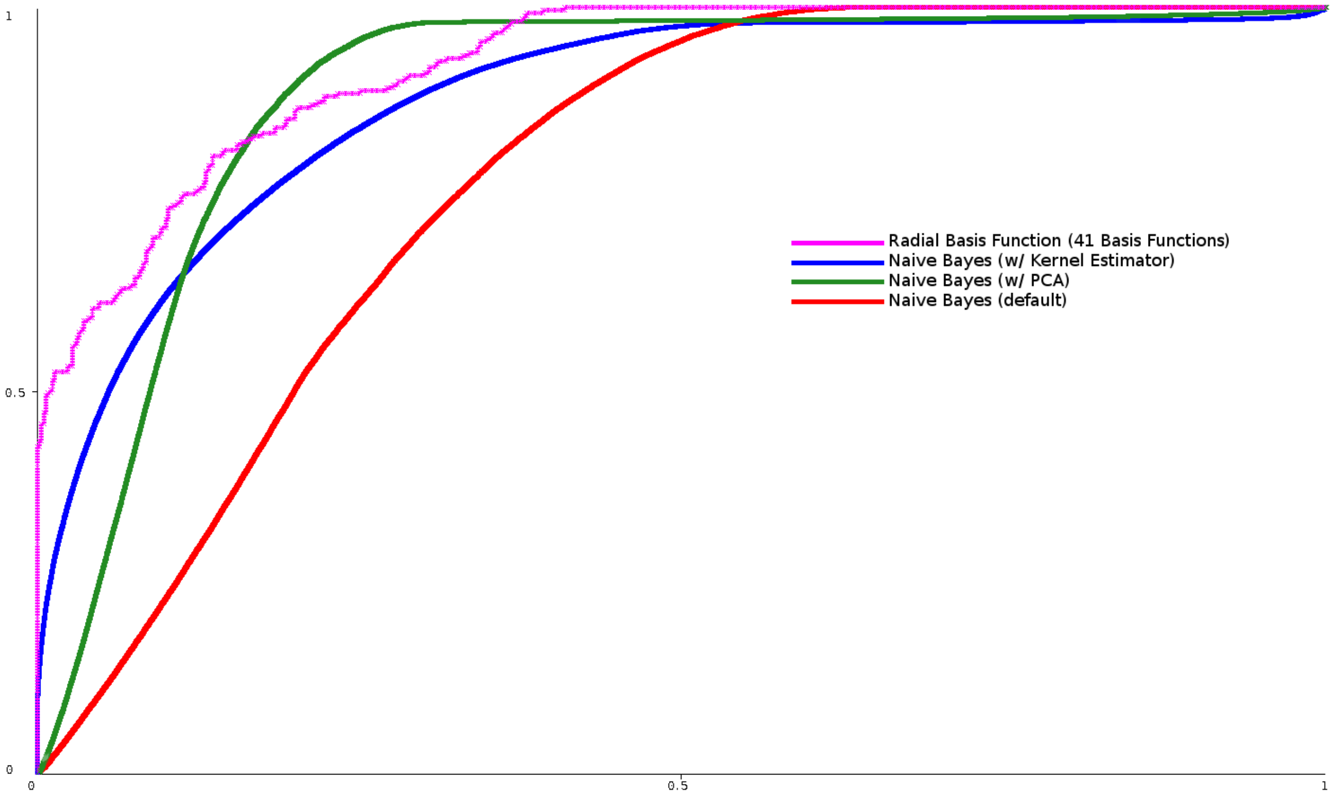
and $\sum_{i=1}^{n} \alpha_i y_i = 0$

9. Cluster Using K-Means

$$\underset{S}{arg\ min} \sum_{i=1}^{k} \sum_{x_j \in s_i} \|x_j - \mu_i\|^2$$

Results:

*Note: For all results shown in this section, the actual output from WEKA appearing here only demonstrates the highest rate of successful classification that was achieved using the specified algorithm. While all required some trial and error to empirically discover the best combination of settings for working with the sliding window stock data, it is correct to assume that each of the preceding trials yielded results some degree poorer than those reported here.



