

## **Predicting Financial Trends**

### **Abstract**

“Technical analysis is the examination of past price movements to forecast future price movements.”<sup>1</sup>

Financial trends are used to favorably leverage investment decisions. Trends are a group of financial charts and transformations that have similar representations and outcomes. There are three premises to which this technical analysis is based:

1. Market action discounts everything
2. Prices move in trends
3. History repeats itself

If these conditions hold, then it is possible to favorably leverage financial markets by leveraging trend analysis. This project proposes to describe and understand financial trends and apply this knowledge to a mechanical trading system.

---

<sup>1</sup> StockCharts.com Chart School

## **Document Table of Contents**

<b>1. Introduction</b>	<b>Page 3</b>
<b>2. Problem</b>	<b>Page 7</b>
<b>3. System Design</b>	<b>Page 8</b>
<b>4. Data</b>	<b>Page 17</b>
<b>5. Experiments</b>	<b>Page 32</b>
<b>6. Performance</b>	<b>Page 34</b>
<b>7. Conclusion</b>	<b>Page 37</b>
<b>8. References</b>	<b>Page 46</b>

## 1. Introduction

The purpose of this research is to describe and understand financial trends. Historical charts describe trends in market behavior, and if correctly identified, can help produce profitable investments. The proposed task is to describe and identify financial trends systematically without bias. Interesting trends [or patterns] that are identified can then be used to complement the investor's decision-making process.

Trends are observed on price charts, a common financial instrument used for charting prices versus time. A price chart [i.e. time series plot] is a sequence of prices plotted over a time series. Financial charts plot the price [typically on the y-axis] versus time [x-axis].

Technicians, technical analysts and chartists use charts to analyze a wide array of securities and forecast future price movements. The word "securities" refers to any tradable financial instrument such as stocks, bonds, commodities, futures or market indices.

Particular charts contain implicit, previously unknown and potentially useful information in the form of trends. What do we look for when analyzing trends? The answer is three fold:

**Understand:** What does a particular trend mean?

**Describe:** Which parts of a trend are significant?

**Identify:** Could we spot trends and if so, how early could we identify them?

Being provided with a handful of financial data irrespective of time dimensionality, is it possible to identify trends?

### 1.1 Trends

This research covers a subset of the most common trends. These are the most frequent trend lines and reversal patterns. There are eight trends in our 'universe' as follows:

#### 1.1.1 Trend lines

The trend line is one of the simplest trends, but is also one of the most valuable. There are three types of trend lines discussed in this paper: the up trend line [figure 1], the down trend line [figure 2] and the accelerating trend line.

##### Up Trend line

"The up trend line is drawn under the rising reaction lows. A tentative trend line is first drawn under two successively higher lows, but needs a third test to confirm the validity of the trend line."<sup>2</sup>

##### Down Trend line

"A down trend line is drawn over the successively lower rally highs. The tentative down trend line needs two points to be drawn and a third test to confirm its validity"<sup>3</sup>

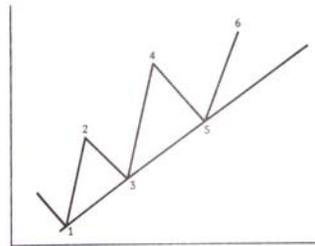
##### Accelerating Uptrend

---

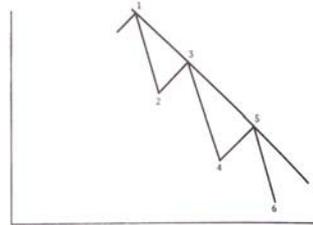
<sup>2</sup> Murphy 65

<sup>3</sup> Murphy 66

"An accelerating uptrend requires the drawing of steeper trend lines. The steepest trend line becomes the most important one."<sup>4</sup>



**Figure 1**



**Figure 2**

### 1.1.2 Reversal

Reversal patterns indicate that an important reversal in trend is taking place.

#### Head and Shoulders

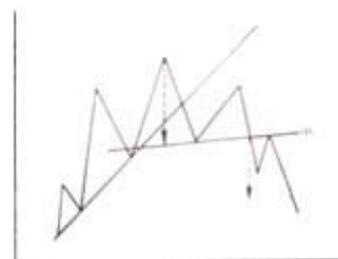
The head and shoulders pattern is probably one of the best known and most reliable of all major reversal patterns.<sup>5</sup> The head and shoulders reversal [figure 3] and the head and shoulders inverse reversal pattern [figure 4] are further refinements of a combination of trend line concepts.

#### Head and Shoulders

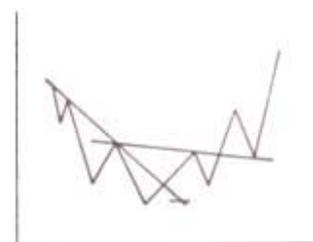
"A head and shoulders reversal pattern forms after an uptrend, and its completion marks a trend reversal. The pattern contains three successive peaks with the middle peak (head) being the highest and the two outside peaks (shoulders) being low and roughly equal. The reaction lows of each peak can be connected to form support, or a neckline."<sup>6</sup>

#### Head and Shoulders Inverse

"As a major reversal pattern, the head and shoulders bottom forms after a downtrend, and its completion marks a change in trend. The pattern contains three successive troughs with the middle trough (head) being the deepest and the two outside troughs (shoulders) being shallower. Ideally, the two shoulders would be equal in height and width. The reaction highs in the middle of the pattern can be connected to form resistance, or a neckline."<sup>7</sup>



**Figure 3**



**Figure 4**

#### Triple Top and Bottom

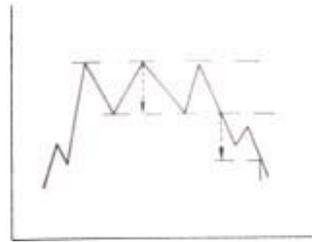
<sup>4</sup> Murphy 79

<sup>5</sup> Murphy 103

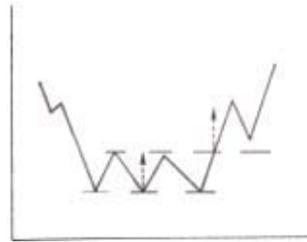
<sup>6</sup> StockCharts.com Chart School – Head and Shoulders

<sup>7</sup> StockCharts.com Chart School – Inverse Head and Shoulders

The triple top [figure 5] or bottom [figure 6] is just a slight variation of the head and shoulders pattern. The main distinction is that the three peaks or troughs are at about the same level, much like the name implies. It is often unclear and frequently argued whether a trend signifies a head and shoulders [inverse] or triple top [bottom] pattern. Machine learning offers a unique approach to clarifying this argument, because it is always consistent and experimentally exact.



**Figure 5**

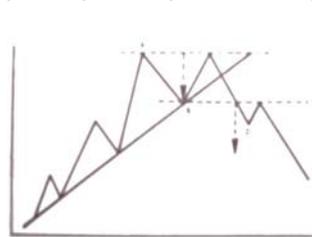


**Figure 6**

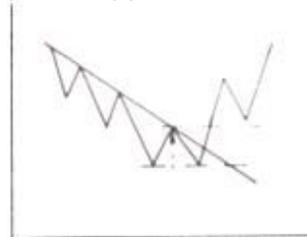
### **Double Top and Bottom**

The double top [figure 7] and bottom [figure 8] are closely related to the triple top and bottom except for the number of peaks and troughs. By now, it is obvious to see the implication in the name. There are two peaks and troughs of roughly the same level. This pattern, next to the head and shoulders, is the most frequently seen and easily recognized.

“The general characteristics of a double top are similar to that of the head and shoulders and triple top except that only two peaks appear instead of three.”<sup>8</sup>



**Figure 7**



**Figure 8**

---

<sup>8</sup> Murphy 118

## **1.1 Motivation**

The motivation to pursue this research is three fold.

1. Economic incentive
2. Combination of interests
3. Burgeoning industry

The economic incentive is straight-forward. By researching trends and the financial markets, it is likely that I'll develop a better understanding of the financial markets which is beneficial in itself. This research adds another tool to the growing bag of market applications that could be used to analyze investments. And the process of understanding the market and how it works allows us to grow as investors.

The research branches out across several industries and academic fields, all of which I'm interested in. Finance and economics play a vital role in understanding the domain. Data mining and data warehousing are burgeoning fields and are central to my area of interest as a computer scientist. And mathematics and statistics overlap many of these fields and provide tools for verification, validation and algorithms that are readily used in this application.

Analytical research in financial markets is not a new concept, but the burgeoning technologies and sciences such as data mining and warehousing are helping revolutionize and pioneer this industry.

## 2. Problem

Technical analysis [i.e. charting] is commonly used and less frequently understood. Each chartist has their own perspectives on trends; what do they mean and how to leverage them to their advantage. These opinions are just that and all too many authorities offer no sound [i.e. statistical] reasoning why trends are what they are.

Murphy's "Technical Analysis of the Financial Markets" describes financial patterns and how to read them. "John Murphy draws upon his thirty years of experience in the field to develop ten basic laws of technical trading. These precepts define the key tools of technical analysis and how to use them to identify buying and selling opportunities".

'Wisdom' is interesting to read but does not define trends. How could we know if the trend summaries or psychology that Murphy describes are accurate? This project allows us to gauge the value of these patterns and describe several types of financial trends. It allows us to see first-hand the impact of these patterns.<sup>9</sup>

This research does not undermine Murphy's years of experience and wisdom. Quite the contrary, Murphy's strategies have made me a more versatile investor and I have profited from the knowledge dug up in his books. This has made me wonder "Is there more?"

Analyzing trends carries an inherent bias. Every investor sees charts differently based on their level of technical awareness and their own personal interest in the market. The personal investor may alter the perception of a chart. [i.e. it may be tempting to perceive an up trend line for a stock you own, although the bigger picture may show something contrary such as a measured move down]. Investors are quick to adopt emerging trends. They are even quicker to drop them. Trends are abused; they are glorified and holy when they work and scoffed at when they don't.

The problem is that trends are difficult to select in large quantities in order to provide statistical support. Trends are defined and it is possible to select a particular trend in a line up of trends, even easy. However, it is a far more difficult task to select a handful of like-trends quickly and efficiently.

The approach taken is to study the trends we have and gain enough insight to automate the selection of observed trends. What we're trying to do is understand trends, learn to describe them and be able to identify them mechanically and efficiently.

---

<sup>9</sup> John Murphy's Ten Laws, <http://stockcharts.com/education/What/TradingStrategies/MurphysLaws.html>

### 3. System Design

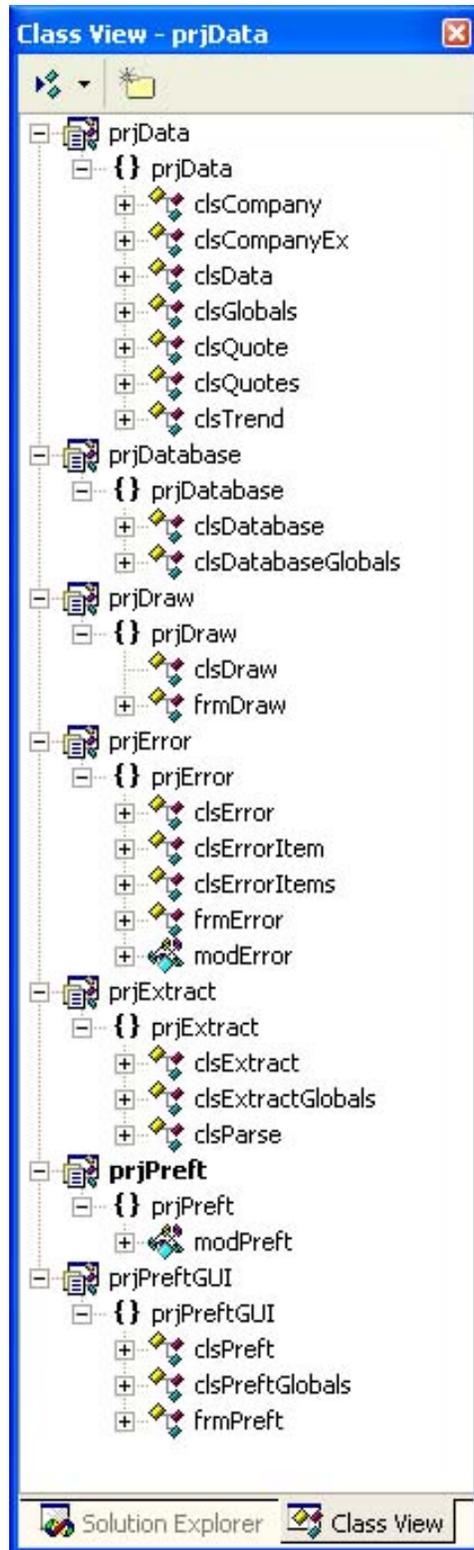
#### 3.1 Infrastructure

This project requires a proprietary infrastructure to extract, clean and transform financial data. The data acquisition, cleaning and preparing are significant parts of the data warehousing and pre-mining process. The following are [just some of] those tasks supported by this application:

1. Extract data from several sources such as Yahoo! Finance. [See 'prjExtract']
2. Warehouse data using SQL Server 2000. [See 'prjDatabase']
3. Provide the mathematical functionality to transform and smooth financial data.
  - a. Binning allows us to smooth the data on the x-axis [default of 10 positions]
  - b. Normalization on the y-axis adjusts all equity prices to a unit percentage scale
4. Charting and a plethora of charting functionality.
5. Supervise and select trends.
6. Output artificial datasets [i.e. ARFF artificial dataset generation].

