Genetic Programming for Security Analysis

A research study to evaluate the performance of decision trees evolved using the Genetic Programming algorithm to solve the security analysis problem based on technical and fundamental financial indicators

by Stuart Reid

Student number: 10026942

Email: stuartgordonreid@gmail.com

Department of Computer Science

Faculty of Engineering, Built Environment and IT

University of Pretoria

1 June 2013

Abstract

In this research study I evolved a population of decision trees using the Genetic Programming algorithm to solve the Security Analysis problem for 62 of the 65 Information Technology stocks listed on the S&P500 stock market index for the year 2011. That is, I evolved decision trees capable of classifying those stocks according to whether they should be bought or sold short. These decision trees made use of technical analysis indicators and fundamental analysis indicators generated and extracted from market data during the year 2010 for the dynamic production of investment rules.

Table of Contents

Introduction and Background

Introduction to Security Analysis

Security analysis is the art and science of analysing publically traded financial instruments called securities. Security analysis can be seen as a real world classification problem. For portfolio managers with an existing investment portfolio and liquid capital, securities can be classified according to whether the investment manager would buy the security, hold onto the security or sell the security. For hedge fund managers with liquid capital, securities can be classified according to whether the hedge fund manager would buy the security or whether the hedge fund manager would short sell[1] the security.

There exist two long-standing schools of thought regarding Security Analysis. The first and oldest school of thought is referred to as Technical Analysis. Technical Analysis is a method of evaluating securities by analyzing statistics generated by market activity, such as past prices and volume. Technical analysts do not attempt to measure a security's intrinsic value, but instead use charts and other tools to identify patterns that can suggest future activity. This is as opposed to the slightly newer, but also old school of thought referred to as Fundamental Analysis. Fundamental Analysis is a method of evaluating a security that entails attempting to measure its intrinsic value by examining related economic, financial and other qualitative and quantitative factors. Since the late 1920's the debate has been raging as to which one of these two techniques is better able to predict sustainable future return on investments. In the interim period a number of new age forms of security analysis have begun to emerge including, but not limited to, quantitative analysis and sentiment analysis.

In this report I will be describing the application of the Genetic Programming algorithm to the construction of security analysis decision trees using technical indicators. A colleague of mine whom I worked with during this assignment will write a similar report except describing the application of the Genetic Programming algorithm to the construction of security analysis decision trees using fundamental indicators. At the end of this report there will be a summary of his findings adjacent with mine and a comment will be made as to whether security analysis decision trees based on Technical or Fundamental indicators performed better.

Introduction to Genetic Programming and Decision Trees

[1] The selling of a security that the seller does not own, or any sale that is completed by the delivery of a security borrowed by the seller. Short sellers assume that they will be able to buy the stock at a lower amount than the price at which they sold short.

The Genetic Programming algorithm is a specialization of a Genetic Algorithm. Genetic Algorithms are population based, this means that they operate within a population consisting of many different individuals. Each individual is represented by a unique genotype (usually encoded as a vector). Genetic Algorithms model the process of genetic evolution through a number of operators inspired by what exists in nature. These operators include the selection operator which models survival of the fittest, the crossover operator which models sexual reproduction and a mutation operator which models genetic mutations that occur randomly to individuals in the population. These operators, when combined, produce what computer scientists refer to as a Genetic Algorithm. The difference then between a Genetic Algorithm and the Genetic Programming Algorithm is the way in which individual genotypes are represented. In Genetic Algorithms genotypes are represented either as Strings or as Vectors, in Genetic Programming these genotypes are represented using tree data structures. The pseudocode for a Genetic Programming algorithm is shown below in green:

```
Let t = 0 be the generation counter;
Create and initialize an n_x dimensional population: P(0);
repeat
        Evaluate the fitness, f(x_i), of each individual, x_i, in the population, P(t);
        Perform crossover to produce offspring
        Perform mutation on offspring;
        Select population P(t+1) of new generation;
        Advance to the new generation, i.e. t=t+1;
until stopping condition is true
```

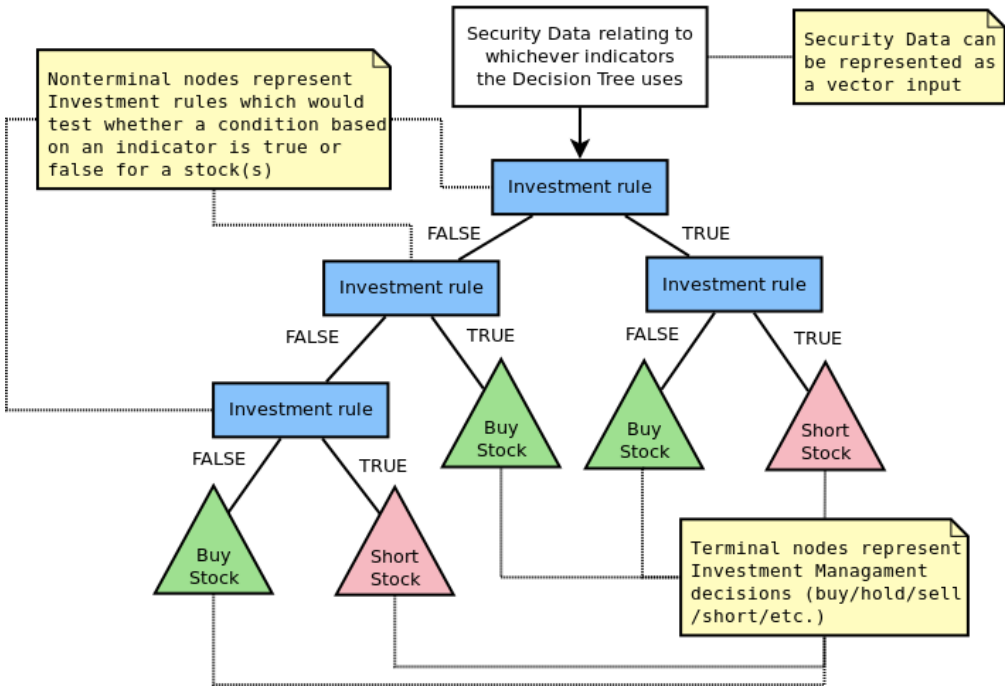
Decision Trees are either used to determine the statistical probabilities of outcomes or possible courses of action given some set of input data. In this research study we deal with only the latter type of decision tree. Decision trees are constructed (or evolved in the case of Genetic Programming) from a set of nonterminal functional nodes and a set of terminal decision nodes. Decision trees are evolved according to some internal grammar. A grammar for a decision tree would define a set of rules for constructing that tree such that the constructed tree is guaranteed to be correct and meaningful.

Using Genetic Programming for Security Analysis

A strategy for Security Analysis, regardless of whether it uses technical or fundamental indicators, will consist of a number of rules for making investment decisions. That strategy can be represented as a decision tree where the terminal nodes represent investment decisions and

the functional nodes represent rules based either on technical or fundamental indicators. Due to this fact, many existing investment strategies are represented in the form of decision trees and decision trees have subsequently become a major component of any Finance curriculum[1].

Visual representation of a theorerical Security Analysis Decision Tree



Previous attempts at automating the production of decision trees for security analysis using the classical Artificial Intelligence C4.5 algorithm have yielded promising results[2]. However, the use of more modern Computational Intelligence algorithms, like the Genetic Programming algorithm, have not been applied to this problem before. This research study, and a related research study done by a colleague of mine, are an initial attempt at trying to evolve strategies for security analysis using Genetic Programming.

Problem Statement

In this research study I aim to evolve a population of decision trees using the Genetic Programming algorithm to solve the Security Analysis problem for 62 of the 65 Information Technology stocks listed on the S&P500 stock market index for the year 2011[3]. That is, I aim to

[1] See the Investopedia article on 'Using Decision Trees in Finance' - http://www.investopedia.com/articles/financial-theory/11/decisions-trees-finance.asp

[2] See the work done by Rakhmawati and Suryani, 'Decision for buying and selling stock with decision tree algorithm' - http://www.its.ac.id/personal/files/pub/4546-erma-is-ICTSbaru.pdf

[3] See appendix A for the full list of companies

evolve decision trees capable of classifying those stocks according to whether they should be bought or sold short. These decision trees must make use of technical analysis indicators generated from market data during the year 2010 for the dynamic production of investment rules (functional nodes). The decision trees should also remain as small as possible without compromising the return on investment realized through the classifications. In addition to this the following research aims will be considered during this research study:

1. Compare the average performance of the investment decisions produced by the decision trees against the average sector return for the Information Technology stocks listed on the S&P500.
2. Investigate how the performance of the above mentioned approach varies over time, specifically at the end of each financial quarter during 2011.
3. Provide a comment on the nature of Security Analysis and the effect that changes in the algorithms control parameters had on performance. More specifically I aim to provide comment on:
    a. The average performance of the decision trees
    b. The average sizes of the decision trees
    c. The average heterogeneity of the decision trees and
    d. The indicators used most frequently in the decision trees
4. Furthermore, I hope to compare the performance of the Genetic Programming algorithm using fundamental indicators (related research study) against the Genetic Programming algorithm using technical indicators (this research study).