Comparison of ROC Curves for Algorithms Classifying Into Discrete Nominal Categories:

1. Classification Using Naive Bayes

a	b	c	d	e		<-- classified as
852	281	76	78	85359	|	a = UP
212	1287	23	31	77335	|	b = DOWN
283	188	1430	122	86450	|	c = TURN-UP
213	105	3	1725	83574	|	d = TURN-DOWN
1858	2050	1666	2000	415369	|	e = VOLATILE

Confusion Matrix

TP Rate	FT Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
0.01	0.004	0.249	0.01	0.019	0.029	0.889	0.39	UP
0.016	0.004	0.329	0.016	0.031	0.053	0.898	0.388	DOWN
0.016	0.003	0.447	0.016	0.031	0.067	0.901	0.44	TURN-UP
0.02	0.003	0.436	0.02	0.039	0.074	0.89	0.423	TURN-DOWN
0.982	0.98	0.555	0.982	0.709	0.009	0.489	0.543	VOLATILE
0.552	0.545	0.471	0.552	0.407	0.03	0.67	0.484	Weighted Avg.

WEKA Naive Bayes Classifier Output

2. Classification Using Radial Basis Function Network

a	b	c	d		<-- classified as
253	0	3	10	|	a = UP
1	86	34	32	|	b = DOWN
28	1	180	3	|	c = TURN-UP
44	1	4	164	|	d = TURN-DOWN

Recall	**F-Measure**	**MCC**	**ROC Area**	**PRC Area**	**Class**
0.951	0.855	0.787	0.924	0.867	UP
0.562	0.714	0.705	0.927	0.771	DOWN
0.849	0.831	0.774	0.935	0.854	TURN-UP
0.77	0.777	0.703	0.94	0.862	TURN-DOWN
0.824	0.809	0.804	0.748	0.931	0.845

3. Clustering Using K-Means

Cluster Centroids

		Cluster#							
Attribute	**Full Data**	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**
	(3430)	**(76)**	**(38)**	**(60)**	**(30)**	**(94)**	**(29)**	**(59)**	**(129)**
open1	108.0091	121.3847	49.0987	54.1657	131.3453	84.7559	110.0007	134.4715	129.5751
open2	108.0155	121.2842	49.1595	54.2117	131.6637	84.5069	109.611	134.3742	129.4409
open3	108.0525	121.3589	49.2274	54.2323	131.4927	84.2376	109.0876	134.4056	129.3381
open4	108.0625	121.9801	49.3382	54.294	131.5247	83.9812	109.2634	134.0947	129.35
open5	108.0758	122.4192	49.4213	54.372	131.4633	83.8104	108.5548	134.1007	129.4278
open6	108.1194	122.9189	49.4576	54.4233	131.1	83.638	108.4317	133.9193	129.4723
open7	108.1444	123.1014	49.4829	54.4637	130.1917	83.843	108.1755	133.7981	129.4247
open8	108.1657	123.4808	49.4682	54.478	129.3523	84.1988	107.3166	133.1059	129.5316
open9	108.1899	123.615	49.5263	54.559	128.684	84.4945	106.2128	132.7353	129.7052
open10	108.242	123.6518	49.6497	54.6282	128.209	84.8239	105.8883	132.168	129.9803
open11	108.2784	124.0079	49.6979	54.7027	127.0877	85.0726	105.5476	131.8237	130.1377
open12	108.295	124.2651	49.8171	54.7635	126.5993	85.2347	105.0062	131.7912	130.42
open13	108.3125	124.5686	49.9047	54.8407	126.0347	85.3374	104.4224	131.0824	130.7002
open14	108.3715	124.7504	49.9547	54.8848	125.69	85.4077	104.4172	131.0439	130.9358
open15	108.3997	124.5988	50.0221	54.9743	125.8817	85.5224	105.2538	130.5222	131.1134
open16	108.4164	124.663	50.1171	55.0523	125.1	85.6864	105.1228	130.6375	131.3557
high1	108.7288	122.2164	49.2645	54.3633	132.3273	85.7707	110.6314	135.3683	130.1884
high2	108.7418	122.0939	49.3105	54.4233	132.3517	85.5177	110.2383	135.2975	130.0485
high3	108.7683	122.3718	49.3979	54.4428	132.2673	85.2855	110.0193	135.2285	130.0121
high4	108.7823	122.9129	49.4766	54.5283	132.363	85.1473	110.1072	134.9361	130.069