The infrastructure: The design is implemented with Microsoft Visual Studio [VB.NET], SQL Server 2000 and makes use of SQL Analysis Manager and Query Analyzer.

**Code snippets:** The following is a list of components and code that highlights the major tasks of each embedded component. [Please refer to the code set on the companion cd.]



The above figure is that of the project class view [April 29, 2003]

**PrjData:** This component maintains all the data and supervised trends. The mathematical functions are contained in each class in an object orientated [and inherited] environment.

1. ClsTrend: Supervised dataset. This is basically a selection of clsData marked as a trend.
2. ClsCompany: Equity information such as ticker symbol, enumerated market, sector and industry.
3. ClsData: The highest-level data object. This contains the clsCompany and clsQuotes.
4. ClsQuotes: Maintains a collection of single day quotes or clsQuote[s].
5. ClsQuote: One day's quote including the open and close, low and high and volume.

```
Public Class clsCompany

'////////////////////////////////////
'//
'// File:          clsCompany.vb
'//
'// Description:   Maintains the concept of an equity
'//
'// Author:       Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

Public Class clsQuote

'////////////////////////////////////
'//
'// File:          clsQuote.vb
'//
'// Description:   Maintains the concept of a quote
'//
'//                Statistical math functions for a single quote object
'//
'// Author:       Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

Public Function GetMovingAverage(ByVal oQuotes As prjData.clsQuotes, _
    Optional ByVal iMovingAverage As Integer = 50, _
    Optional ByVal nwWeight As enumMathNormalizeWeight = nwBack) _
    As prjData.clsQuote

Public Function GetRatio(ByVal oQuotes As prjData.clsQuotes, _
    Optional ByVal iRatio As Integer = 50, _
    Optional ByVal nwWeight As enumMathNormalizeWeight = nwBack) _
    As prjData.clsQuote
```

Public Class clsQuotes

```
'////////////////////////////////////
'//
'// File:           clsQuotes.vb
'//
'// Description:    Maintains the concept of a collection of quotes
'//
'//                 Statistical math functions for a quotes object
'//
'// Author:         Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function GetStandardize() As prjData.clsQuotes
'//
'// Parameters:     None
'//
'// Returns:        clsQuotes
'//
'// Scope:         Public
'//
'// Description:    Normalizes the values [the y-axis]
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function GetStretch(ByVal dblSplit As Double, _
'//                 Optional ByVal bmEnum As enumMathBinMethods = bmMean) _
'//                 As prjData.clsQuotes
'//
'// Parameters:     (IN) Double - the number of partitions desired [i.e. def 10]
'//                 [IN] enumMathBinMethods - leverage of the binning
'//                 i.e., mean, median, mode, weighted, etc.,
'//
'// Returns:        clsQuotes
'//
'// Scope:         Public
'//
'// Description:    Stretches [or constricts] the data [the x-axis]
'//
'////////////////////////////////////
```

```
Public Class clsData

'////////////////////////////////////
'//
'// File:          clsData.vb
'//
'// Description:   Maintains the concept of an entire equity
'//
'// Author:       Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

Public Class clsTrend

'////////////////////////////////////
'//
'// File:          clsTrend.vb
'//
'// Description:   Maintains the concept of a trend
'//
'//                This is basically a supervised clsData wrapper
'//
'// Author:       Eldar Radovici, 03/21/03
'//
'////////////////////////////////////
```

**PrjExtract:** This component is focused on extracting data from several sources

1. Extract from web resources such as Yahoo finance and Nasdaq.com.
2. Parse pages into readily available class structures [clsData].

```
Public Class clsExtract

'////////////////////////////////////
'//
'// File:          clsExtract.vb
'//
'// Description:   Extraction functionality
'//
'// Author:       Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:       Function Extract(ByVal strStock As String, _
'//              Optional ByVal strStart As String = vbNullString, _
'//              Optional ByVal strEnd As String = vbNullString) _
'//              As prjData.clsQuotes
'//
'// Parameters:   (IN) String          - Stock [or company] ticker symbol
'//              [IN] String          - Start date
'//              [IN] String          - End date
'//
'// Returns:     prjData.clsQuotes    - Quotes object
'//
'// Scope:       Public
'//
'// Description:   Extract financial quotes from the web and feed them through the database
'//
'// Assumptions:   If the optional parameters are vbNullString, then it's date() and
'//
'////////////////////////////////////
```

```
Public Class clsParse

'////////////////////////////////////
'//
'// File:           clsParse.vb
'//
'// Description:    Parsing functionality
'//
'// Author:         Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function ParseYahoo(ByVal strInput As String) As prjData.clsQuotes
'//
'// Parameters:     (IN) String           - Quotes spreadsheet
'//
'// Returns:        prjData.clsQuotes    - Quotes object
'//
'// Scope:          Public
'//
'// Description:    Converts the passed in Yahoo formatted quotes [via HTTP request]
'//                to a new clsQuotes instance
'//
'// Assumptions:    Yahoo quotes format doesn't change too often
'//
'////////////////////////////////////
```

**PrjDatabase:** Interaction with the development SQL Server 2000 database

1. Communicate with SQL Server and OLAP services.
2. Save/load data and trend data structures..

Public Class clsDatabase

```
'////////////////////////////////////
'//
'// File:           clsDatabase.vb
'//
'// Description:    SQL 2000 functionality. Saving/loading data/trends.
'//
'// Author:         Eldar Radovici, 03/21/03
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function SaveAsTrendCollection(ByVal colTrends As Collection, _
'//                Optional ByVal bNominal As Boolean = False) As Boolean
'//
'// Parameters:     (IN) Collection - collection of clsTrends
'//                [IN] Boolean    - if the accuracy is binned/nominal
'//
'// Returns:        Boolean        - True if successful, otherwise False
'//
'// Scope:          Public
'//
'// Description:    Saves the collection of supervised financial data
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function SaveAsDataCollection(ByVal colData As Collection)
'//                As Boolean
'//
'// Parameters:     (IN) Collection - collection of clsData
'//
'// Returns:        Boolean        - True if successful, otherwise False
'//
'// Scope:          Public
'//
'// Description:    Saves the collection of financial data
'//
'////////////////////////////////////

'////////////////////////////////////
'//
'// Method:         Function LoadTrends() As Collection
'//
'// Parameters:     -
'//
'// Returns:        Collection     - collection of clsTrends
'//
'// Scope:          Public
'//
'// Description:    Returns a collection of all the supervised trends in the
'//                SQL Server datamart.
'//
'////////////////////////////////////
```

```
'////////////////////////////////////  
'//  
'// Method:          Function LoadDataStore() As Collection  
'//  
'// Parameters:      -  
'//  
'// Returns:         Collection - collection of clsData  
'//  
'// Scope:           Public  
'//  
'// Description:     Returns a collection of all clsData members in the  
'//                  SQL Server datawarehouse.  
'//  
'////////////////////////////////////
```

## 4. Data

The Quotes table in diagram 4 is the fact table that warehouses all the data information. The following is the extended star schema [snowflake] structured relationship.

The relationship:

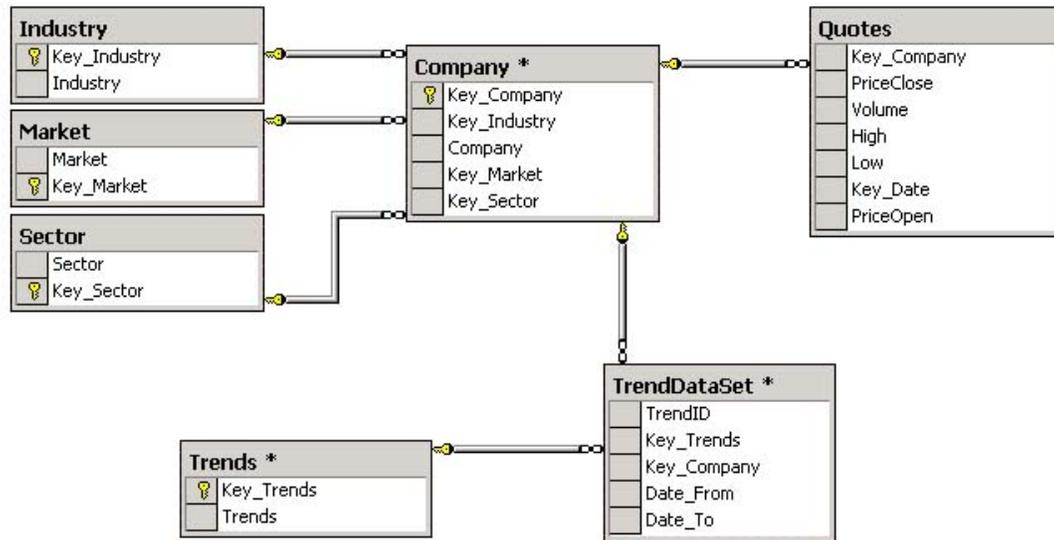


Diagram 4

### 4.1 Universe

The 'universe' is all the securities used in this project. These include 35 stocks from the NYSE and NASDAQ markets and across several sectors and industries. The cubes below segment the stocks into the two markets and illustrate the wide variety of sectors and industries.

- Market	- Sector	- Industry	Company	
All Company Names	All Company Names Total			
- nasdaqm	nasdaqm Total			
	- services	services Total		
		- retail (specialty)	retail (specialty) Total	
			amzn	ebay
	- technology	technology Total		
		- communications equipme	communications equipment	
			andw	cscoc
		- computer hardware	computer hardware Total	
			dell	sunw
		- computer services	computer services Total	
			cnet	yhoo
		+ semiconductors	semiconductors Total	
	- software programming	software programming Tol		
		merq	msft	
		orcl	vrsn	
- nasdaqsc	nasdaqsc Total			
	- technology	technology Total		
		- software programming	software programming Tol	
		aklm		

**Figure 4.1a**

- nyse	nyse Total		
	- basic materials	basic materials Total	
		- metal mining	metal mining Total aa
	- conglomerates	conglomerates Total	
		- conglomerates	conglomerates Total utx
	- consumer cyclical	consumer cyclical Total	
		- audio video equipment	audio video equipment Total sne
		- auto truck manufacturers	auto truck manufacturers f
	- consumer non-cyclical	- footwear	footwear Total nke
		consumer non-cyclical Total	
	- energy	- personal household products	personal household products
		energy Total	
	- financial	- oil gas operations	oil gas operations Total apc
		financial Total	
	- healthcare	- insurance (prop. casualty)	insurance (prop. casualty) ace
		- money center banks	money center banks Total abn
	- services	healthcare Total	
		- biotechnology drugs	biotechnology drugs Total frx
	- technology	- major drugs	major drugs Total pfe
		services Total	
	- transportation	- communications services	communications services Total t
		- retail (department discount)	retail (department discount) wmt
	- technology	technology Total	
		- communications equipment	communications equipment mot
		- computer services	computer services Total aol
		- computer storage devices	computer storage devices emc
		- electronic instruments companies	electronic instruments companies a rok
		- semiconductors	semiconductors Total amd
	- transportation	transportation Total	
		- airline	airline Total alk

Figure 4.1b

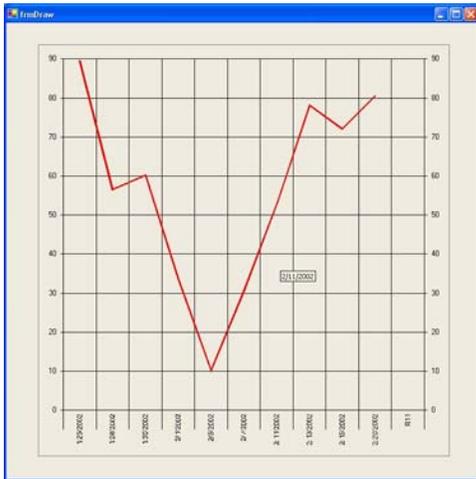
## 4.2 Selection and Supervision

The data selection process is the most time-consuming part of this project. There is an abundant amount of data but the actual trends are scarce. Initially I had expected to find exhaustive resources online and in technical analysis textbooks. That couldn't have been further from the truth. Technical analysis resources that were used explained the trends and concepts well, but failed to provide any observed examples or data, the guts of any data mining project. The resources online were similar and rarely provided any acceptable examples.

Therefore the data selection process consisted of me charting time-series selections of equities and honing in on useful trends. This was a slow and dizzying process, which is why the number of trends were reduced and continuation patterns were left out entirely.

Although the selected trends were small in number, there was a trick of the trade that increased the bounty of the warehoused trends. I was able to stretch the utility of a single instance. For example, the two figures below are separated by a mere weekend but their representation varies making them distinct and equally useful.

Sony 01/25/2002 – 02/21/2002



**Figure 4.2a**

Sony 01/25/2002 – 02/25/2002



**Figure 4.2b**

With this technique, I was able to roughly triple the sample size provided for the rarest of trends, such as inverse head and shoulders and triple tops and bottoms. The increase of instances from 53 [approximately April 16<sup>th</sup>, 2003] to 153 [presently] increased the average accuracy from 66% to 95%! There were other factors such as cleanliness that weighed into the increased accuracy. "Data, data, data" are the three most important parts of any data mining process.

The next few pages detail the entire list of supervised instances. For each trend, there is an accompanying chart produced by the proprietary application. This chart demonstrates the transformation and normalization necessary to produce these trends [i.e. useful information] from the charts [i.e. enormous amount of data].

The normalization transformed the securities across different prices and times into a comparable trend unit. This chart was stretched to 10 points on the x-axis and zero to one hundred percent on the y-axis to produce the usable trends.