Design and Implementation

Approach Taken

During this research study a lot of time was dedicated to the design phase. This was done to make sure that the implementation and simulations would run smoothly and meet all of the aims laid out at the beginning of this research study. Our approach loosely followed five higher level phases: Data preparation, Data validation, Algorithmic Design and Implementation and

Algorithmic Testing. I will now go into more detail as to what exactly each one of these phases entailed.

Data preparation

There was a very large amount of data needed during this research study. First and foremost, we needed historical data relating to the stocks in question. This data came from two sources namely the Morningstar API[1] and the Google finance API[2]. The Morningstar API was used to obtain all the indicators and key ratios pertaining to fundamental analysis and the Google finance API was used to obtain all historical price data including daily highs, lows, closing prices and volumes traded.

Once all of this data was downloaded in its raw format, the fundamental indicators being used for the fundamental analysis portion of this research study (related research) needed to be extracted for the correct time periods. In addition to this the technical indicators being used in the technical analysis portion of this research study (this research) needed to be calculated over the training time period. A full list of the technical indicators I calculated can be found on the following page. All of this data was then outputted to individual text files for each one of the 62 stocks being analysed.

In total across all of the stocks analyzed, 2604 technical indicators were calculated and used in this research study. That number does not include the additional 2604 fundamental indicators that were either calculated or extracted from the data. The implementation of the algorithm was designed in such a way that the correct indicators (technical or fundamental) would be read in from the file and parsed through to the tree for security analysis.

Full List of Technical Indicators

In total 42 indicators were calculated (with the different time frames), which is the same number of indicators that was used in the related Fundamental analysis research.

Indicator	**Meaning of the indicator**	**Time frames**
Price Movement (%)	This indicator calculated the percentage price movement from the date that the securities were either bought or shorted (3 Jan 2011) until the current 2011 quarter being tested.	The prices movement indicator for 1, 7, 14, 30, 60 and 90 days were calculated prior to the buy / short date

[1] Web front-end to the API for AAPL key company financials and ratios from 2003 to 2012: http://financials.morningstar.com/ratios/r.html?t=AAPL&region=USA&culture=en-us

[2] Web front-end to the API for AAPL historical stock prices: https://www.google.com/finance/historical?q=NASDAQ%3AAAPL&ei=DFSrUdDHF4WZwQPa8wE

Simple Moving Average (SMA)	A simple, or arithmetic, moving average that is calculated by adding the closing price of the security for a number of time periods and then dividing this total by the number of time periods. Short-term averages respond quickly to changes in the price of the underlying, while long-term averages are slow to react.	The SMA indicator for 7, 14, 30, 60, 90 and 180 days was calculated prior to the buy / short date
Volume Price Trend (VPT)	A technical indicator consisting of a cumulative volume line that adds or subtracts a multiple of the percentage change in share price trend and current volume, depending upon their upward or downward movements	The VPT indicator for 7, 14, 30, 60, 90 and 180 days was calculated prior to the buy / short date
Momentum	The rate of acceleration of a security's price or volume. The idea of momentum in securities is that their price is more likely to keep moving in the same direction than to change directions. In technical analysis, momentum is considered an oscillator and is used to help identify trendlines.	The Momentum indicator for 7, 14, 30, 60 and 90 days was calculated prior to the buy / short date
Rate of Change (ROC)	The speed at which a variable changes over a specific period of time. Rate of change is often used when speaking about momentum, and it can generally be expressed as a ratio between a change in one variable relative to a corresponding change in another. Graphically, the rate of change is represented by the slope of a line.	The ROC indicator for 7, 14, 30, 60 and 90 days was calculated prior to the buy / short date
Aroon	A technical indicator used for identifying trends in an underlying security and the likelihood that the trends will reverse. It is made up of two lines: one line is called "Aroon up", which measures the strength of the uptrend, and the other line is called "Aroon down", which measures the downtrend. The indicator reports the time it is taking for the price to reach, from a starting point, the highest and lowest points over a given time period, each reported as a percentage of total time.	The Aroon was calculated up 25 days and down 25 days prior to the buy / short date
Force	The force index (FI) is an indicator used in technical analysis to illustrate how strong the actual buying or selling pressure is. High positive values mean there is a strong rising trend, and low values signify a strong downward trend. The FI is calculated by multiplying the difference between the last and previous closing prices by the volume of the commodity, yielding a momentum scaled by the volume. The strength of the force is determined by a larger price change or by a larger volume.	The Force indicator 1, 7, 14, 30, 60 and 90 days was calculated prior to the buy / short date
True Range	A measure of volatility introduced by Welles Wilder	The True Range

	in his book: New Concepts in Technical Trading Systems. The true range indicator is the greatest of the following: ● Current high less the current low. ● The absolute value of the current high less the previous close. ● The absolute value of the current low less the previous close.	indicator was calculated for 7, 14, 30, 60, 90 and 180 days prior to the buy / short date

Data validation

After the data was prepared and extracted I completed a 'sanity check' on the numbers to make sure that they were being correctly calculated and that there were no outlier cases that would affect the performance of the algorithm later on. During this stage the issue of missing or incomplete data was addressed and any missing values were supplemented with the average for that value across the other 62 securities being analyzed.

Algorithmic Design and Implementation

Prior to implementation the different components of the Genetic Programming algorithm were designed. Most of the operators used in the implementation of this research study were the 'default' implementations and the algorithm was not optimized to fit the Security Analysis problem. What follows is a discussion of the following algorithmic components: the overall algorithmic framework, fitness function, individuals in the population, crossover operator, selection operator and the mutation operator.

Algorithmic Framework

The algorithm was developed in a Java framework. The implementation included a number of different packages and fully utilized Object Oriented design principles including inheritance and polymorphism. The packages were:

Package	**Function of package + <*classes in package*>**
gpfinance	This package was the 'root' package. It included all the logic required to initialize the framework, create new genetic programming algorithms with specific parameters and run simulations and output the results to files *<GPFinance, Simulator, U (Universal reusable methods)>*
gpfinance - algorithm	This package contained the class definition of the Genetic Programming algorithm and the class definition of the Individuals constituting the population

	<GP, Individual>
gpfinance - algorithm - interfaces	This package contained reusable interfaces to the different Genetic Algorithm operators that are used by the Genetic Programming algorithm *<CrossoverStrategy, MutationStrategy, SelectionStrategy>*
gpfinance - algorithm - strategies	This package contains the implementations of specific Genetic Algorithm operators and strategies as used by the Genetic Programming algorithm. *<InitializationStrategy, MuLamdaSelectionStrategy, RandomSelectionStrategy, RankBasedSelectionStrategy, SexualCrossoverStrategy, SexualRootCrossoverStrategy, StochasticMuLamdaSelectionStrategy, TreeMutationStrategy>*
gpfinance - datatypes	This package contained a number of datatypes as used by the algorithm and in the individuals in the population (trees) *<Decision, Fitness, FitnessData, Fund, Indicator, Security, Tech>*
gpfinance - tree	This package contained the logic responsible for creating decision trees using the data types in the package explained above as well as the logic for evaluating the performance of those trees *<CriteriaNode, DecisionNode, DecisionTree, Node>*

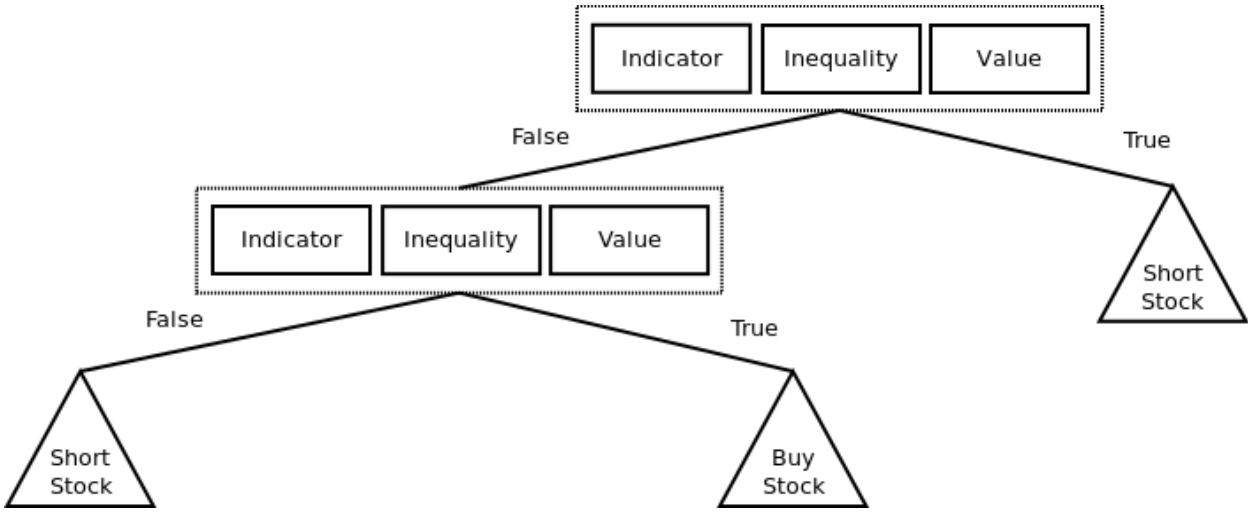
Individuals in the population

Individuals in the population are represented by their genotype. In the case of Genetic Programming, the genome uses a tree based representation. In this research study specifically that tree was used to represent a decision tree where the terminal nodes represent investment decisions and the functional nodes represent investment rules (see diagram in the section of this report entitled ‘Using Genetic Programming for Security Analysis’).