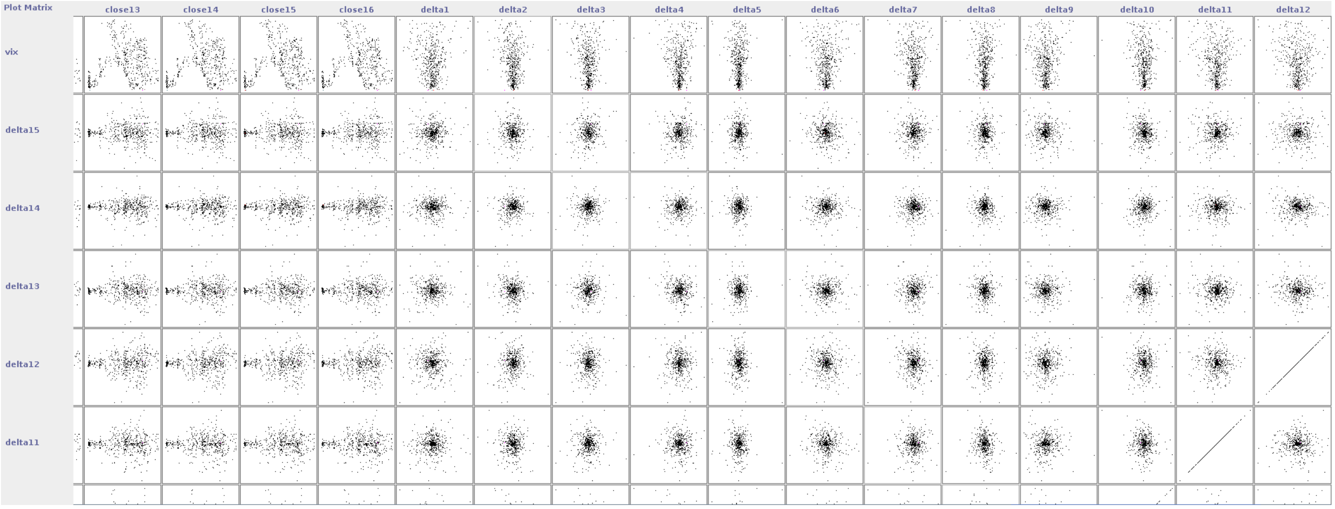
Display of Cluster Centroids for K-Means with 41 Clusters.

Partial Output Displayed Here

	Clustered Instances	
0	35	2.00%
1	23	1.00%
2	28	2.00%
3	12	1.00%
4	34	2.00%
5	19	1.00%
6	52	3.00%
7	62	4.00%
8	2	0.00%
9	29	2.00%

Display of Clustered Instances for K-means with 41 Clusters.