## Up Trend line

1. MSFT. 02/25/99 – 03/01/00
2. MSFT. 02/25/99 – 04/04/00
3. MSFT. 02/1/99 – 04/04/00
4. MSFT. 01/1/99 – 04/04/00
5. MSFT 1/22/99 – 3/24/00
6. MSFT 1/22/99 – 4/1/00
7. MSFT 4/25/97 – 7/3/97
8. MSFT 4/25/97 – 8/1/97
  
9. AMZN. 09/01/01 – 03/01/03
10. AMZN 10/15/01 – 04/15/02
11. AMZN 11/6/01 – 04/15/02
- 12. AMZN 07/25/02 – 11/15/02**
13. AMZN 07/25/02 – 11/5/02
14. AMZN 07/25/02 – 10/3/02
  
15. YHOO 6/5/98 – 7/1/98
16. YHOO 6/5/98 – 6/26/98
17. YHOO 6/5/98 – 7/15/98
  
18. ACE 5/15/95 – 12/15/95
19. ACE 5/15/95 – 10/27/95
  
20. VRSN 4/3/01 – 4/22/01
21. VRSN 12/4/98 – 1/25/99
  
22. INTC 6/9/99 – 1/7/00
23. CSCO 6/10/99 – 10/27/99
24. ANDW 6/23/94 – 10/5/94
25. FRX 12/6/96 – 12/15/97

## Amazon.com

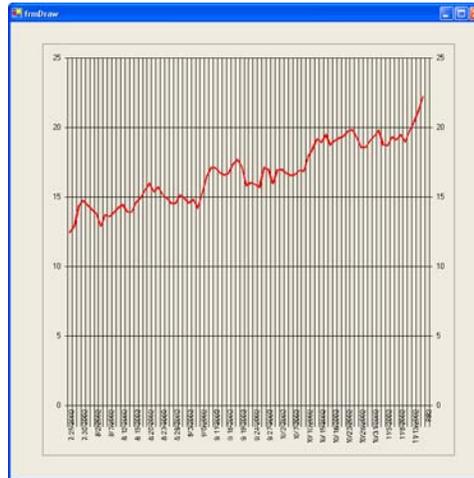


Figure 4.2.1a. Chart.

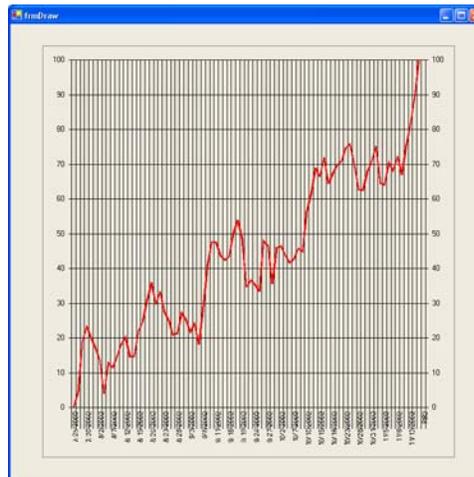


Figure 4.2.1b. Normalized [0-100]%

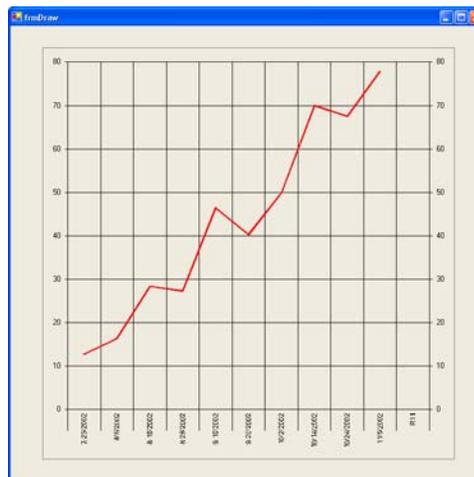


Figure 4.2.1c. Stretched [0..9] points.

## Down Trend line

26. **ACE 5/10/99 – 8/1/99**
27. ACE 5/10/99 – 8/20/99
28. ACE 5/20/99 – 7/20/99
29. ACE 6/15/02 – 7/10/02
  
30. DELL 7/10/00 – 10/7/00
31. DELL 7/10/00 – 8/18/00
  
32. EBAY 9/20/00 – 11/6/00
33. EBAY 9/20/00 – 10/9/00
34. EBAY 6/22/01 – 8/7/01
35. EBAY 7/16/01 – 8/7/01
  
36. SUNW 10/13/00 – 1/20/01
37. SUNW 10/13/00 – 12/7/01
  
38. MOT 7/1/97 – 8/1/98
39. MOT 7/1/97 – 12/9/97
40. MOT 7/1/97 – 5/12/98
41. MOT 11/19/97 – 7/31/98
42. MOT 10/2/00 – 12/12/00
  
43. AOL 5/3/01 – 8/19/02
44. AOL 11/15/01 – 8/19/02
  
45. MERQ 9/25/00 – 10/25/00
46. MERQ 2/14/01 – 3/9/01
  
47. WMT 12/20/99 – 2/10/00
48. VRSN 10/18/01 – 4/15/02
49. AMD 3/6/02 – 7/19/02
50. CNET 8/10/01 – 8/31/01

## ACE



Figure 4.2.2a. Chart.



Figure 4.2.2b. Normalized [0-100]%

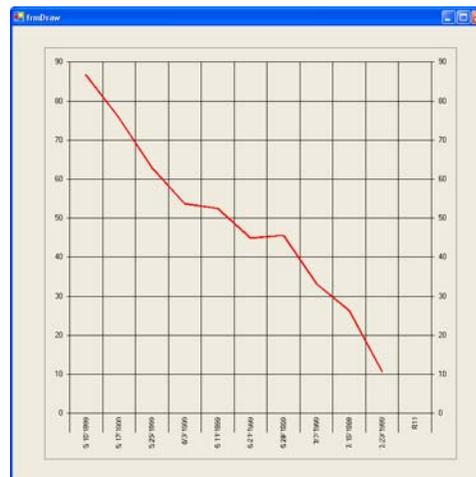


Figure 4.2.2c. Stretched [0..9] points.

## Accelerating Up Trend lines

51. EMC. 12/ 20/97 – 01/10/00
52. EMC. 12/ 22/97 – 4/12/99
53. EMC. 4/21/97 – 01/10/00
54. EMC. 4/20/99 – 01/4/00
  
55. YHOO 9/1/98 – 1/6/99
56. YHOO 9/1/98 – 02/1/99
57. YHOO 10/13/99 – 11/23/99
58. YHOO 10/15/99 – 12/26/99
59. YHOO 10/15/99 – 12/26/99
60. YHOO 10/10/98 – 11/23/98
  
61. VRSN 10/15/99 – 12/19/99
62. VRSN 10/15/99 – 12/7/99
  
- 63. MSFT 7/16/95 – 7/22/97**
64. MSFT 1/11/96 – 7/22/97
65. MSFT 1/11/96 – 1/29/97
  
66. DELL 6/12/96 – 7/24/97
67. DELL 6/1/96 – 5/11/98
  
68. NKE 8/30/93 – 11/21/95
69. NKE 8/30/93 – 6/30/93
  
70. AOL 1/23/97 – 7/6/98
71. AOL 1/23/97 – 1/15/99
  
72. SUNW 7/21/99 – 9/24/99
73. CSCO 12/26/97 – 7/20/98
74. EMC 12/23/97 – 2/12/99
75. ROK 1/10/95 – 5/2/95
76. INTC 12/10/95 – 1/9/97

## Microsoft

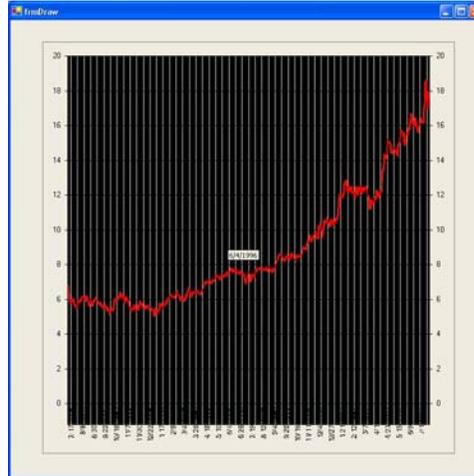


Figure 4.2.3a. Chart.

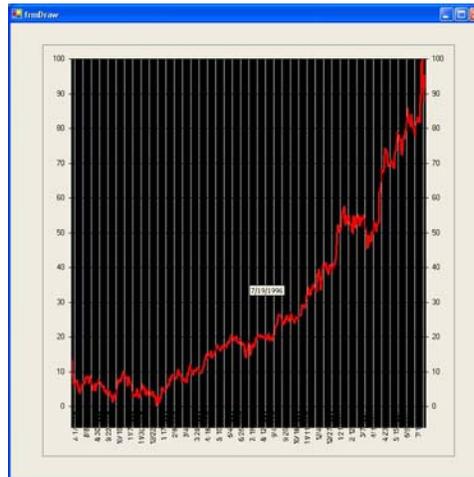


Figure 4.2.3b. Normalized [0-100]%

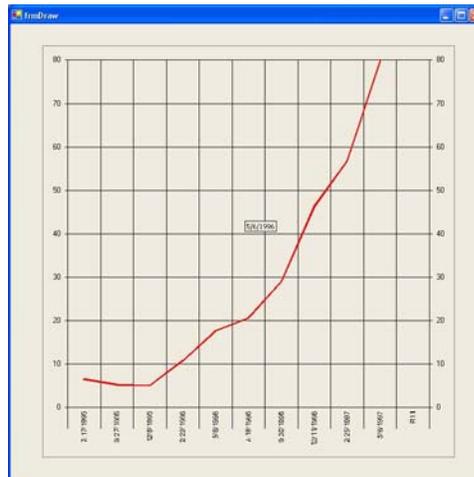


Figure 4.2.3c. Stretched [0..9] points.

## Head and Shoulders Reversal

- 77. CNET 8/1/99 – 6/1/00
- 78. CNET 9/8/99 – 4/12/00

### 79. NKE 6/1/96 – 1/1/98

- 80. NKE 6/1/96 – 10/15/97
- 81. NKE 8/20/96 – 8/29/97
- 82. NKE 9/1/96 – 8/29/97

- 83. YHOO 12/15/99 – 1/18/00
- 84. YHOO 12/16/99 – 1/14/00
- 85. YHOO 12/17/99 – 1/12/00

- 86. MOT 1/27/00 – 4/10/00
- 87. MOT 1/31/00 – 4/01/00
- 88. MOT 2/1/00 – 4/01/00
- 89. MOT 11/11/97 – 12/8/97
- 90. MOT 11/13/97 – 12/3/97
- 91. MOT 1/31/00 – 3/30/00

## Nike

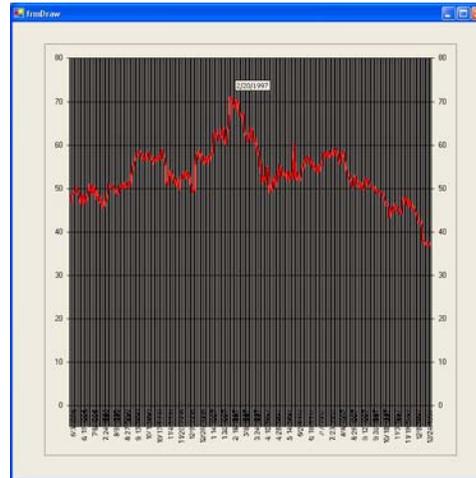


Figure 4.2.4a. Chart.

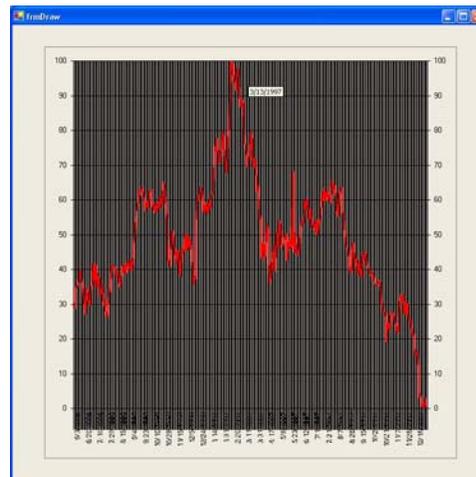


Figure 4.2.4b. Normalized [0-100]%

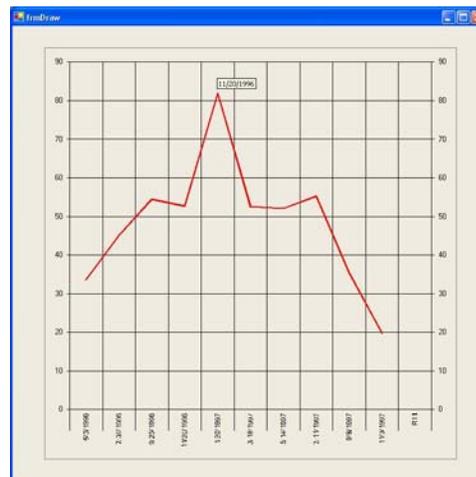


Figure 4.2.4c. Stretched [0.9] points.

## Inverse Head and Shoulders

- 92. ALK 8/15/98 – 12/15/98
- 93. ALK 8/25/98 – 12/4/98**
- 94. ALK 8/17/98 – 12/25/98
- 95. ALK 8/25/98 – 12/25/98
  
- 96. T 6/20/98 – 10/22/98
- 97. T 6/22/98 – 10/26/98
  
- 98. ABN 2/15/03 – 4/9/03
- 99. ABN 2/21/03 – 4/5/03
  
- 100. SNE 1/24/02 – 2/25/02
- 101. SNE 1/24/02 – 2/22/02
- 102. SNE 1/25/02 – 2/21/02
- 103. SNE 1/25/02 – 2/25/02

## Alaska Air



Figure 4.2.5a. Chart.