Investment rules are constructed using three objects: an indicator, an inequality and a value. The indicator in the investment rule is selected from the list of available technical indicators. The inequality can either be greater than or less than which strictly forces the decision tree to be a binary tree. As a part of the data preparation phase, the mean values and standard deviations for each indicator were calculated across the sample set of securities. These are used to construct a gaussian distribution from which the value in the investment rule is sampled.

Investment decisions can either be to buy the security or to short sell the security. Shorting involves selling a security that the investor does not own and promising ‘delivery’ of that security at some predetermined date in the future. This is done in the hope that the price of that security will fall between the date that the security is sold and the day that is must be delivered. Therein allowing the investor to buy the security at a lower price than he sold it for and net a profit.

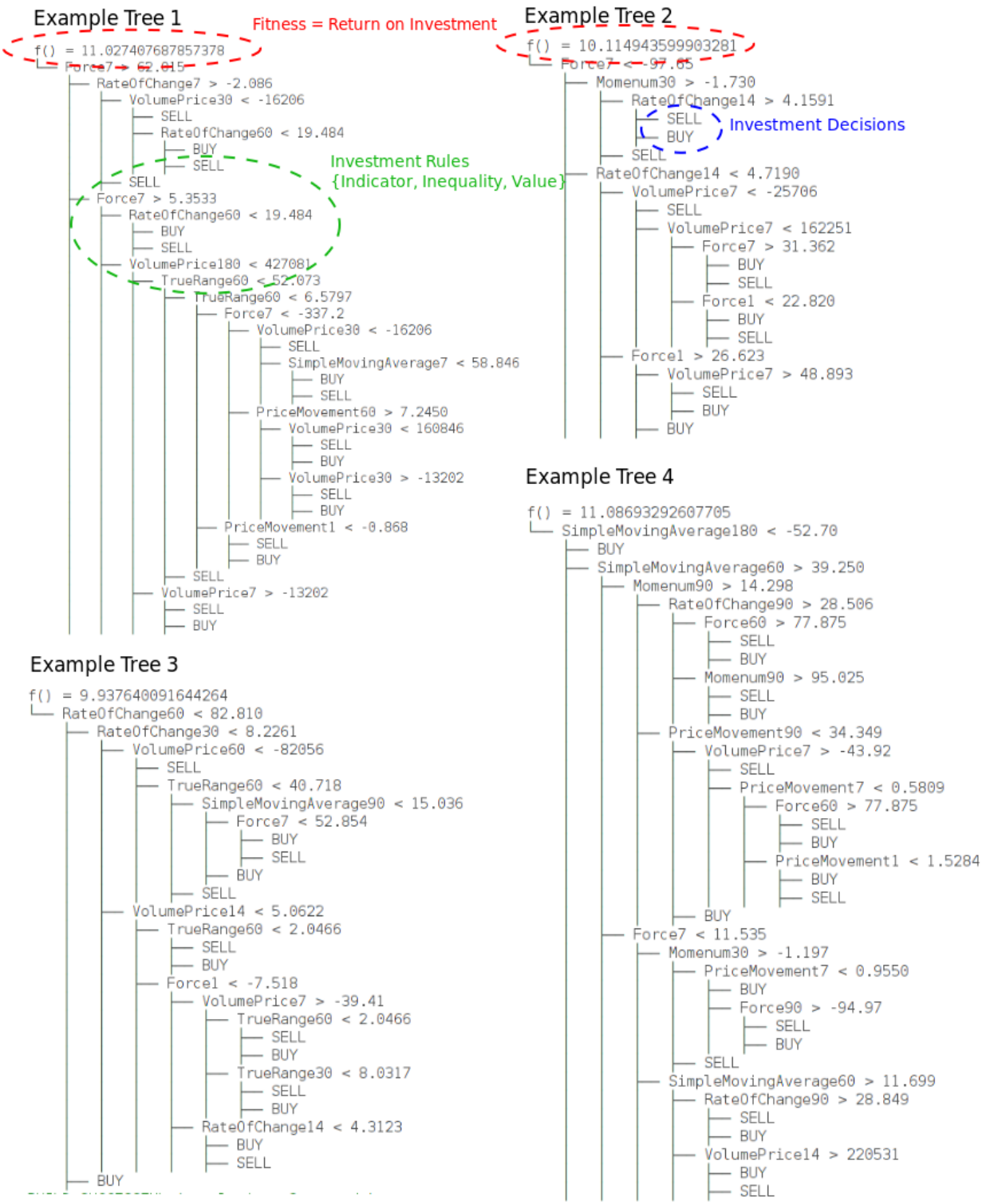
An example theoretical decision tree



On the next two pages I have shown some example decision trees as evolved by the Genetic Programming algorithm in this research study.

As a side note, short selling is traditionally regarded as quite risky because unlike when buying a security the downside risk is unlimited. In other words, a security could rise infinitely and the investor who short sold that stock could make unlimited losses. On the other hand, a security can only lose 100% of it's value, meaning that an investor who bought it can only lose up to the price he paid for that security. This is assuming no leverage has been employed by the investor in executing his or her trades.

Example decision trees produced by the implementation



Formal Grammar to describe the decision trees

The below grammar shown in green is a formal grammar representation in EBNF (Extended Backus Naur Form) of the strategy explained in the aforementioned pages for constructing valid security analysis decision trees that use technical indicators.

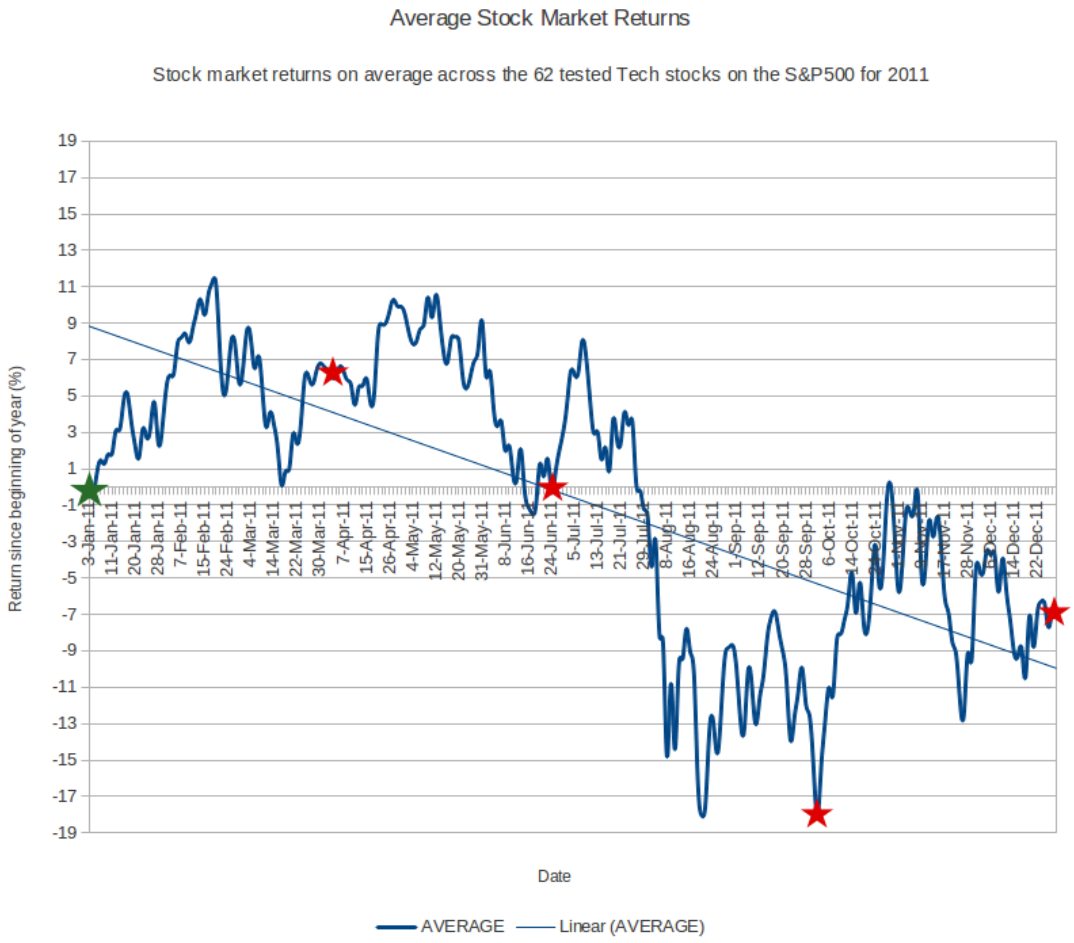
digit = “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9” ;
float = digit, “.”, digit ;
value = float ;
indicator = “1 Day % price movement” | “7 Day % price movement” | “14 Day % price movement” | “30 Day % price movement” | “60 Day % price movement” | “90 Day % price movement” | “7 Day Simple Moving Average” | “14 Day Simple Moving Average” | “30 Day Simple Moving Average” | “60 Day Simple Moving Average” | “90 Day Simple Moving Average” | “180 Day Simple Moving Average” | “7 Day Volume Price Trend” | “14 Day Volume Price Trend” | “30 Day Volume Price Trend” | “60 Day Volume Price Trend” | “90 Day Volume Price Trend”” | “180 Day Volume Price Trend” | “7 Day Momentum” | “14 Day Momentum” | “30 Day Momentum” | “60 Day Momentum” | “90 Day Momentum” | “7 Day % Rate of change” | “14 Day % Rate of change” | “30 Day % Rate of change” | “60 Day % Rate of change” | “90 Day % Rate of change” | “Aroon up 25” | “Aroon down 25” | “1 day force index” | “7 day force index” | “14 day force index” | “30 day force index” | “60 day force index” | “90 day force index” | “7 Day True Range” | “14 Day True Range” | “30 Day True Range” | “60 Day True Range” | “90 Day True Range” | “180 Day True Range” ;
inequality = “>” | “<” ;
decision = “BUY” | “SHORT” ;
criteria = indicator, inequality, value ;
node = decision | node, node, criteria ;
tree = node ;