Partial Output Displayed here



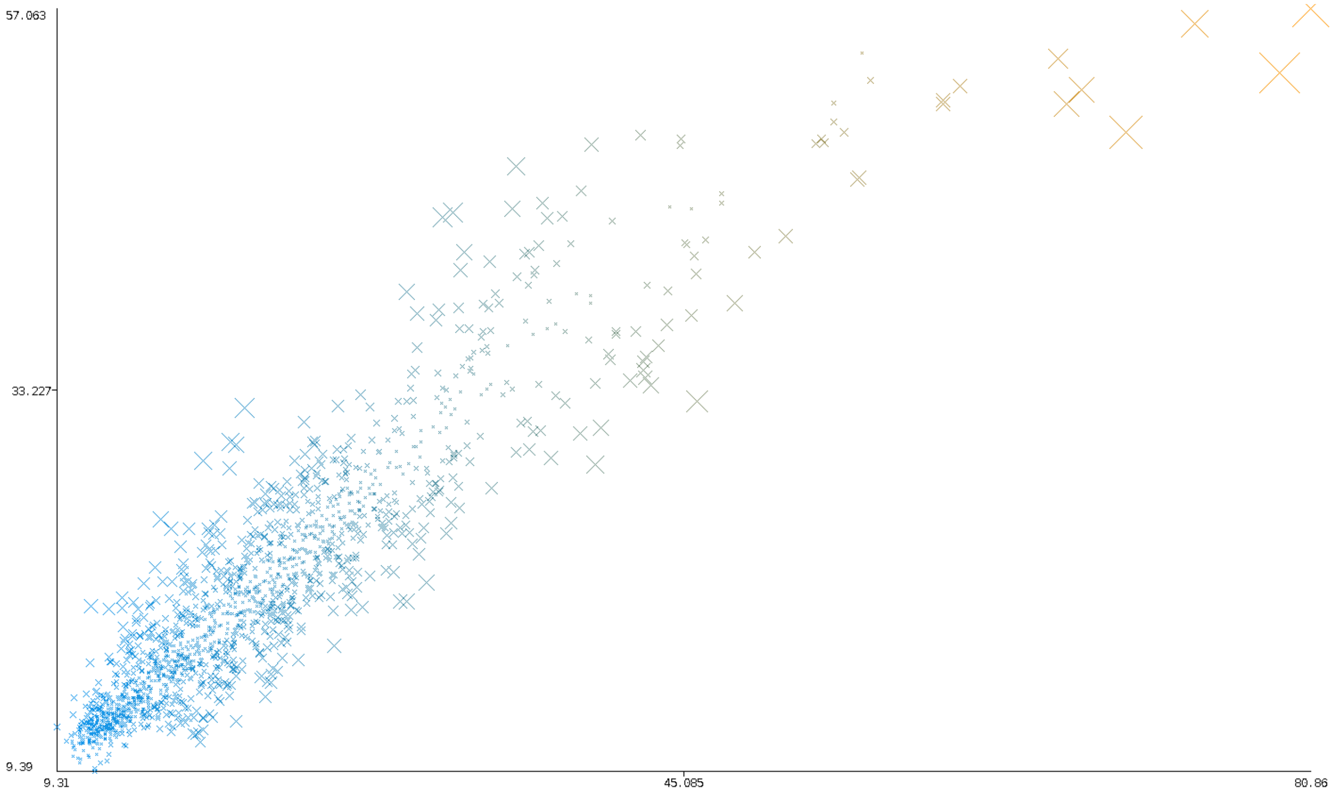
Visualization of Clusters Generated by K-Means as Viewed from Various Perspectives

4. Regression Analysis with Linear Regression:

Correlation Coefficient:	0.89
Mean Absolute Error:	2.73
Root Mean Squared Error:	3.74
Relative Absolute Error:	44.37%
Root Relative Squared Error:	44.21%
Total Number of Instances:	1767