Figure 4.2.5b. Normalized [0-100]%

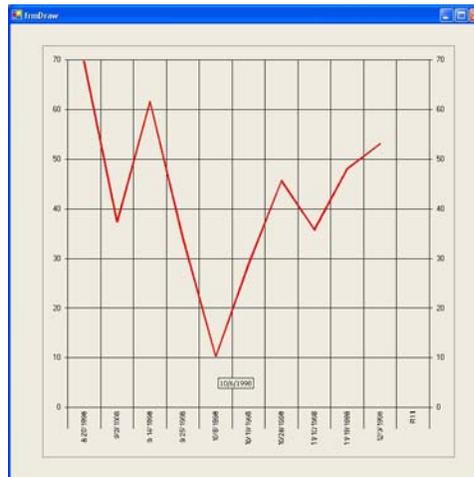


Figure 4.2.5c. Stretched [0..9] points.

## Triple Tops

- 104. UTX 3/1/99 – 9/22/99
- 105. UTX 3/1/99 – 9/21/99
- 106. UTX 3/5/99 – 9/22/99
- 107. UTX 3/10/99 – 9/22/99

### 108. ROK 4/28/99 – 9/15/99

- 109. ROK 5/1/99 – 9/10/99
- 110. ROK 5/1/99 – 9/9/99
- 111. ROK 5/1/99 – 9/8/99

- 112. G 2/24/98 – 7/30/98
- 113. G 3/2/98 – 7/30/98
- 114. G 3/5/98 – 7/30/98

## Rockwell

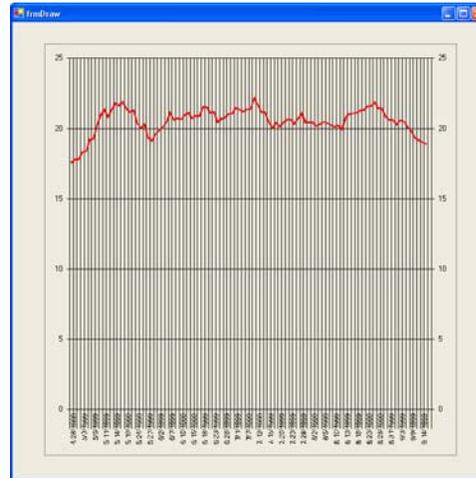


Figure 4.2.6a. Chart.

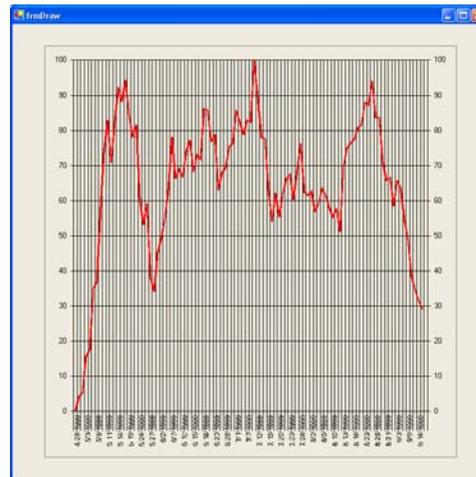


Figure 4.2.6b. Normalized [0-100]%

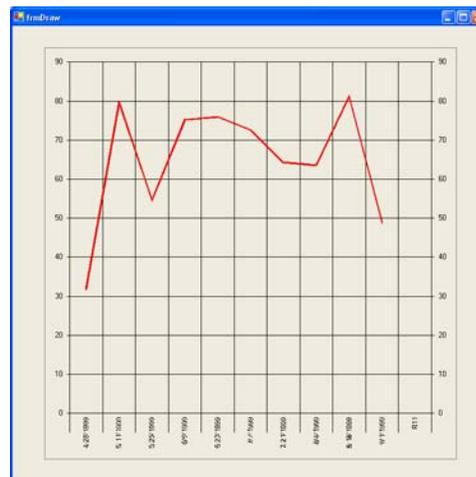


Figure 4.2.6c. Stretched [0..9] points.

## Triple Bottoms

- 115. APC 9/1/99 -4/20/00
- 116. APC 9/1/99 -3/31/00
- 117. APC 9/10/99 -3/31/00
- 118. APC 9/8/99 - 4/2/00
  
- 119. ANDW 5/1/98 - 2/1/00
- 120. ANDW 5/27/98 - 2/1/00
- 121. ANDW 6/1/98 - 2/1/00
- 122. ANDW 6/1/98 - 1/10/00
- 123. ANDW 6/1/98 - 1/15/00
- 124. ANDW 6/1/98 - 1/20/00**
- 125. ANDW 5/1/98 - 1/25/00
  
- 126. EMC 9/17/01 - 10/25/01

## Andrew Corporation

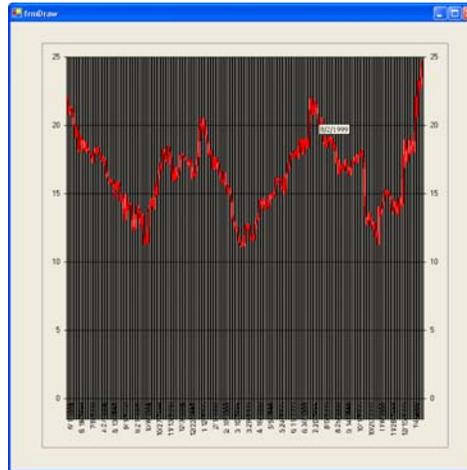


Figure 4.2.7a. Chart.

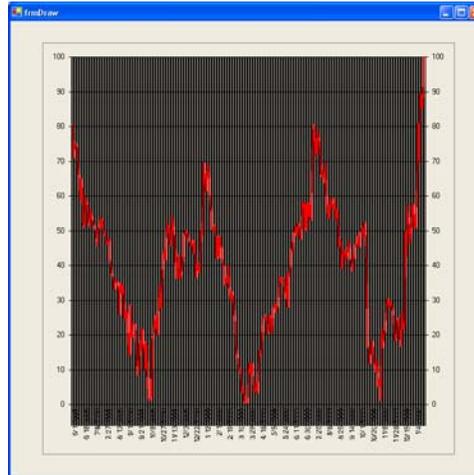


Figure 4.2.7b. Normalized [0-100]%

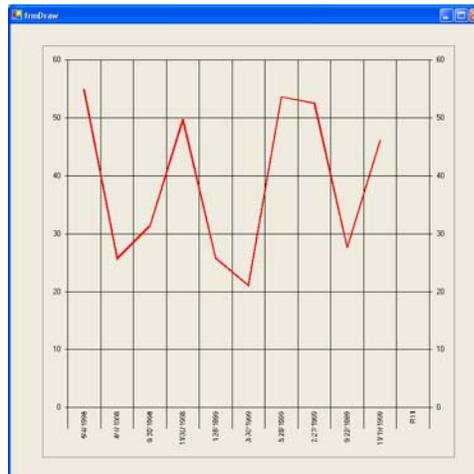


Figure 4.2.7c. Stretched [0..9] points.

## Double Tops

- 127. G 8/20/97 – 10/26/99
- 128. G 8/20/97 – 10/1/99
- 129. G 8/20/97 – 9/25/99**
  
- 130. F 12/1/98 – 7/1/99
- 131. F 12/1/98 – 6/16/99
- 132. F 12/10/98 – 6/16/99
- 133. F 12/5/98 – 6/16/99
  
- 134. PFE 4/8/98 – 8/20/98
- 135. PFE 4/12/98 – 8/13/98
  
- 136. UTX 1/20/98 – 9/1/98
- 137. UTX 1/22/98 – 9/1/98
- 138. UTX 1/30/98 – 9/1/98
- 139. UTX 1/30/98 – 8/20/98
- 140. UTX 2/10/98 – 8/20/98
- 141. UTX 3/1/98 – 8/20/98
  
- 142. ABN 7/5/98 – 8/5/98
- 143. ABN 7/13/98 – 8/4/98

## Gillette

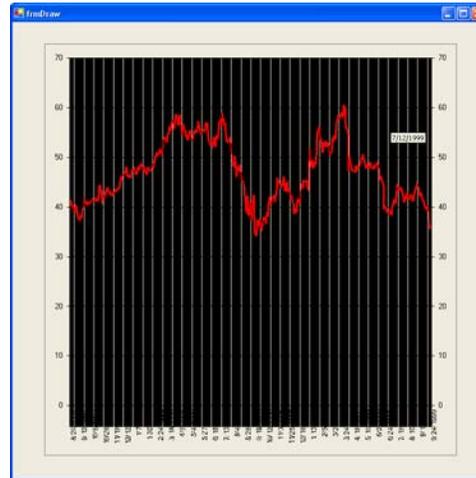


Figure 4.2.8a. Chart.

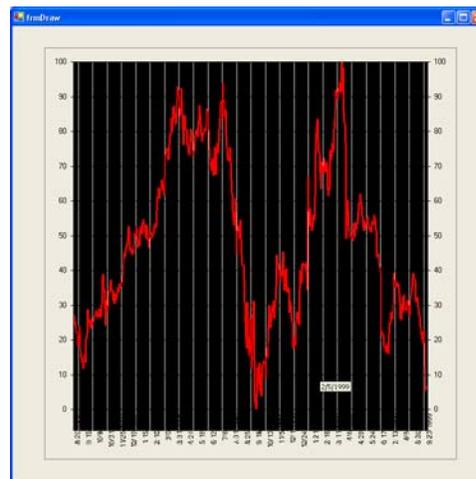


Figure 4.2.8b. Normalized [0-100]%

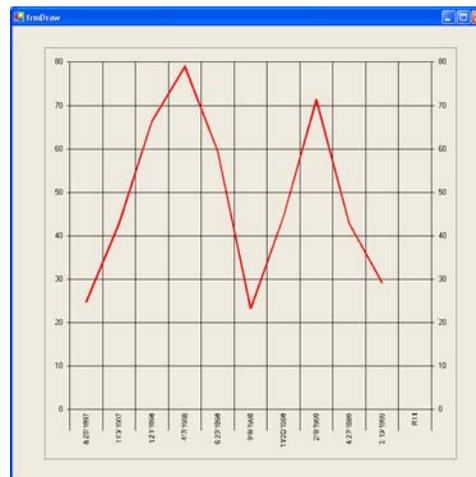


Figure 4.2.8c. Stretched [0.9] points.

## Double Bottoms

- 144. MSFT 12/14/00 – 1/8/01
- 145. MSFT 12/16/00 – 1/8/01
- 146. MSFT 12/16/00 – 1/5/01
  
- 147. SNE 3/8/01 – 3/22/01
- 148. SNE 10/22/98 – 11/23/98**
- 149. SNE 10/23/98 – 11/18/98
- 150. SNE 10/25/98 – 11/18/98
  
- 151. UTX 10/15/97 – 1/28/98
- 152. UTX 10/20/97 – 1/27/98
  
- 153. PFE 11/20/99 – 4/2/00

## Sony

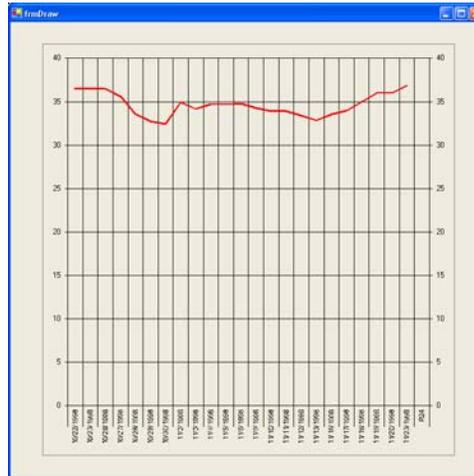


Figure 4.2.9a. Chart.



Figure 4.2.9b. Normalized [0-100]%

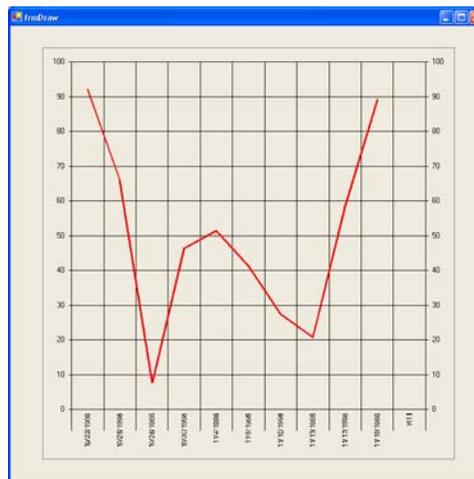


Figure 4.2.9c. Stretched [0..9] points.

## 4.4 Warehousing

The database contains roughly 35 equities with at most 10 years of financial data. That equates to an estimated 100 thousand records stored in the Quotes fact table [see relationship diagram 4].

There are additional tables such as the TrendSet and TrendSetNominal tables. These are used exclusively with SQL Server Analysis Manager. TrendSet contains the trends with numeric classes and TrendSetNominal discretizes the classes into nominal values. This was necessary to make DB Miner's classifier happy. The analysis package was used with DBMiner, but was soon abandoned. WEKA's functionality exceeded DB Miner and was more accommodating to use.

## 4.5 Artificial Data Sets

The dataset is a collection of financial historical primitive attributes.