Fitness function

A number of fitness functions were employed in this research study. The selection of the fitness function by the algorithm was dependent on the duration for which the securities were being bought or sold short for. During this research study we simulated the returns on investments accrued by our best decision trees at the end of each quarter in the year 2011.

The aim of this was to see how our decision tree's investment decisions would fair against the average movement in the Information Technology sector of the S&P500 as well as to see how well the performance of our investment decisions did over time.

The period of 2011 was a very challenging year for Technology stocks globally and many professional investors lost money during this period. The graph below shows the average performance of the 62 Information Technology stocks we covered in our research. The green star represents the time that our investment decisions were made and the red stars indicate the ends of each of the fiscal quarters in 2011 where our fitnesses were calculated.



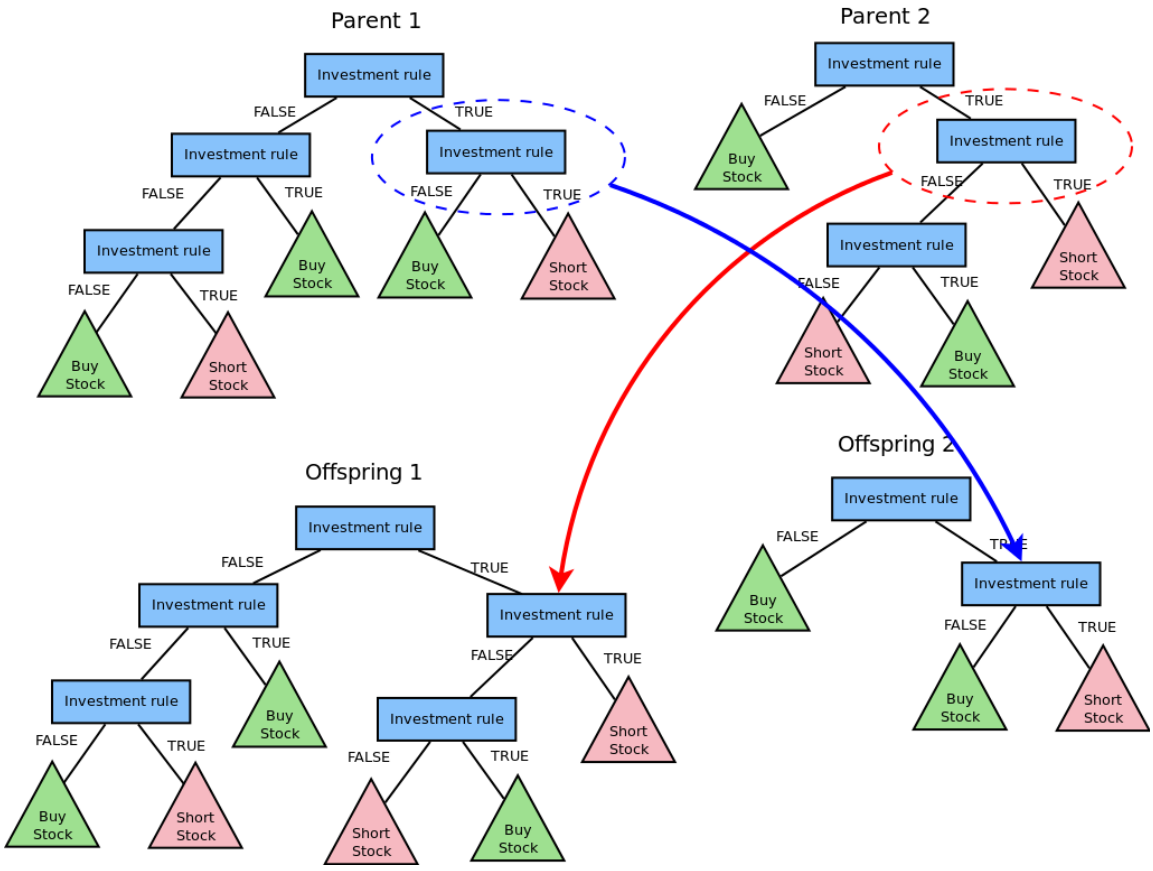
The fitness functions implemented in this research study favoured smaller decision trees over larger ones but not at the expense of return on investment. This was done by gradually increasing the tax on the fitness of the decision trees as they grew larger in size.

Crossover operator

Crossover is the operator that mimics sexual reproduction in nature. There are three broad types of crossover: asexual crossover, sexual crossover and multi-recombination crossover. In asexual crossover one parent produces one offspring, in sexual crossover two parents are used to produce two offspring and lastly in multi-recombination crossover, multiple parents are used to produce one or more offspring.

In the implementation of this research study sexual crossover was used to produce new offspring. In Genetic Programming this involves swapping sub-trees of the individual with another individual. This process is displayed in the diagram below.

Visual representation of sexual crossover of two Security Analysis Decision Trees



Selection operator

Two separate selection strategies have been used in the implementation of this research study namely the rank based selection strategy and a slightly modified mu-lambda selection strategy. In selecting individuals from the population to perform crossover on, the rank based selection strategy is used. This selection strategy was chosen because of its low selective pressure. In selecting the next population to constitute the next generation of individuals, the mu-lambda selection strategy is used.[1]

In both instances the stronger, fitter individuals will be selected for crossover and are more likely to survive onto the next generation. This has the net effect of evolving increasingly better security analysis decision trees, however, a problem with the Genetic Programming algorithm is that the individual decision trees can overfit the backtesting period, otherwise known as the training data. Two counteractive measures employed in this study is the selection of a large sample set of securities as well as a higher-than-usual mutation probability for investment rules.

Mutation operator(s)

Mutation in Genetic Algorithms introduces new genetic material into an existing individual. This is done to diversify the genetic characteristics of the population and promote greater 'exploration'. Since individuals in the Genetic Programming algorithm are expressed using tree based genotypes, the mutations that can occur to an individual relate to changing the structure of the tree in some way. These mutations include mutation of functional and terminal nodes, swap mutations, grow mutations, trunc mutations and gaussian mutations. More specifically, in this research study the following mutations were implemented:

1. Grow mutation - An investment decision is randomly selected and replaced with a randomly constructed investment rule.
2. Trunc mutation - An investment rule is randomly selected and replaced with a randomly generated investment decision.
3. Investment rule / Functional node mutations - An investment rule is randomly selected and mutated. Investment rules can have their indicator, inequality or their value mutated.
    a. Indicator mutation - The indicator is replaced with a randomly selected indicator
    b. Inequality mutation - The inequality is flipped from > to < or from < to >
    c. Value mutation - A new value is sampled from the indicators gaussian distribution

Algorithmic Testing

In addition to the above algorithmic framework a separate testing framework was implemented that tested the different operators across a variety of control parameters. Through observation

[1] For more information about these strategies see - A Engelbrecht, An Introduction to Computational Intelligence, 2nd edition

and investigation the following set of control parameters were decided upon as being sufficient for experimentation purposes. Most importantly these value remained the same for the experiments done using Fundamental Analysis indicators as well as Technical Analysis Indicators:

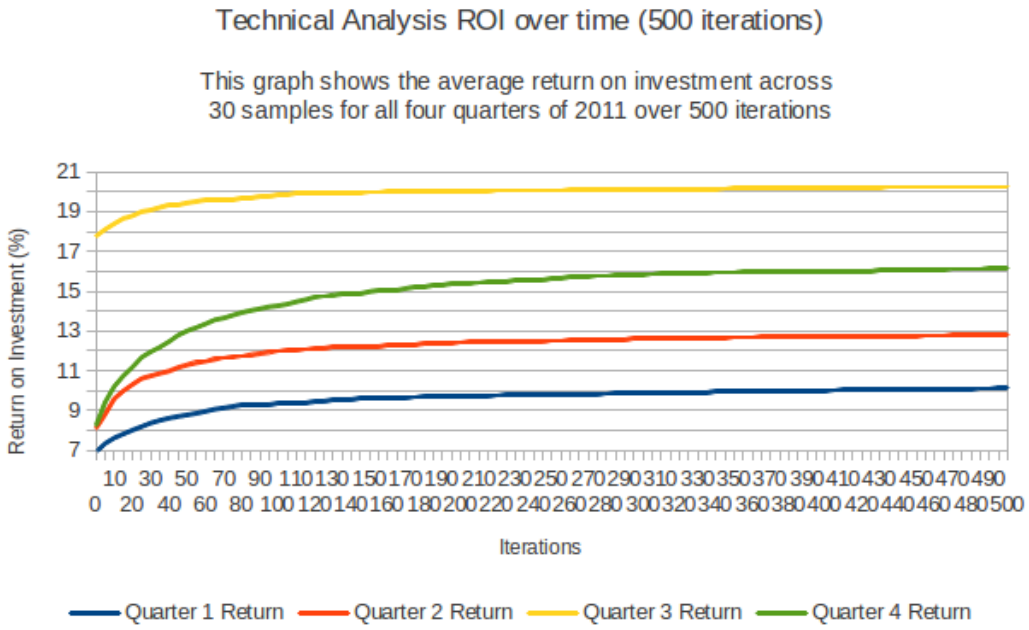
Control Parameters

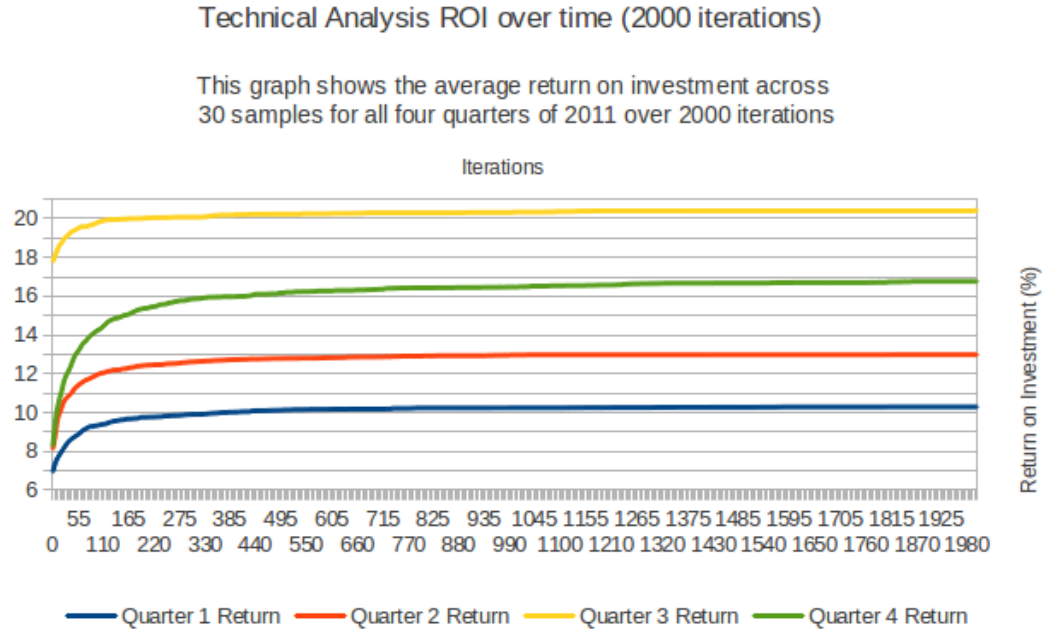
Parameter	**Value(s)**
# Generations	2000
Population size	100
Fitness reward size contribution	0.2
Stochastic selection restart rate	{initial: 0.6, final: 0.02}
Crossover probability	{initial: 0.75, final: 0.25}
Mutation rates	grow = {initial: 0.9, final: 0.4} trunc = {initial: 0.0, final: 0.4} indicator = {initial: 0.6, final: 0.1} leaf = {initial: 0.5, final: 0.1} inequality = {initial: 0.6, final: 0.1} gauss = {initial: 0.9, final: 0.4}

Results for Decision Trees using Technical Indicators

Returns on Investment over time

The graph below shows the improvement to the return on investment over time (iterations). The majority of improvement occurs within the first 150 iterations however the population does not stagnate after that point as improvements are still being made albeit slowly.





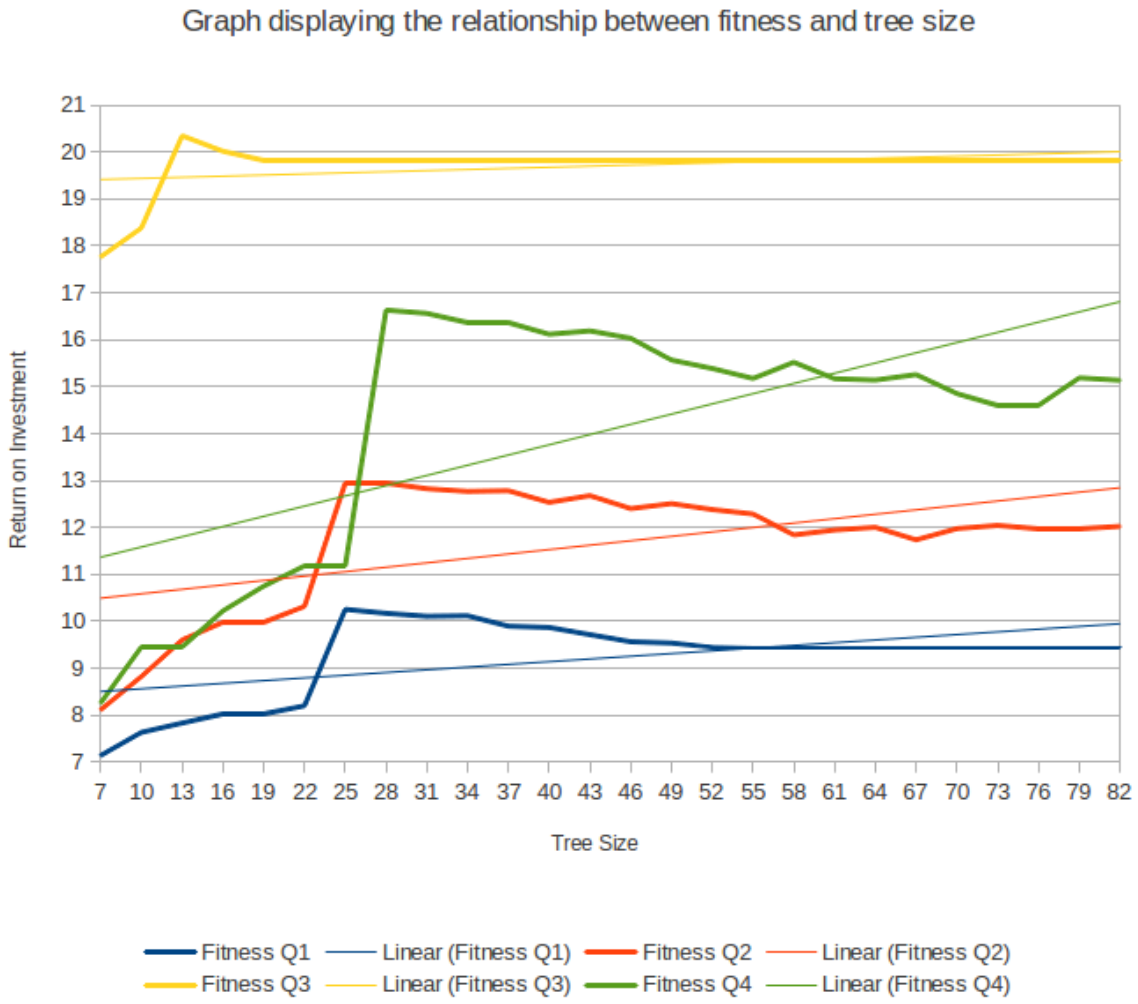
Comment on the Security Analysis problem

Provide a comment on the nature of Security Analysis and the effect that changes in the algorithms control parameters had on performance.

The average performance of the decision trees relative to size

The graph below illustrates what the average return on investment was across the 30 samples simulated for the different tree sizes for each one of the quarters. It is an interesting graph because it shows at which point an increase in the size of the tree stopped having a positive effect on the performance of the genetic programming algorithm. In other words it shows the optimal size of the decision tree for each quarter.