5. Regression Analysis with Support Vector Machines:

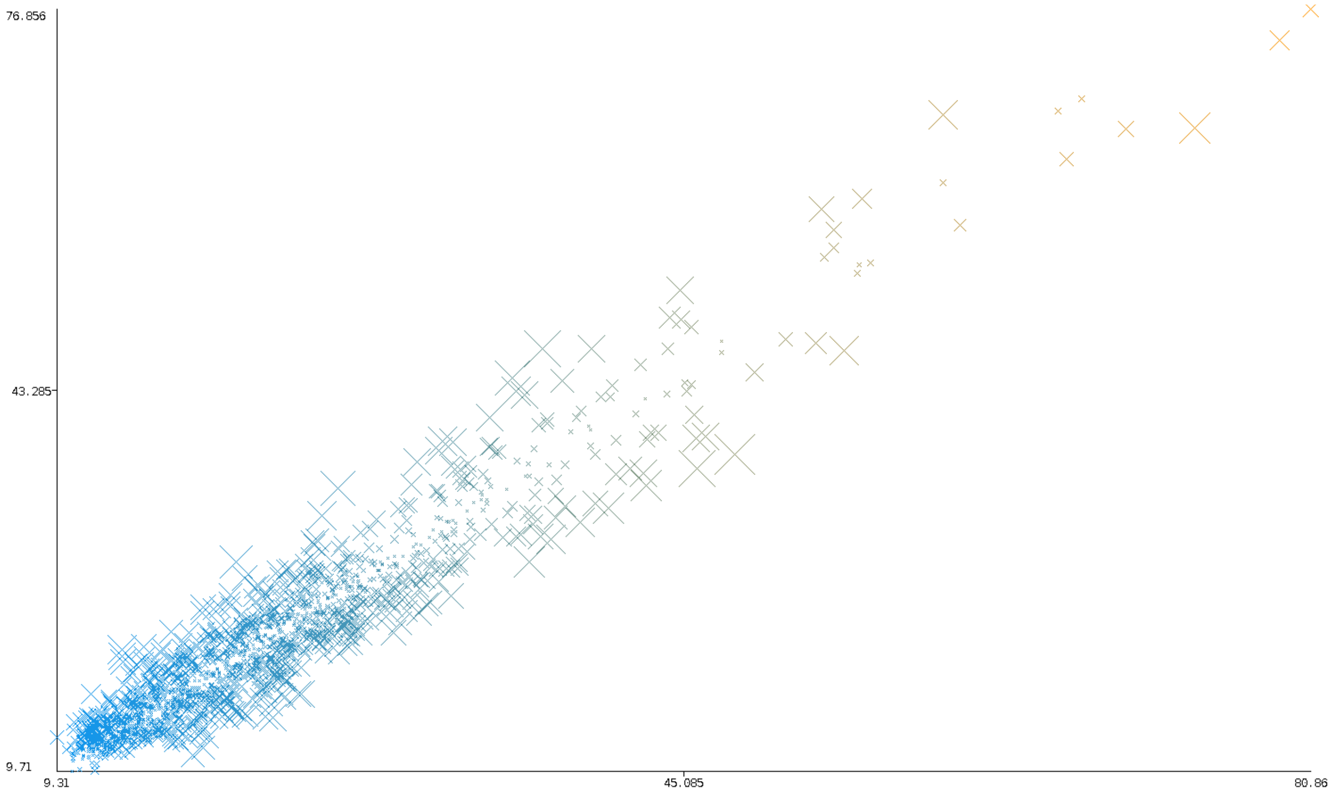
Correlation Coefficient:	0.93
Mean Absolute Error:	2.13
Root Mean Squared Error:	3.13
Relative Absolute Error:	34.62%
Root Relative Squared Error:	37.02%
Total Number of Instances:	1767



Visualization of Errors Generated by the Support Vector Machines Regression Classifier

6. Regression Analysis with Radial Basis Functions:

Correlation Coefficient:	0.96
Mean Absolute Error:	1.82
Root Mean Squared Error:	2.43
Relative Absolute Error:	29.51%
Root Relative Squared Error:	28.77%
Total Number of Instances:	1767



Visualization of Errors Generated by the Radial Basis Function Regression Classifier

Discussion:

Despite the seemingly erratic movement in the stock market, there are embedded patterns that can be discovered and predicted using the Artificial Intelligence algorithms found in this project. Initially the goal was to simply predict whether the market close value for the next day would be higher or lower than the previous day. This was expanded upon to also identify changes in the market so that a turning point from an increase to a decrease (and vice versa) could be predicted. Both of these goals were met with limited success. Using simple classification techniques like Naive Bayes, the model was able to predict the correct market conditions ~55% of the time (versus 20% for random guessing). Using more complex classification techniques like Radial Basis Function Networks, the maximum result achieved was greater than 80%. This result is expected since the stock market data being used is clearly not linearly separable in the current dimension in which it is being analyzed. Thus the ability of the basis functions to project the data into a higher dimension to discover linear separability was directly applicable to this project. While those success rate pales in comparison to that achieved by JPMorgan and Virtu, each was able to generate significant net positive income during a mock ROI computation.