### Data Attributes [Data Table]:

1. Points [0..9] - Real
2. MovingAverage [0..9] - Real
3. Market - Nominal
4. Sector - Nominal
5. Industry - Nominal

### Class:

2. Trends<sup>10</sup> - Nominal
  - a. Trend line
    - i. Up Trend line
    - ii. Down Trend line
    - iii. Accelerating Up Trend lines
  - b. Reversal Patterns
    - i. Head and Shoulders Reversal Pattern
    - ii. Inverse Head and Shoulders
    - iii. Triple Tops
    - iv. Triple Bottoms
    - v. Double Tops
    - vi. Double Bottoms

The dataset needed to be converted to WEKA's proprietary ARFF format.

The artificial dataset is as follows:

```
@attribute Point0 real
@attribute Point1 real
@attribute Point2 real
@attribute Point3 real
@attribute Point4 real
@attribute Point5 real
@attribute Point6 real
@attribute Point7 real
@attribute Point8 real
@attribute Point9 real
@attribute MovingAverage0 real
@attribute MovingAverage1 real
@attribute MovingAverage2 real
```

---

<sup>10</sup> Murphy: Technical Analysis of Financial Markets

**@attribute MovingAverage3 real**  
**@attribute MovingAverage4 real**  
**@attribute MovingAverage5 real**  
**@attribute MovingAverage6 real**  
**@attribute MovingAverage7 real**  
**@attribute MovingAverage8 real**  
**@attribute MovingAverage9 real**

**@attribute Market** {"nasdaqnm", "nasdaqsc", "nyse"}

**@attribute Sector** {"technology", "services", "financial", "basic materials", "consumer cyclical", "healthcare", "transportation", "consumer non-cyclical", "conglomerates", "energy"}

**@attribute Industry** {"computer services", "software programming", "retail (specialty)", "semiconductors", "communications equipment", "insurance (prop. casualty)", "electronic instruments controls", "communications services", "computer hardware", "metal mining", "audio video equipment", "computer storage devices", "retail (department discount)", "footwear", "biotechnology drugs", "airline", "money center banks", "personal household products", "auto truck manufacturers", "conglomerates", "major drugs", "oil gas operations"}

**@attribute Trend** {"Trend Line Up", "Trend Line Down", "Accelerating Trend Line Up", "Accelerating Trend Line Down", "Fan Principle Up", "Fan Principle Down", "Head and Shoulders", "Head and Shoulders Inverse", "Triple Tops", "Triple Bottoms", "Double Tops", "Double Bottoms"}

**@data**

Additional preprocessing was done at WEKA's run-time, such as filtering attributes adjusting machine learning algorithm parameters. Some of the filtered attributes included the market, sector and industry [which do not help in analyzing the trend]. Other attributes were filtered to produce a combination of attributes. The results from the different combinations were compared to see how well the attribute selection process weighed into the calculations.

## 5. Experiments

WEKA was used to analyze the data. The attribute selection algorithms were used to gauge the importance of the attributes.

Several ARFF artificial datasets were created consisting of combinations of trends and filtered attributes [see the companion cd]. The combinations of classes for these datasets are:

1. All the trends ('CompleteTrends.ARFF').
2. The trend lines ('Trendlines.ARFF').
3. The reversals ('ReversalTrends.ARFF').
4. Only the top reversal trends [i.e. head and shoulders, double top and triple top] ('ReversalTrendsTop.ARFF').
5. Only the bottom reversal trends ('ReversalTrendsBottom.ARFF').

All the classification results are saved as text files in the 'Project' directory on the accompanying CD.

### 5.1 WEKA

#### 5.1.1 Attribute Selection

The two attribute selection algorithms (best first and forward selection) produced identical results. The details confirmed that the points were important in the machine learning process and the 3<sup>rd</sup> and 9<sup>th</sup> moving average [i.e. which are defined as the 20% and 80% in the time range] were additional supplements that were thought to possibly increase the classification accuracy.

#### Attribute Evaluator

"Evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them."

#### Search Method

The forward selection performs a "greedy forward search through the space of attribute subsets." The best first "searches the space of attribute subsets by hillclimbing augmented with a backtracking facility."<sup>11</sup>

---

<sup>11</sup> WEKA

## Attribute Selection:

Attributes: 21

Point0  
Point1  
Point2  
Point3  
Point4  
Point5  
Point6  
Point7  
Point8  
Point9  
MovingAverage0  
MovingAverage1  
MovingAverage2  
MovingAverage3  
MovingAverage4  
MovingAverage5  
MovingAverage6  
MovingAverage7  
MovingAverage8  
MovingAverage9  
Trend

Evaluation mode: evaluate on all training data

=== Attribute Selection on all input data ===

<p><b>Search Method:</b> <b>Forward Selection.</b> Start set: no attributes Merit of best subset found: 0.85</p> <p>Attribute Subset Evaluator (supervised, Class (nominal): 21 Trend): CFS Subset Evaluator</p>	<p><b>Search Method:</b> <b>Best first.</b> Start set: no attributes Search direction: forward Stale search after 5 node expansions Total number of subsets evaluated: 204 Merit of best subset found: 0.85</p> <p>Attribute Subset Evaluator (supervised, Class (nominal): 21 Trend): CFS Subset Evaluator</p>
--	---

Selected attributes: 1,2,3,4,6,7,8,9,10,14,20 : 11

Point0  
Point1  
Point2  
Point3  
Point5  
Point6  
Point7  
Point8  
Point9  
MovingAverage3  
MovingAverage9

## 6. Performance

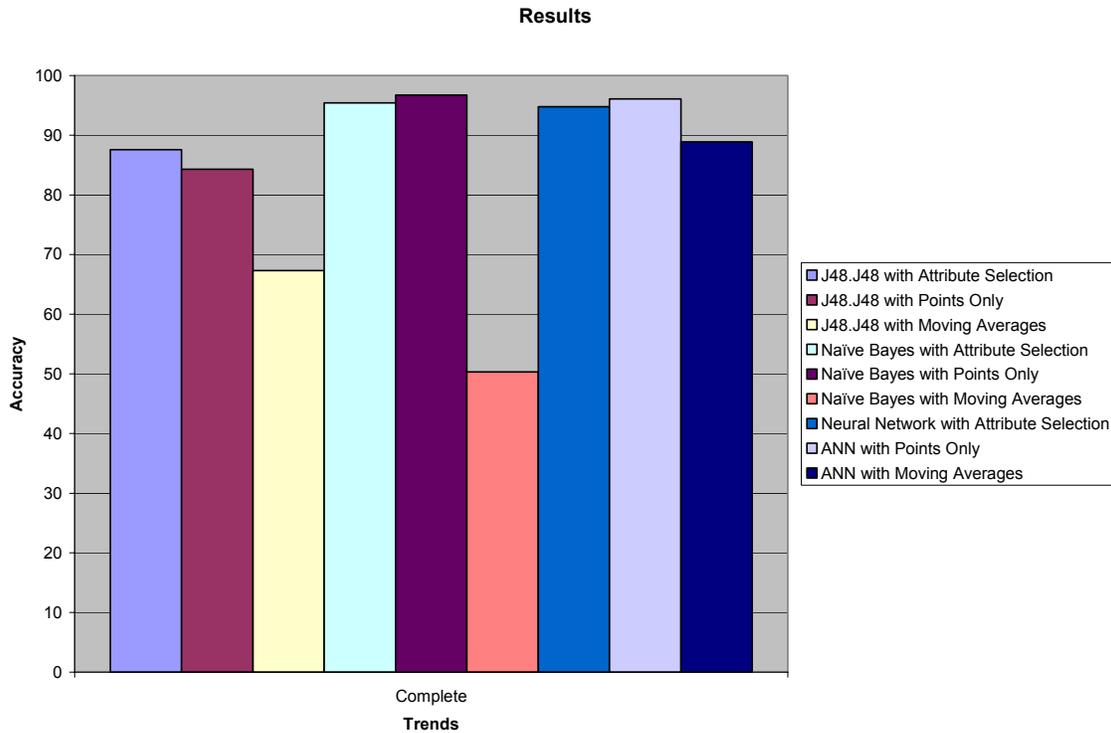
The best performing algorithm with the subset feature selection attributes is Naïve Bayes, slightly beating out the Neural Network's accuracy. Considering that the Naïve Bayes classifier is a simple algorithm relative to the Neural Network this is a surprising result. The Naïve Bayes classifier produced an accuracy of 95.4248% with a mean absolute error of 0.0077. The neural network managed an accuracy of 94.7712% with a mean absolute error of 0.0167. And the C4.5 [or J48] algorithm performed well with 87.5817% accuracy and a mean absolute error of 0.0227.

Every result from these machine learning algorithms is that of a 10-fold cross validation process. Initially, both 10-fold and 66% split [2/3 training and 1/3 test set] were used. The 10-fold cross validation parameter was selected for all the algorithms because it consistently produced higher accuracies with less than or equal error.

Naïve Bayes outperformed the other two algorithms when the moving averages from the subset feature selection [the 3<sup>rd</sup> and the 9<sup>th</sup> moving average] were removed. Under this condition, Naïve Bayes achieved 96.732% accuracy with a mean absolute error of 0.0077 [Classifier-NaiveBayes-Points.txt]. The C4.5 algorithm scored an accuracy of 84.9673% with a mean absolute error of 0.0271. And the Neural Network managed an accuracy of 96.0784% with a mean absolute error of 0.0155, putting it right below the Naïve Bayes results.

The results are only part of the full story. For other combination of attributes, Naïve Bayes did not perform as well as the other two algorithms. With the moving averages only Naïve Bayes performed horribly with an accuracy of 50.3268% and a mean absolute error of 0.0816. The C4.5 algorithm outperformed the Naïve Bayes algorithm with an accuracy of 62.0915% and a mean absolute error of 0.0633. The neural network algorithm performed the best with all this noise that the other two algorithms could not handle. It was able to achieve an accuracy of 88.8889% with a mean absolute error of 0.0462.

Figure 6.1 details the accuracy of each machine learning algorithm [J48/C4.5, Naïve Bayes and the Neural Network] for the complete set of trends against the subset feature selection attributes [i.e. only the points and the 3<sup>rd</sup> and 9<sup>th</sup> moving average].



**Figure 6a**

The next three figures [Figure 6b, 6c, and 6d] demonstrate the accuracy of each algorithm against each type of dataset [i.e. complete, trend lines, reversal, and combinations]. Excluding the results from only the moving averages, Naïve Bayes produced the highest results quickly and efficiently. The Neural Network is up there, but the time to learn the training set, the complexity of the algorithm and the similar results to the Naïve Bayes algorithm make the Neural Network less desirable. However, the more the moving averages are included in the dataset the less accurate the Naïve Bayes results were. In practical applications, the decision of which to use would require additional factors other than the accuracy alone. Each algorithm produced high results.