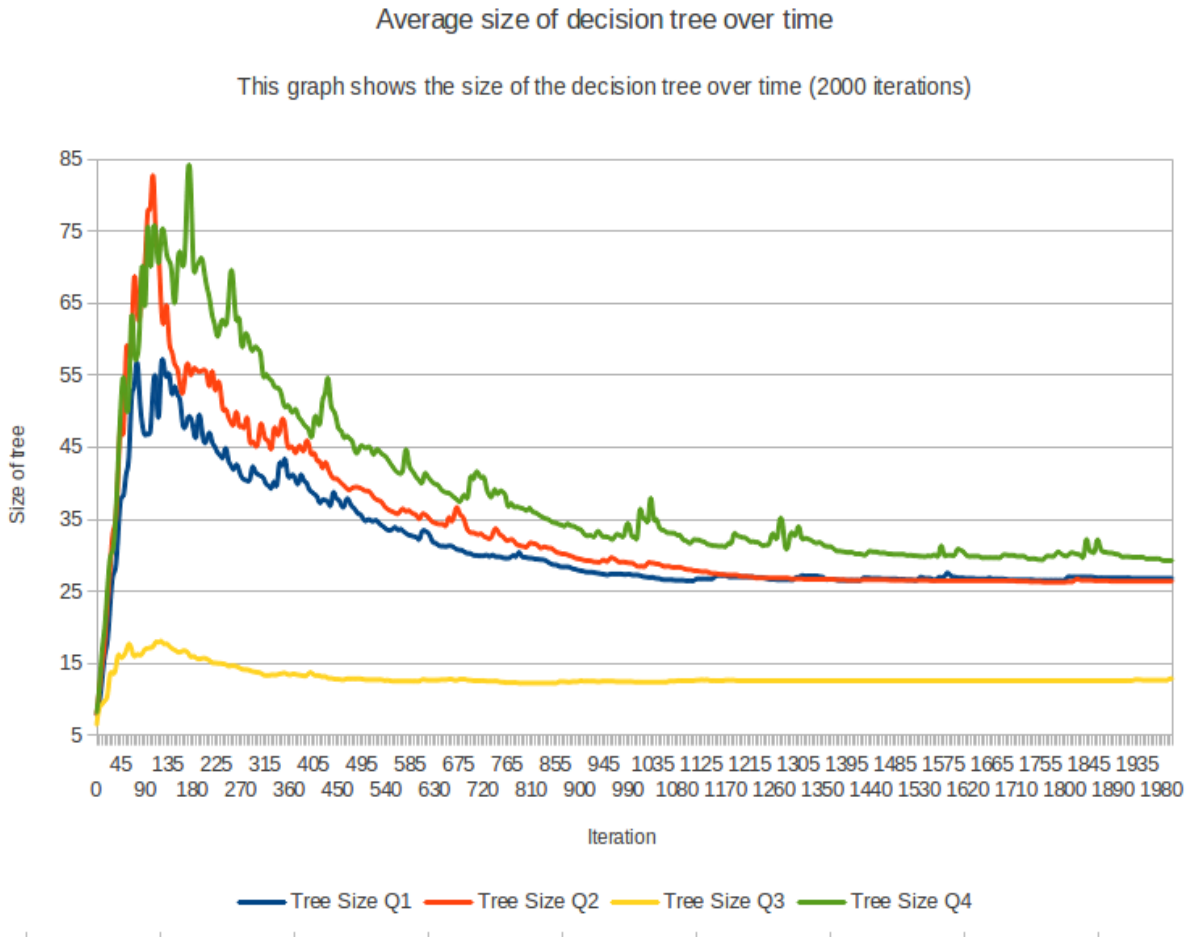
[1]



[1] Note that this graph was produced using 'buckets'. As in different buckets into which the tree size could fall were created and the fitnesses of the decision trees in that bucket were averaged.

The average sizes of the decision trees

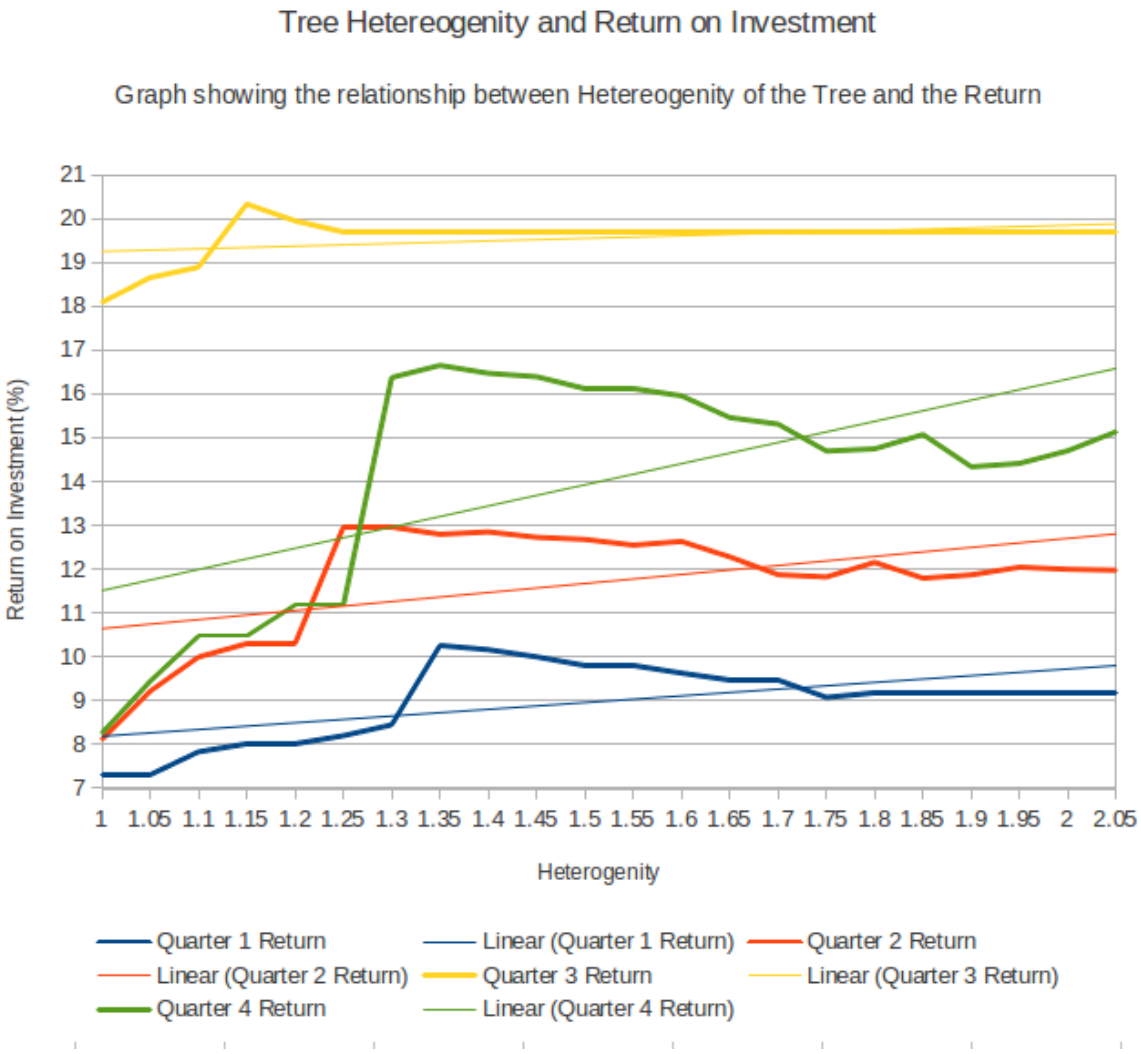
On average the size of the decision trees grew quite extensively for the first 500 iterations before beginning to shrink as the tax for being a large tree grew. This has the effect of producing decision trees of a manageable size that also perform well. The graph below shows the average tree size per iteration across all 30 simulations for each of the four quarters tested. It is interesting because it illustrates after how many iterations the size of the tree begins to converge on it's optimum (see the graph above). This could be helpful in designing future experiments, although the size of the tree is sensitive to certain operators like the growth mutation rate and the crossover probability.



The average heterogeneity of the decision trees

Heterogeneity is the quality of being different. In terms of Genetic Programming this measures the degree to which the tree is different from itself. It indicates how many unique indicators exist in the tree and how many indicators are repeated. A high value here indicates that there are many different indicators in the decision tree whereas a smaller value indicates that there are few indicators forming the decision tree. The graph below illustrates the relationship between the heterogeneity of the tree and the return on investment it produced. It is interesting because it indicates whether complex (highly heterogeneous) or simple (not heterogeneous) strategies perform better over time.

[1]

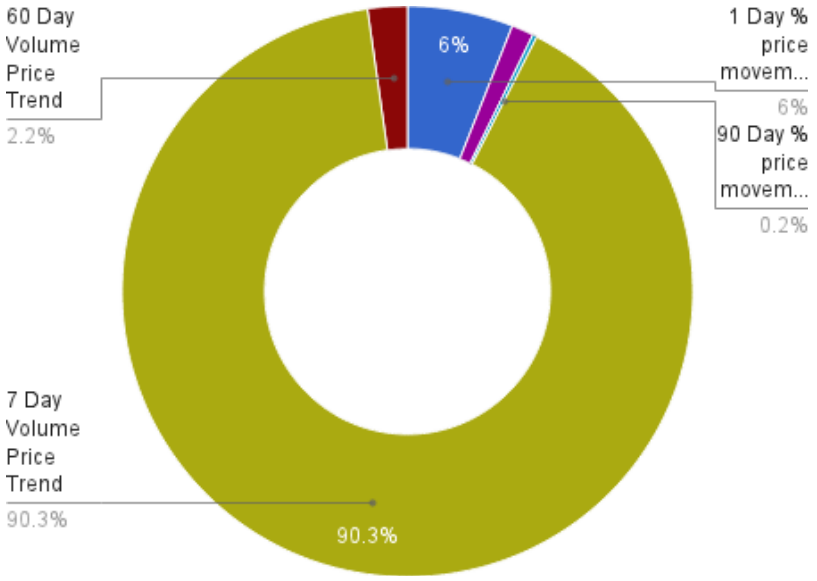


[1] Note that this graph was produced using 'buckets'. As in different buckets into which the heterogeneity could fall were created and the fitnesses of the decision trees in that bucket were averaged.