Examining the model's performance from an investment prospective suggests a greater degree of success beyond the 55% or 80% correct classification rate. Specifically, the model was able to avoid drastic down-turns in the market such as the one caused by the housing crash in 2008. In addition, the model was able to successfully predict unstable market conditions that indicated too much volatility. Thus, the total ROI was improved by avoiding buying and selling when the market has many tiny and rapid fluctuations. Both of these points illustrate that while making money is the best type of success, not losing money is another form of success. The only failure then is losing money; which the model was able to avoid in the majority of situations. And as illustrated by the 2008 crash example, the money that was lost was typically in small fluctuating conditions but never in a large drop.

The influence of volatility on the model lead to the next phase of this project; attempt to predict market volatility from past data. The CBOE index, VIX, is a measure of how volatile the stock market is and it is directly computed from the S&P 500. Thus, the existing data from phase one was easily adapted and augmented with VIX data to construct a second and third model.

The second model consisted of nominalizing the VIX data and attempting to categorize market conditions into one of the discretized clusters. The K-Means algorithm was used to experimentally derive the optimal number of clusters. Trials were run with 2, 10, 41, 200 and 500, 2000 and 20000 clusters. In all cases 41 and greater, the data was categorized best using only 41 centroids. While this discovery itself wasn't particularly beneficial, it did lead to vastly improved results in the third model. Using the knowledge of 41 categories, model 2 attempted to then classify the volatility into its nearest neighbor category which was then used to predict the next day's volatility. It is worth noting here that according to CBOE, the creators of VIX, that index is created using 16 day and 44 day sliding windows. Thus this project constructed data sets to train and test model 2 using the same size sliding windows. Despite having this insight into how the VIX data was constructed, the results were poor. In fact they were so poor, they are not reported here and instead work was started on model 3 using what was learned from model 2.

The third model is where the results from this project really start to get interesting. Finally a substantial success rate was achieved (as high as 70% for Radial Basis Functions). As mentioned previously, the knowledge of 41 clusters was used in model 3. For Radial Basis Functions, the 41 value mapped directly to the number of hidden basis nodes in the network. This was clearly a good choice as those results were greater than Linear Regression by about 25% and even greater than Support Vector Machine Regression by approximately 8%. This is a striking improvement since the model type is regression rather than classification and the likelihood of an accidentally correct misclassification is extremely low. The other reason that the high success rate is surprising is illustrated by the cluster visualization from K-Means. Clearly the image above shows that in the best circumstances, the data falls into only one or two clusters (an inverted V [top left] or centered with many outliers [middle and right]).

Despite the challenges inherent within the data, Radial Basis Functions for Regression, like RBF Networks, were able to learn and predict patterns in the VIX data. It is likely that being able to combine the knowledge from the RBF Regression with the RBF Network would yield even higher results. Currently the RBF Network model is skewed toward predicting a volatile market. Incorporating the RBF Regression data could help to set a threshold at which a market really is volatile. This in turn should enable the model to more accurately predict the desires up/down market conditions with a higher rate of success. Ultimately, the model's total accuracy would increase because more data would be correctly classified.

Conclusion:

While this project only scratches the surface of the work that can be done with predicting the stock market using Artificial Intelligence, it nonetheless achieved some surprisingly good results. Even taking a naïve approach to model construction and prediction, the Bayesian classifier was able to produce a net positive result. Shifting gears into predicting market volatility lead to additional classification success that came at the expense of some initial failure. Clustering using K-Means was ultimately a unsuccesful, as were the classification results of attempting to lump data points into one of the categories defined by a cluster centroid. However, using regression analysis algorithms to predict market volatility produced some of the best results seen in this project. Thus the K-Means failures were an essential stepping stone toward what was ultimately achieved since as mentioned previously, the knowledge of 41 clusters was derived from the K-Means analysis. Once the discovery of 41 clusters was made, the regression results immediately jumped to higher success rates capping at an ultimate value of 70% for Radial Basis Function Regression.

Future work to extend this project would ideally focus on combining the prediction model with the regression model. Thus, having insight to the market volatility at any given point could help produce substantial gains (and avoid substantial losses).