J48.J48 [C4.5] Accuracy

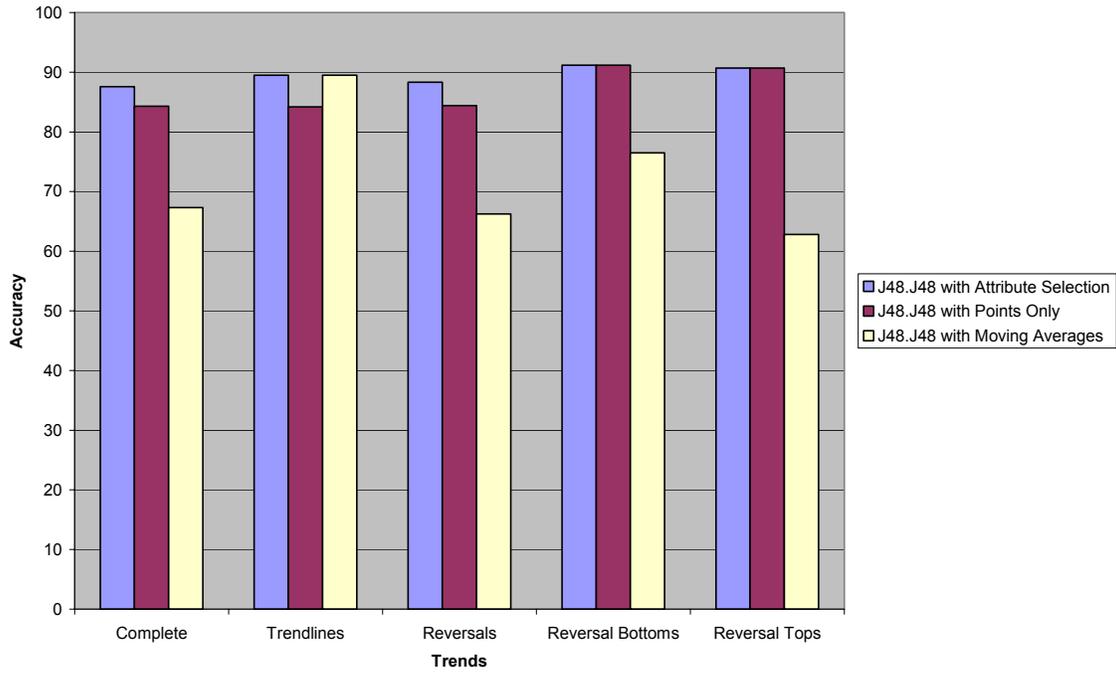


Figure 6b

Naive Bayes

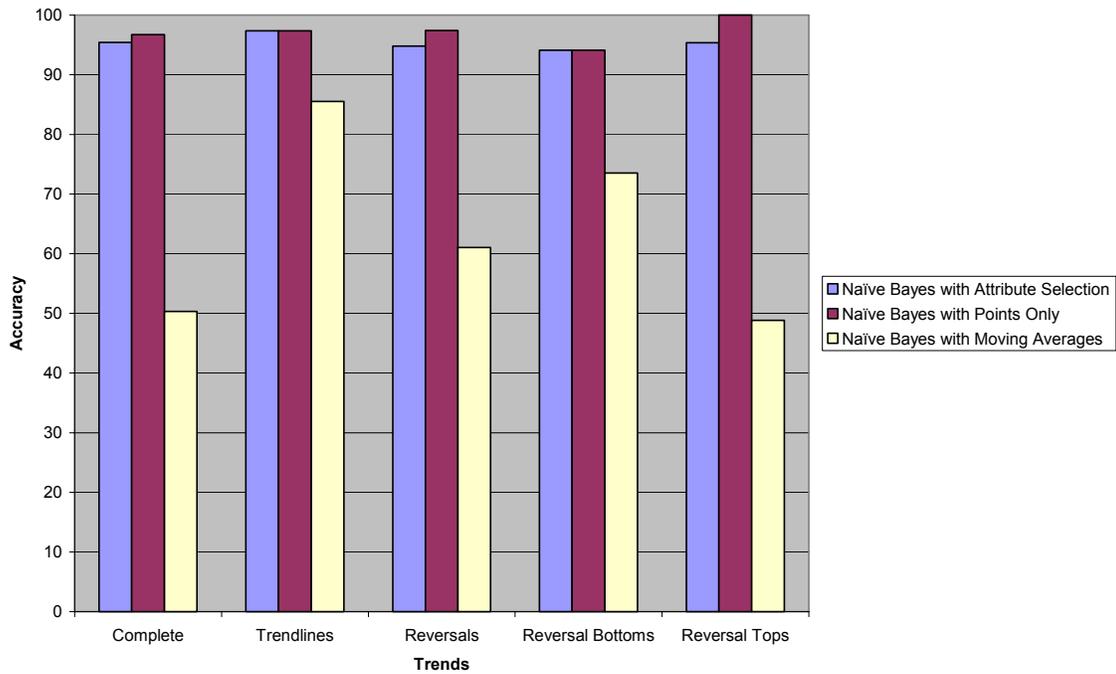


Figure 6c

Neural Network

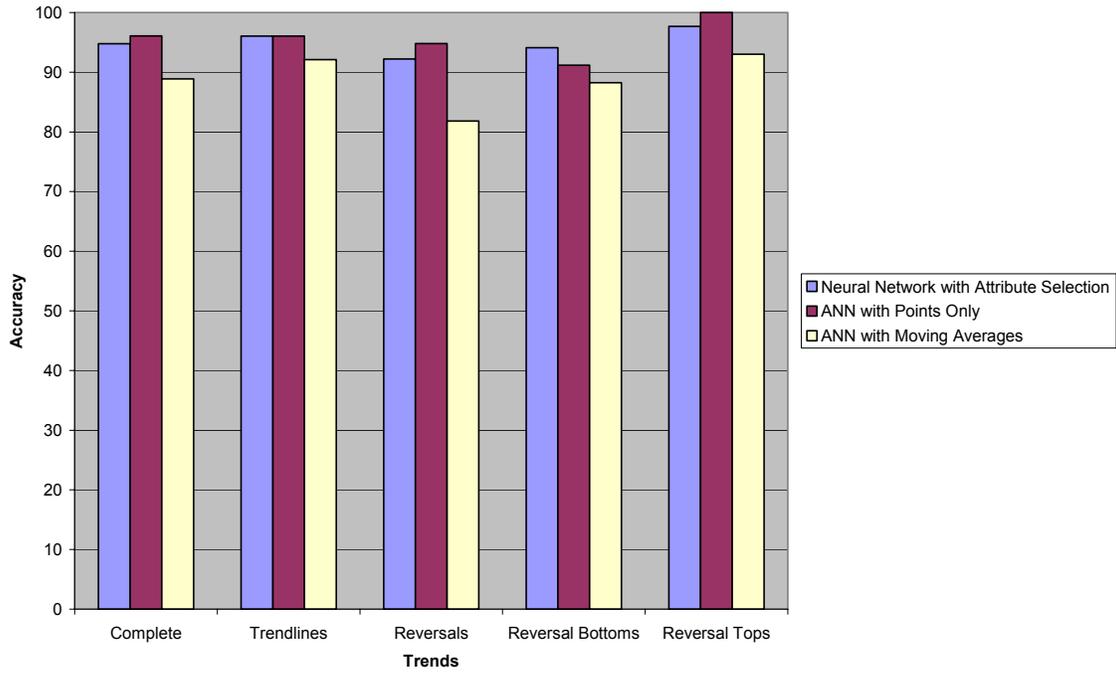


Figure 6d

## 7. Conclusions

The project was successful in breaking down and analyzing trends. The results and accuracy were better than I had expected after poor preliminary results when the dataset contained only 50 instances. The increase in accuracy is associated with the increase in the sample size or the number of instances in the dataset. It is also worthwhile to mention that the dataset has also been cleaned several times more and some erroneous data was eliminated from the dataset.

The complete dataset performed very well with all three data mining algorithms. Each algorithm proved to have strengths associated with the accuracy seen in this experiment. For example, the C4.5 algorithm performed worse than the Naïve Bayes algorithm on the complete dataset with the subset feature selection attributes. However the Achilles heel of the Naïve Bayes algorithm was the moving averages [which are more dependent attributes than the points]. In these cases, the C4.5's accuracy is higher than that of the Naïve Bayes.

The following figures are detailed results for each classifier on the complete dataset. The confusion matrixes graphically point out the true and false positives and negatives for each class. This shows which trends are more difficult to classify. By looking at the confusion matrix it is observed that the accelerating trend line up has been incorrectly confused with the trend line up. And the triple tops and bottoms were sometimes mixed up with the head and shoulders and its inverse.

Profiling the precision, recall and combined f-measure of each algorithm we can see that the Naïve Bayes performed the best against this measuring stick. The J48 had problems correctly distinguishing triple tops and the Neural Network anguished with the double bottoms.

However, these occurrences were seldom and the overall the results were high. The recall [measured as the  $TP / (TP + FN)$ ] and the precision [ $TP / (TP + FP)$ ] and the f-measure were overwhelmingly high [above 90%]. The F-Measure is a popular performance measure in information retrieval<sup>12</sup>. It is equal to:

$$(2 \times \text{recall} \times \text{precision}) / (\text{recall} + \text{precision}) \text{ or } (2 \times TP) / (2 \times TP + FP + FN)$$

With this analysis we now better understand these trends and can describe them. The identification process remains a challenge but one that I look forward to pursue.

---

<sup>12</sup> Witten/Frank 146

```

Classifier - J48.J48-AttributeSelection.txt - Notepad
File Edit Format View Help
J48 pruned tree
-----
Point9 <= 46.242906
|
| Point0 <= 52.529605
| |
| | Point2 <= 63.556212
| | |
| | | Point6 <= 50.993073: Head and Shoulders (13.0)
| | | Point6 > 50.993073
| | | |
| | | | Point3 <= 52.360398: Head and Shoulders (3.0/1.0)
| | | | Point3 > 52.360398: Triple Tops (9.0)
| | | |
| | | | Point2 > 63.556212: Double Tops (16.0)
| | |
| | | Point0 > 52.529605
| | | |
| | | | MovingAverage3 <= 84.887292: Accelerating Trend Line Up (24.0)
| | | | MovingAverage3 > 84.887292
| | | | |
| | | | | Point8 <= 60.478765
| | | | | Point4 <= 39.134469: Accelerating Trend Line Up (3.0/1.0)
| | | | | Point4 > 39.134469: Trend Line Up (24.0)
| | | | |
| | | | | Point8 > 60.478765: Triple Tops (2.0)
| | |
| | | Point9 > 46.242906
| | | |
| | | | MovingAverage9 <= 98.102009
| | | | Point7 <= 43.025927
| | | | |
| | | | | MovingAverage3 <= 87.278203: Triple Bottoms (11.0)
| | | | | MovingAverage3 > 87.278203: Double Bottoms (10.0)
| | | | | Point7 > 43.025927: Head and Shoulders Inverse (13.0/1.0)
| | | | |
| | | | | MovingAverage9 > 98.102009: Trend Line Down (25.0)
|
|
| Number of Leaves : 12
|
| Size of the tree : 23
|
| Time taken to build model: 0.04 seconds
|
| === Stratified cross-validation ===
| === Summary ===
|
| Correctly Classified Instances 134 87.5817 %
| Incorrectly Classified Instances 19 12.4183 %
| Kappa statistic 0.8576
| Mean absolute error 0.0228
| Root mean squared error 0.1426
| Relative absolute error 15.6275 %
| Root relative squared error 52.8393 %
| Total Number of Instances 153
|
| === Detailed Accuracy By Class ===
|
| TP Rate FP Rate Precision Recall F-Measure Class
| 0.92 0.047 0.793 0.92 0.852 Trend Line Up
| 1 0.023 0.893 1 0.943 Trend Line Down
| 0.885 0.008 0.958 0.885 0.92 Accelerating Trend Line Up
| 0 0 0 0 0 Accelerating Trend Line Down
| 0 0 0 0 0 Fan Principle Up
| 0 0 0 0 0 Fan Principle Down
| 0.8 0 1 0.8 0.889 Head and Shoulders
| 0.833 0.014 0.833 0.833 0.833 Head and Shoulders Inverse
| 0.636 0.035 0.583 0.636 0.609 Triple Tops
| 0.917 0 1 0.917 0.957 Triple Bottoms
| 0.882 0.007 0.938 0.882 0.909 Double Tops
| 0.8 0.007 0.889 0.8 0.842 Double Bottoms
|
| === Confusion Matrix ===
|
| a b c d e f g h i j k l <-- classified as
| 23 0 1 0 0 0 0 0 1 0 0 0 | a = Trend Line Up
| 0 25 0 0 0 0 0 0 0 0 0 0 | b = Trend Line Down
| 3 0 23 0 0 0 0 0 0 0 0 0 | c = Accelerating Trend Line Up
| 0 0 0 0 0 0 0 0 0 0 0 0 | d = Accelerating Trend Line Down
| 0 0 0 0 0 0 0 0 0 0 0 0 | e = Fan Principle Up
| 0 0 0 0 0 0 0 0 0 0 0 0 | f = Fan Principle Down
| 0 0 0 0 0 0 12 0 3 0 0 0 | g = Head and Shoulders
| 0 1 0 0 0 0 0 10 0 0 0 1 | h = Head and Shoulders Inverse
| 2 1 0 0 0 0 0 0 7 0 1 0 | i = Triple Tops
| 0 0 0 0 0 0 0 1 0 11 0 0 | j = Triple Bottoms
| 1 0 0 0 0 0 0 0 1 0 15 0 | k = Double Tops
| 0 1 0 0 0 0 0 1 0 0 0 8 | l = Double Bottoms

```

Figure 7a

```

Classifier-NaiveBayes-AttributeSelection.txt - Notepad
File Edit Format View Help
=== Summary ===
Correctly Classified Instances      146          95.4248 %
Incorrectly Classified Instances     7            4.5752 %
Kappa statistic                    0.9477
Mean absolute error                 0.0077
Root mean squared error             0.0778
Relative absolute error             5.2656 %
Root relative squared error        28.831 %
Total Number of Instances          153

=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall  F-Measure  Class
0.96     0.016    0.923     0.96    0.941      Trend Line Up
0.96     0        1         0.96    0.98       Trend Line Down
0.923    0        1         0.923   0.96       Accelerating Trend Line Up
0        0        0         0        0          Accelerating Trend Line Down
0        0        0         0        0          Fan Principle Up
0        0        0         0        0          Fan Principle Down
1        0        1         1        1          Head and Shoulders
1        0.007    0.923    1        0.96       Head and Shoulders Inverse
0.909    0        1         0.909   0.952      Triple Tops
0.833    0        1         0.833   0.909      Triple Bottoms
1        0.015    0.895    1        0.944      Double Tops
1        0.014    0.833    1        0.909      Double Bottoms

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  k  l  <-- classified as
24  0  0  0  0  0  0  0  0  0  1  0 | a = Trend Line Up
 0 24  0  0  0  0  0  0  0  0  0  1 | b = Trend Line Down
 2  0 24  0  0  0  0  0  0  0  0  0 | c = Accelerating Trend Line Up
 0  0  0  0  0  0  0  0  0  0  0  0 | d = Accelerating Trend Line Down
 0  0  0  0  0  0  0  0  0  0  0  0 | e = Fan Principle Up
 0  0  0  0  0  0  0  0  0  0  0  0 | f = Fan Principle Down
 0  0  0  0  0  0 15  0  0  0  0  0 | g = Head and Shoulders
 0  0  0  0  0  0  0 12  0  0  0  0 | h = Head and Shoulders Inverse
 0  0  0  0  0  0  0  0 10  0  1  0 | i = Triple Tops
 0  0  0  0  0  0  0  0  1 10  0  1 | j = Triple Bottoms
 0  0  0  0  0  0  0  0  0  0 17  0 | k = Double Tops
 0  0  0  0  0  0  0  0  0  0  0 10 | l = Double Bottoms