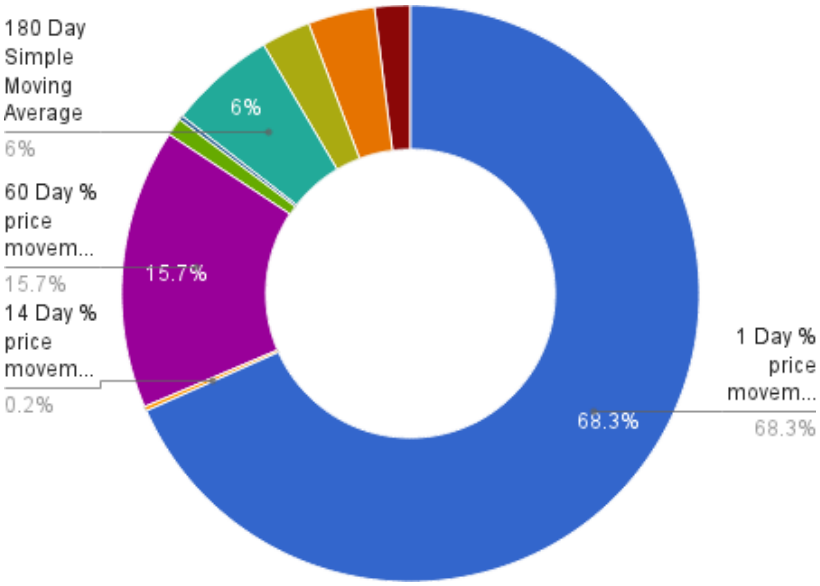
The indicators used most frequently in the decision trees

For each one of the quarters simulated, pie charts have been produced showing which indicators on average across all 30 simulations were most commonly found in the decision trees. This analysis is interesting because it could indicate which indicators are better at predicting stock returns over differing period of time.

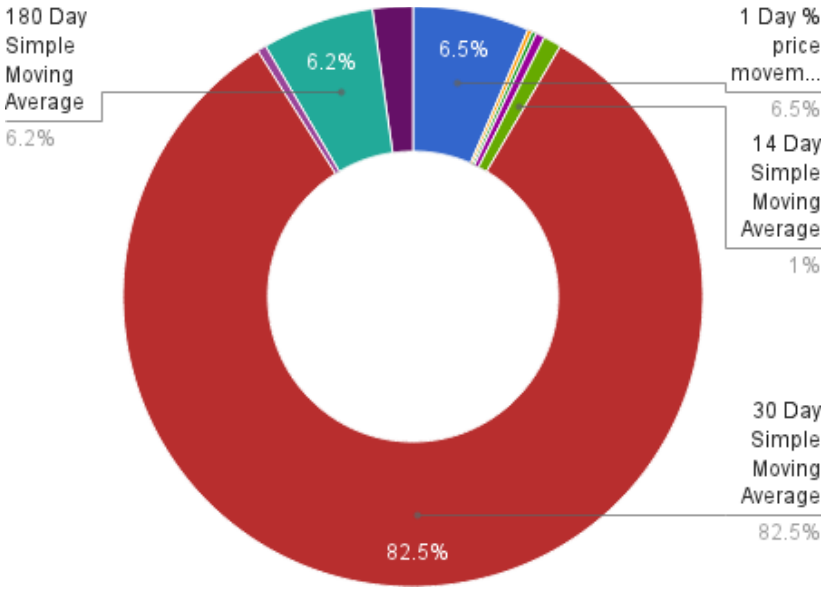
Quarter One most used technical indicators



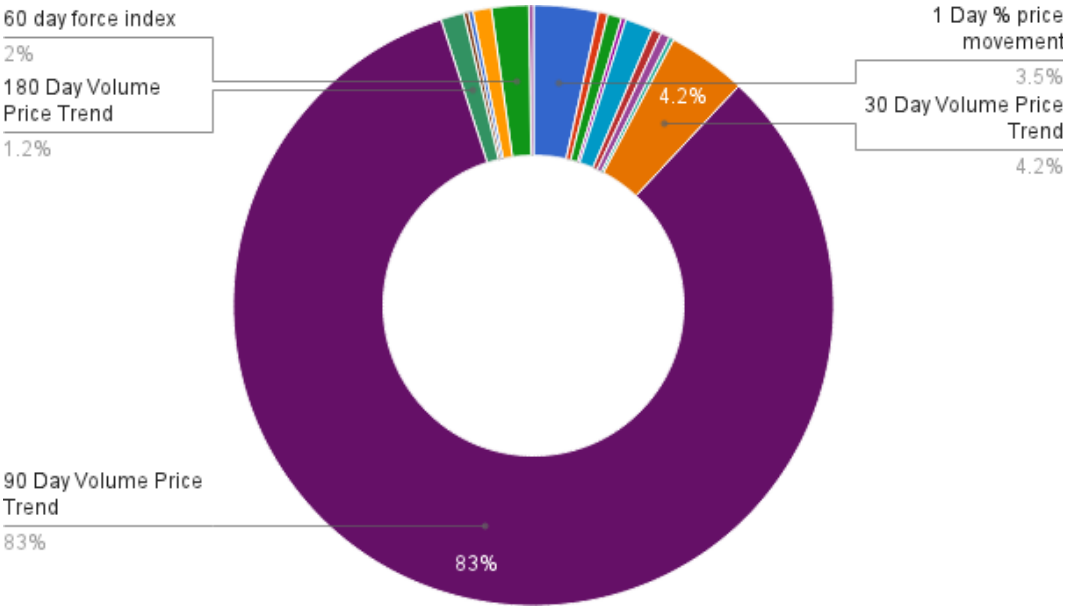
Quarter Two most used technical indicators



Quarter Three most used technical indicators



Quarter Four most used technical indicators

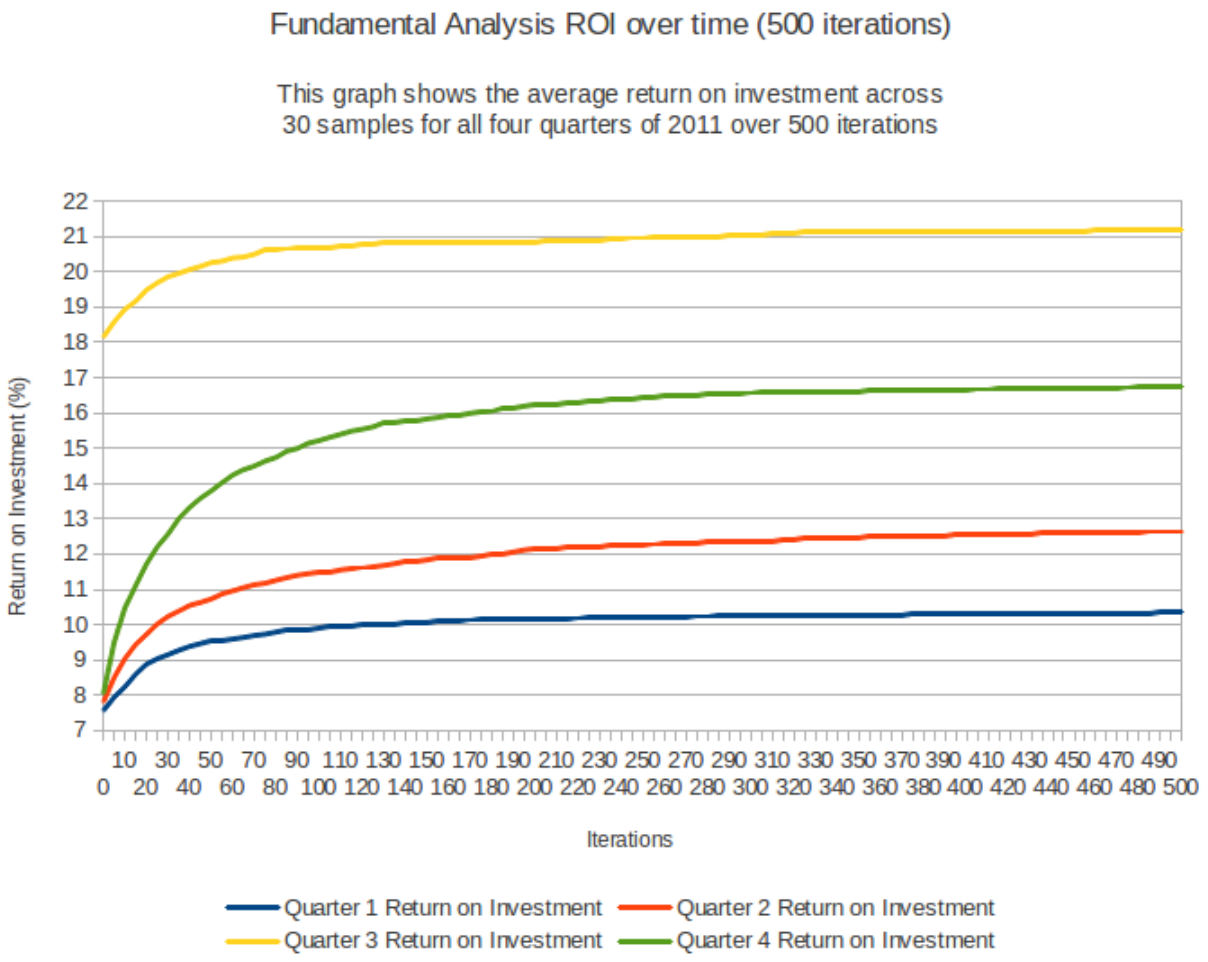


Results for Decision Trees using Fundamental Indicators

Please note that these results were obtained during the experiments done by my colleague, Simon Van Dyk. They were produced using the same algorithmic framework and control parameters and are thus comparable. The only difference is that his decision trees were evolved using fundamental analysis indicators[1].