```

Figure 7b

```

Classifier-NeuralNetwork-AttributeSelection.txt - Notepad
File Edit Format View Help
=== Summary ===
Correctly Classified Instances      145          94.7712 %
Incorrectly Classified Instances    8            5.2288 %
Kappa statistic                    0.9401
Mean absolute error                 0.0167
Root mean squared error             0.0794
Relative absolute error             11.4379 %
Root relative squared error         29.406 %
Total Number of Instances          153

=== Detailed Accuracy By Class ===
TP Rate  FP Rate  Precision  Recall  F-Measure  Class
0.96     0.008    0.96       0.96    0.96       Trend Line Up
1        0.008    0.962     1       0.98       Trend Line Down
0.962    0.016    0.926     0.962  0.943     Accelerating Trend Line Up
0        0        0         0       0         Accelerating Trend Line Down
0        0        0         0       0         Fan Principle Up
0        0        0         0       0         Fan Principle Down
0.933    0        1         0.933  0.966     Head and Shoulders
1        0.014    0.857     1       0.923     Head and Shoulders Inverse
1        0        1         1       1         Triple Tops
0.833    0        1         0.833  0.909     Triple Bottoms
1        0.007    0.944     1       0.971     Double Tops
0.7      0.007    0.875     0.7    0.778     Double Bottoms

=== Confusion Matrix ===
 a  b  c  d  e  f  g  h  i  j  k  l  <-- classified as
24  0  1  0  0  0  0  0  0  0  0  0  a = Trend Line Up
 0 25  0  0  0  0  0  0  0  0  0  0  b = Trend Line Down
 1  0 25  0  0  0  0  0  0  0  0  0  c = Accelerating Trend Line Up
 0  0  0  0  0  0  0  0  0  0  0  0  d = Accelerating Trend Line Down
 0  0  0  0  0  0  0  0  0  0  0  0  e = Fan Principle Up
 0  0  0  0  0  0  0  0  0  0  0  0  f = Fan Principle Down
 0  0  0  0  0  0 14  0  0  0  1  0  g = Head and Shoulders
 0  0  0  0  0  0  0 12  0  0  0  0  h = Head and Shoulders Inverse
 0  0  0  0  0  0  0  0 11  0  0  0  i = Triple Tops
 0  1  0  0  0  0  0  0  0 10  0  1  j = Triple Bottoms
 0  0  0  0  0  0  0  0  0  0 17  0  k = Double Tops
 0  0  1  0  0  0  0  0  2  0  0  0  l = Double Bottoms

```

Figure 7c

## 7.1 Interesting Patterns

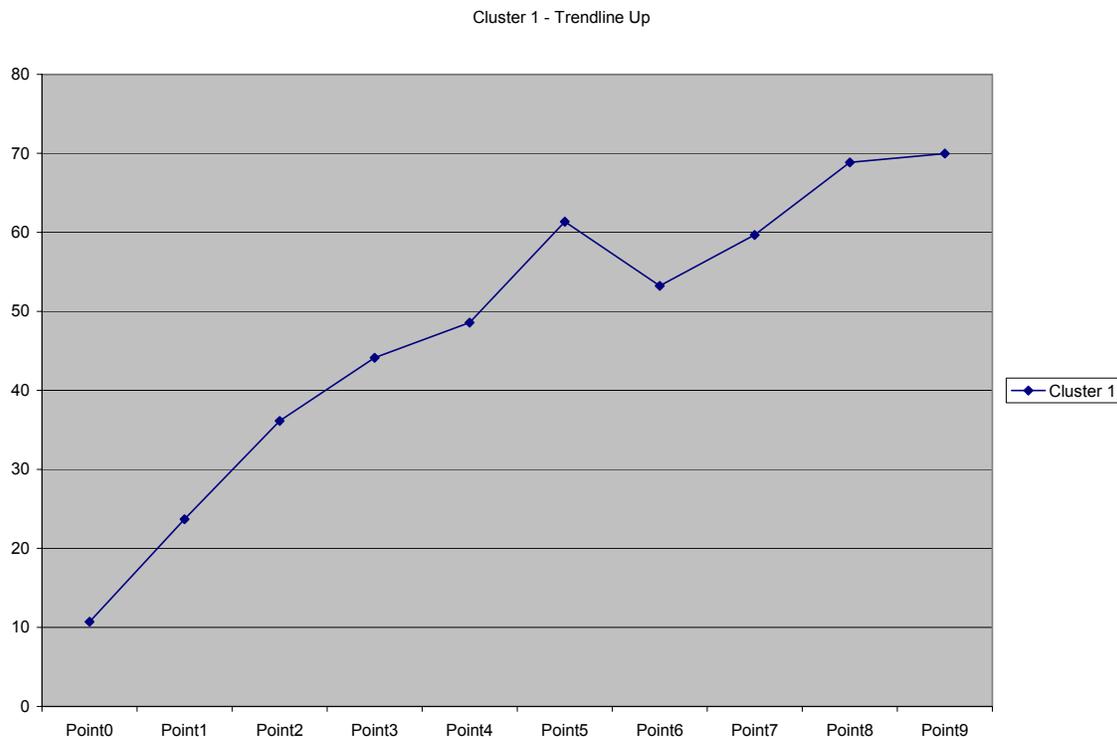
### 7.1.1 Cluster Analysis

WEKA uses the EM [expectation maximization] algorithm for clustering. The cluster analysis is run on the complete dataset with the subset feature selection attributes filtered in the artificial dataset, which contains all the points and the 3<sup>rd</sup> and 9<sup>th</sup> moving average. The clustering algorithm produced nine clusters which is [coincidentally] the number of trends in the complete dataset. This was interesting because clustering has no prior knowledge of the supervised classes and for it to come up with 9 clusters is either an odd coincidence or highly correlated match with the supervised classes. The results were exported to Microsoft Excel where I plotted the cluster medians for each cluster.

It turned out the clustering algorithm produced seven similar trends to the supervised classes. The trend line up, trend line down [x2], Accelerating trend line up, the head and shoulders and its inverse, the double top and the triple bottom were all distinguished. The double bottom was less distinct but recognizable. The triple top was left out. In practice the triple top is the least common of the reversal patterns and is often mistaken for the head and shoulders.<sup>13</sup> The ninth cluster appears to be a second trend line down. The two trend lines differ in that one is convex and the other is concave.

Overall, the clustering algorithm provided interesting results. It would be interesting to run the algorithm on a larger and more complete dataset and see the results.

The following figures are the plotted clusters.



**Figure 7.1a**

<sup>13</sup> Murphy 118

Cluster 2 - Head and Shoulders

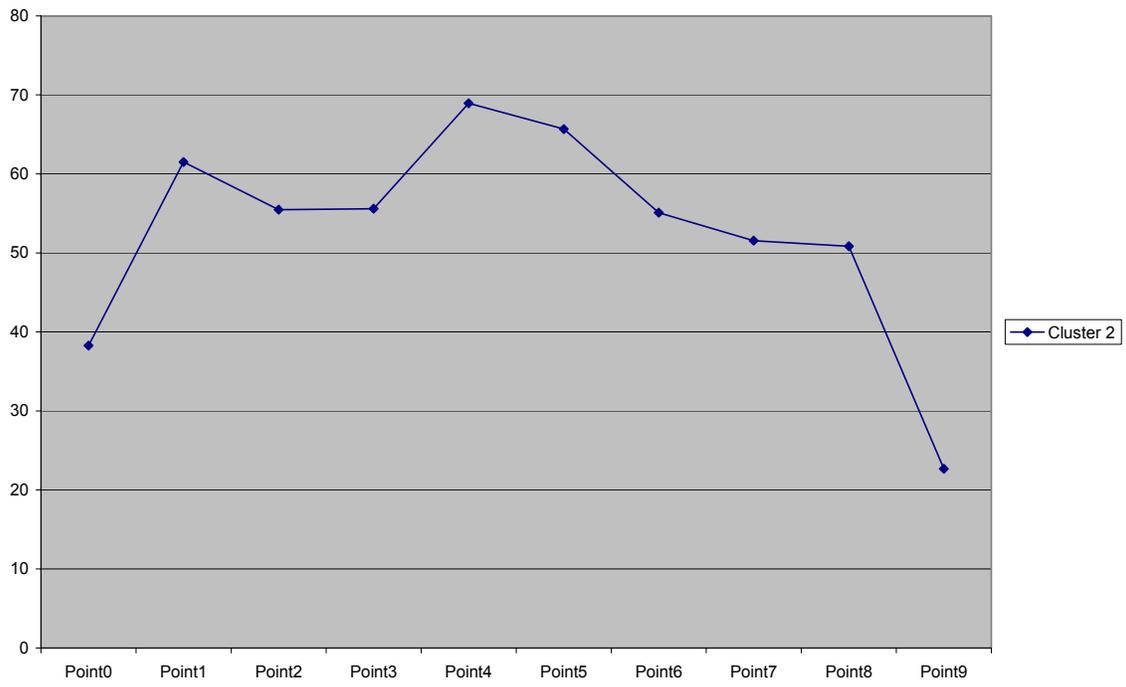


Figure 7.1b

Cluster 3 - Double Top

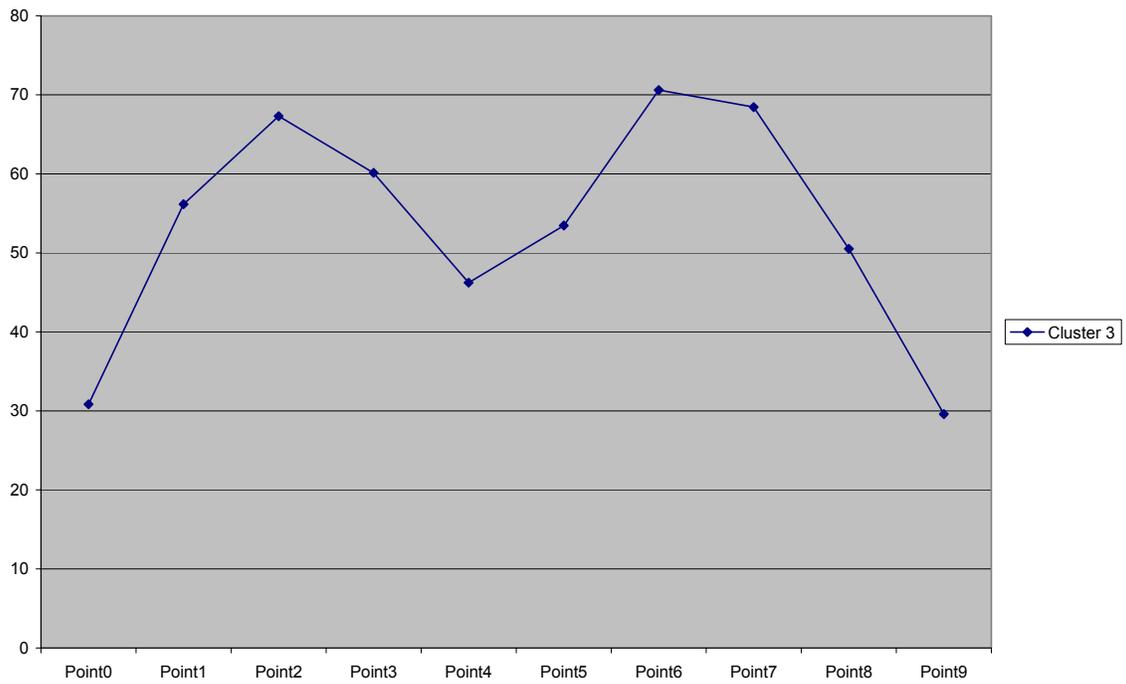


Figure 7.1c

Cluster 4 - De-Accelerating Down Trendline

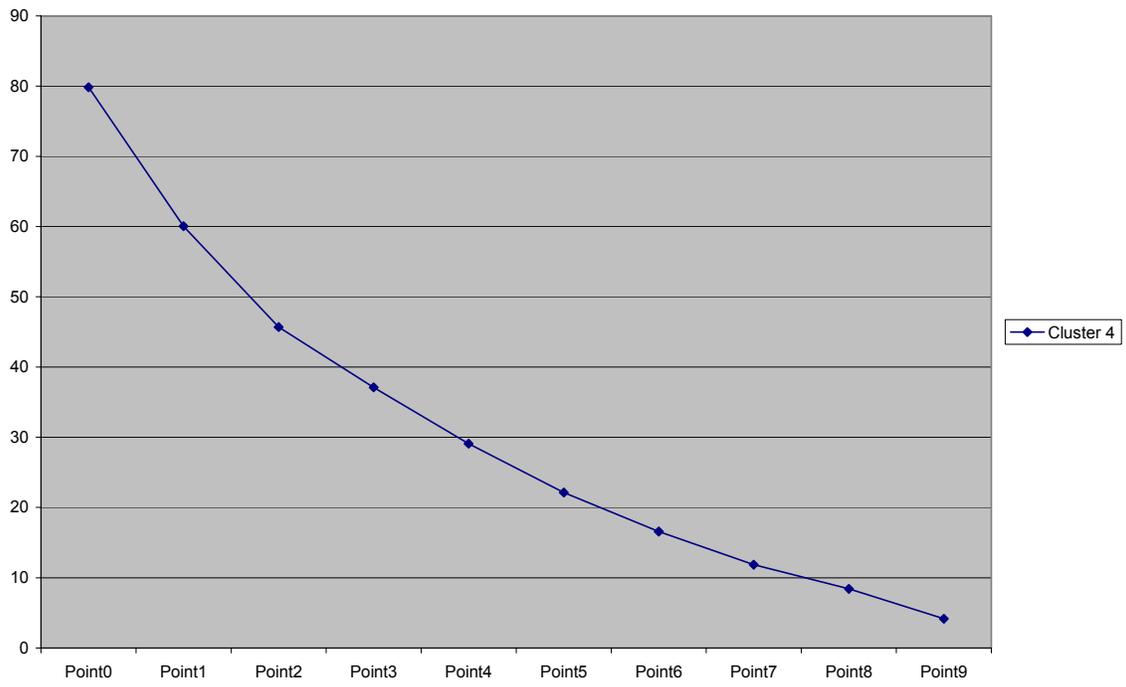


Figure 7.1d

Cluster 5 - Trendline Down

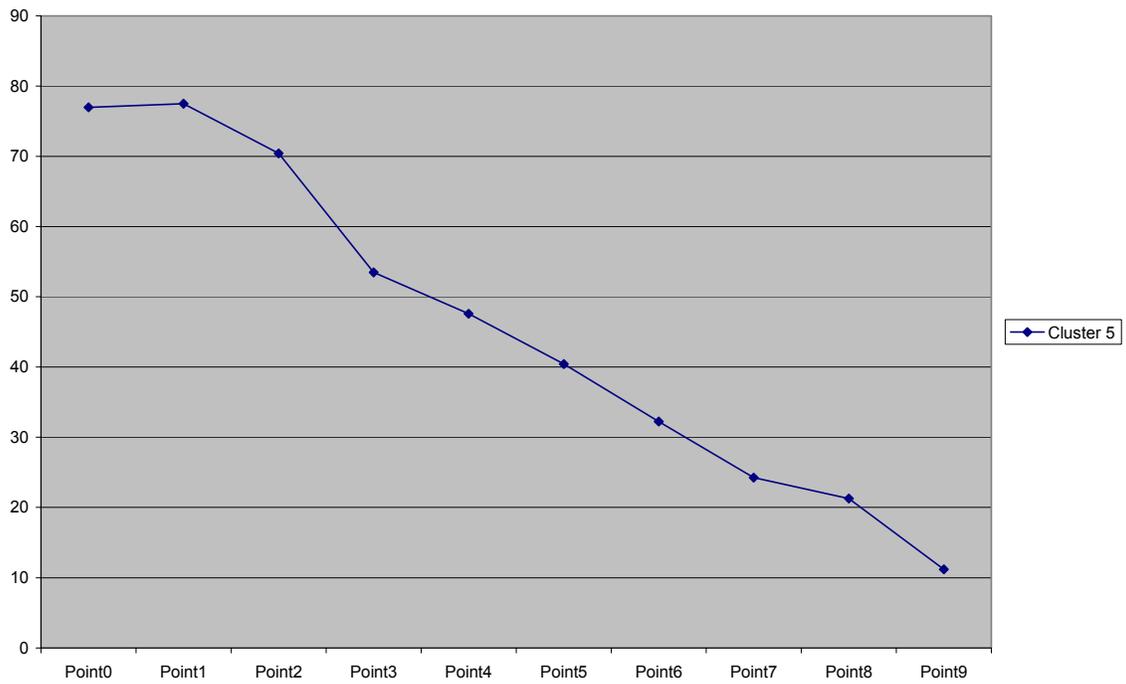


Figure 7.1e

Cluster 6 - Accelerating Trendline Up

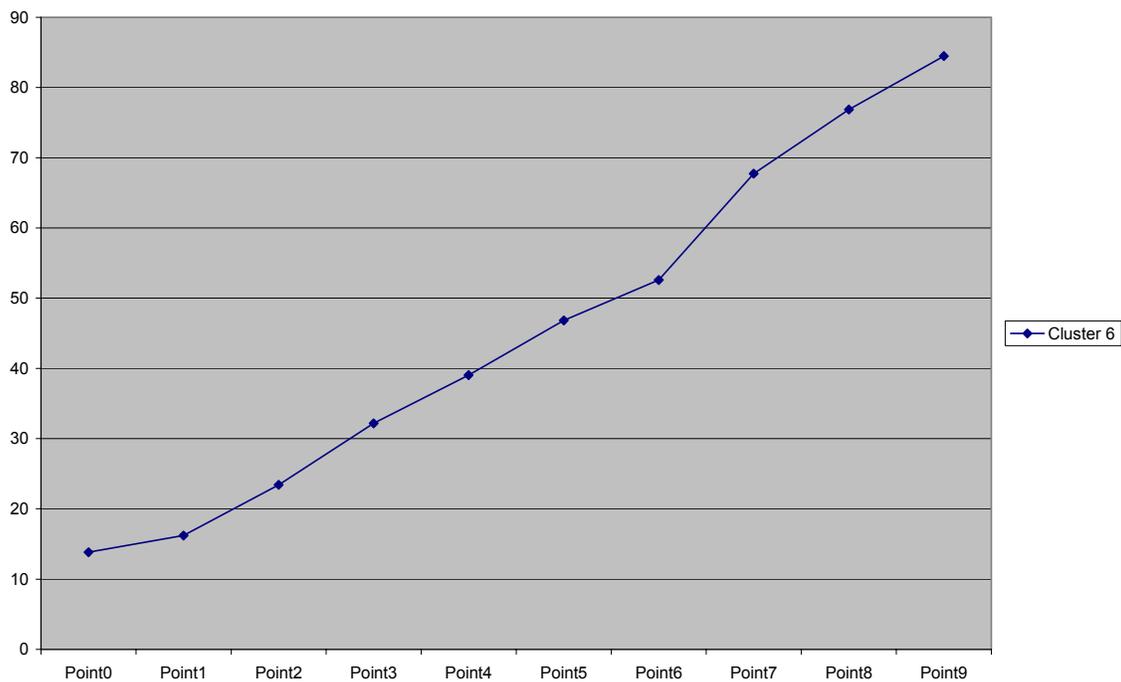


Figure 7.1f

Cluster 7 - Head and Shoulders Inverse

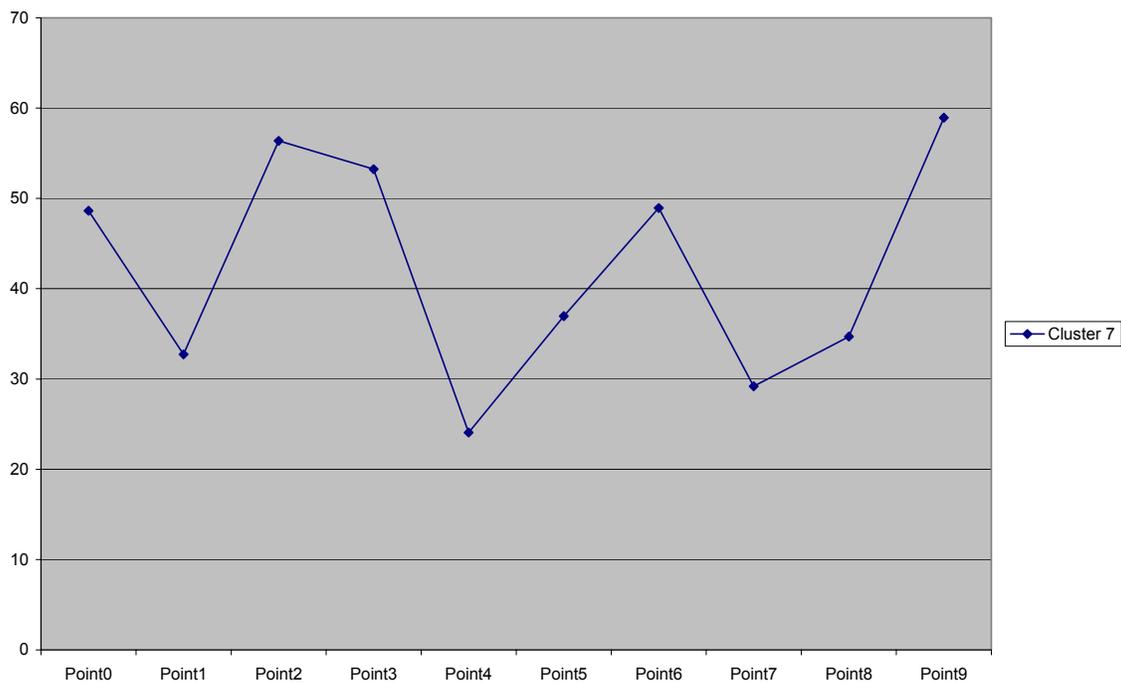


Figure 7.1g

Cluster 8 - Triple Bottom

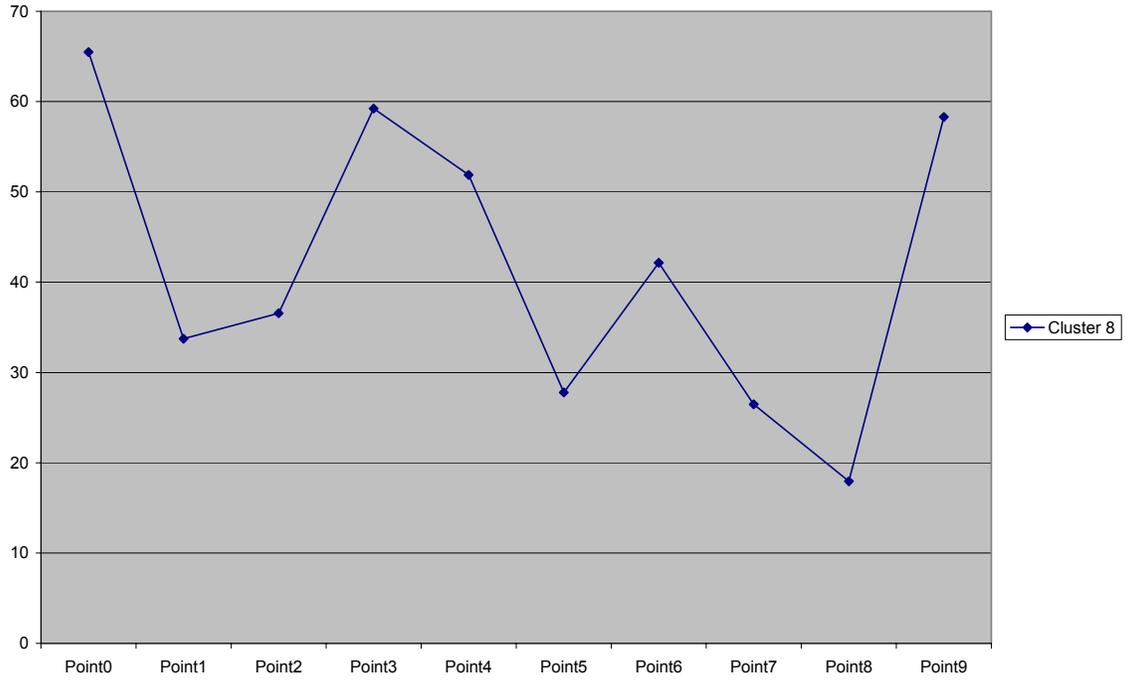


Figure 7.1h

Cluster 9 - Double Bottom [Attempted]

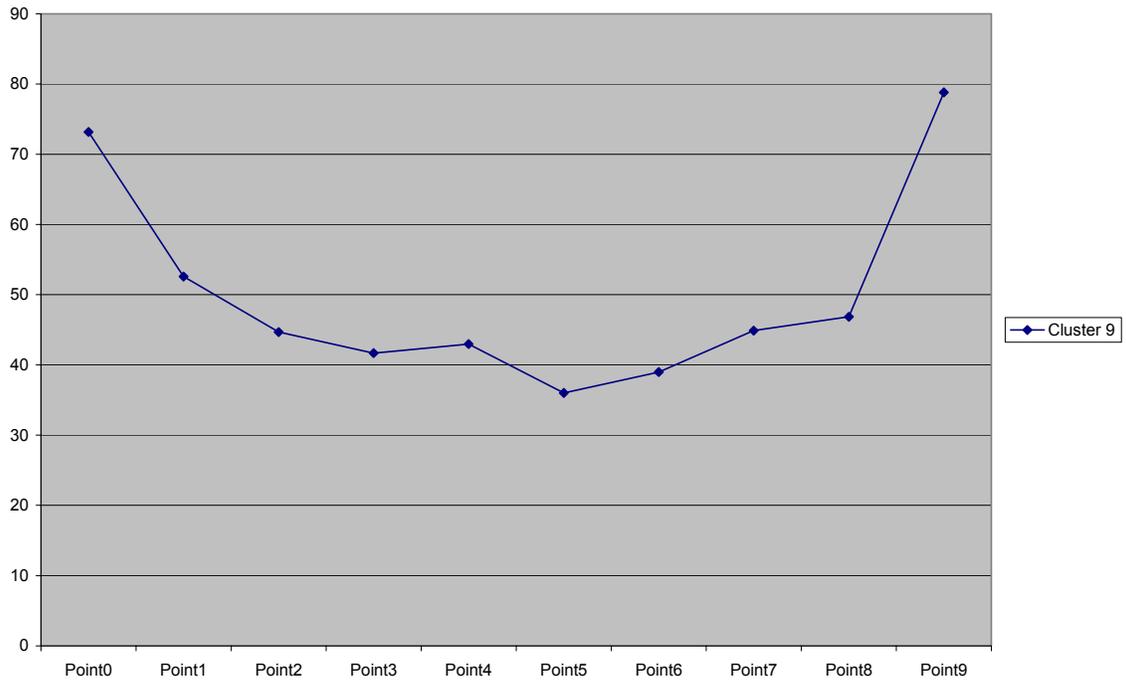


Figure 7.1i

## **8. References**

1. Witten, Ian H. and Eibe Frank. (1999) Data Mining Practical Machine Learning Tools and Techniques. 146
2. Murphy, John. (1999) Technical Analysis of the Financial Markets.
3. StockCharts.com. <http://stockcharts.com>