Returns on Investment over time

The graph below shows the improvement to the return on investment over time (iterations) for the first 500 iterations only. As with the results seen in the Technical Analysis results, the majority of improvement occurs within the first 150 iterations however the population does not stagnate after that point as improvements are still being made albeit slowly. On average the returns produced by Fundamental Analysis appear to be higher than by Technical Analysis.



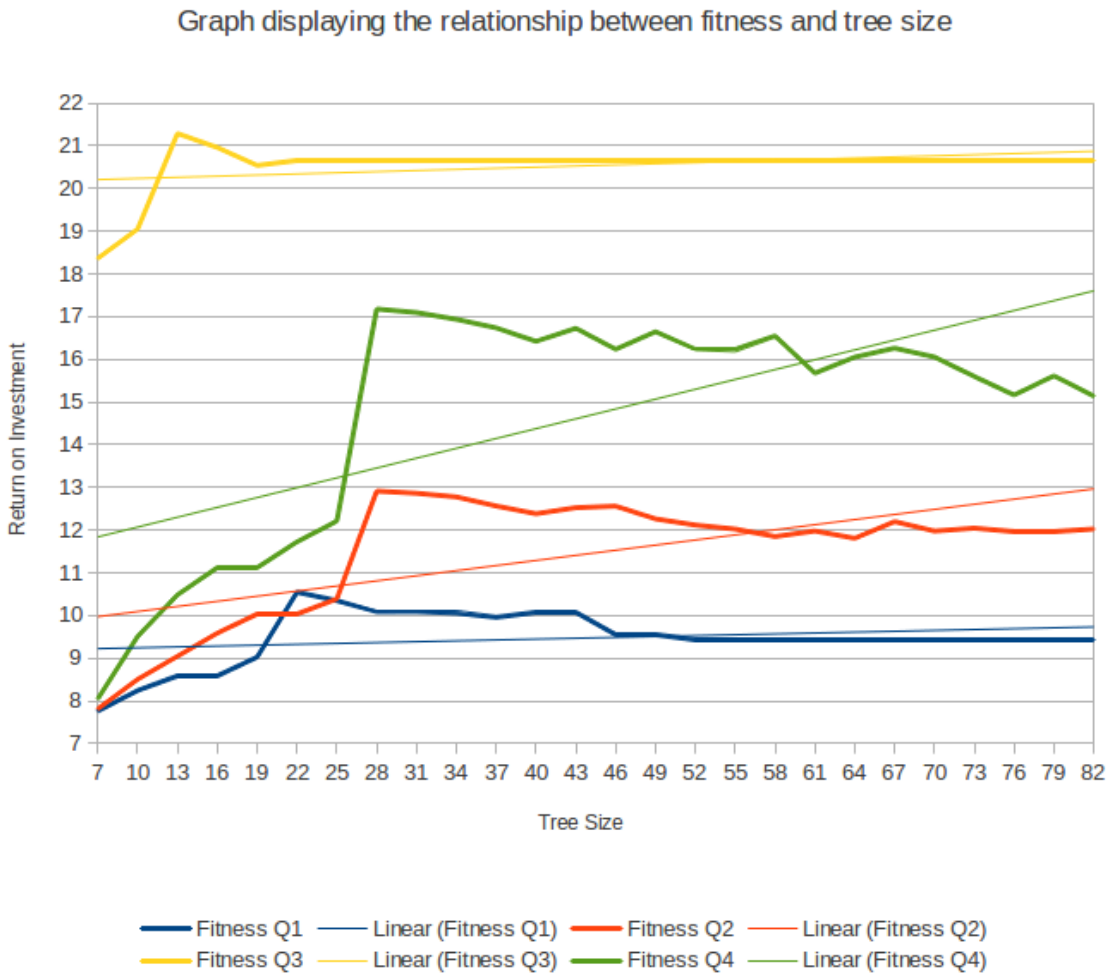
[1] For a list of the Fundamental Analysis Indicators that Simon made use of please refer to Appendix B

Comment on the Security Analysis problem

Provide a comment on the nature of Security Analysis and the effect that changes in the algorithms control parameters had on performance.

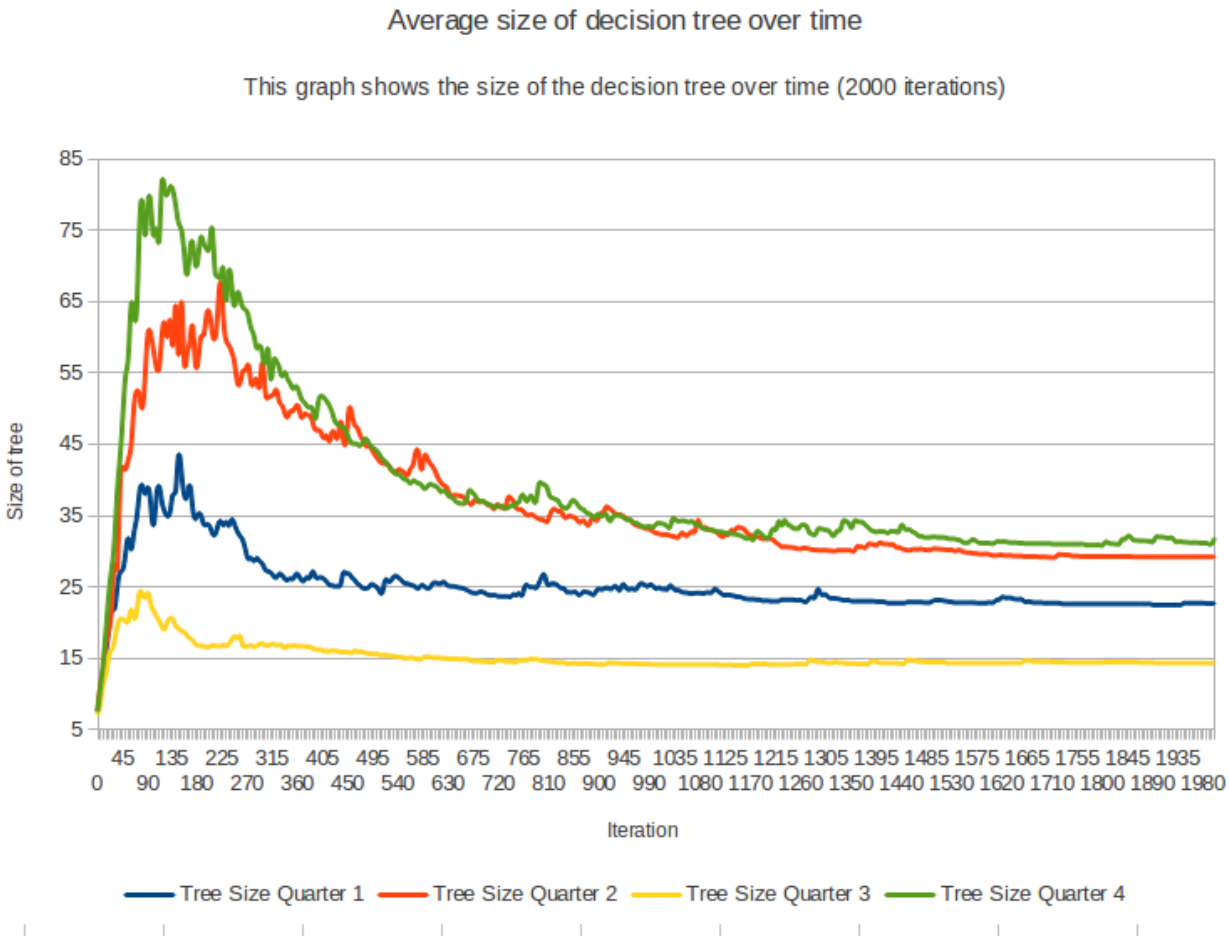
The average performance of the decision trees relative to size

The graph below illustrates what the average return on investment was across the 30 samples simulated for the different tree sizes for each one of the quarters. It is an interesting graph because it shows at which point an increase in the size of the tree stopped having a positive effect on the performance of the genetic programming algorithm. In other words it shows the optimum size of the decision tree for each quarter.



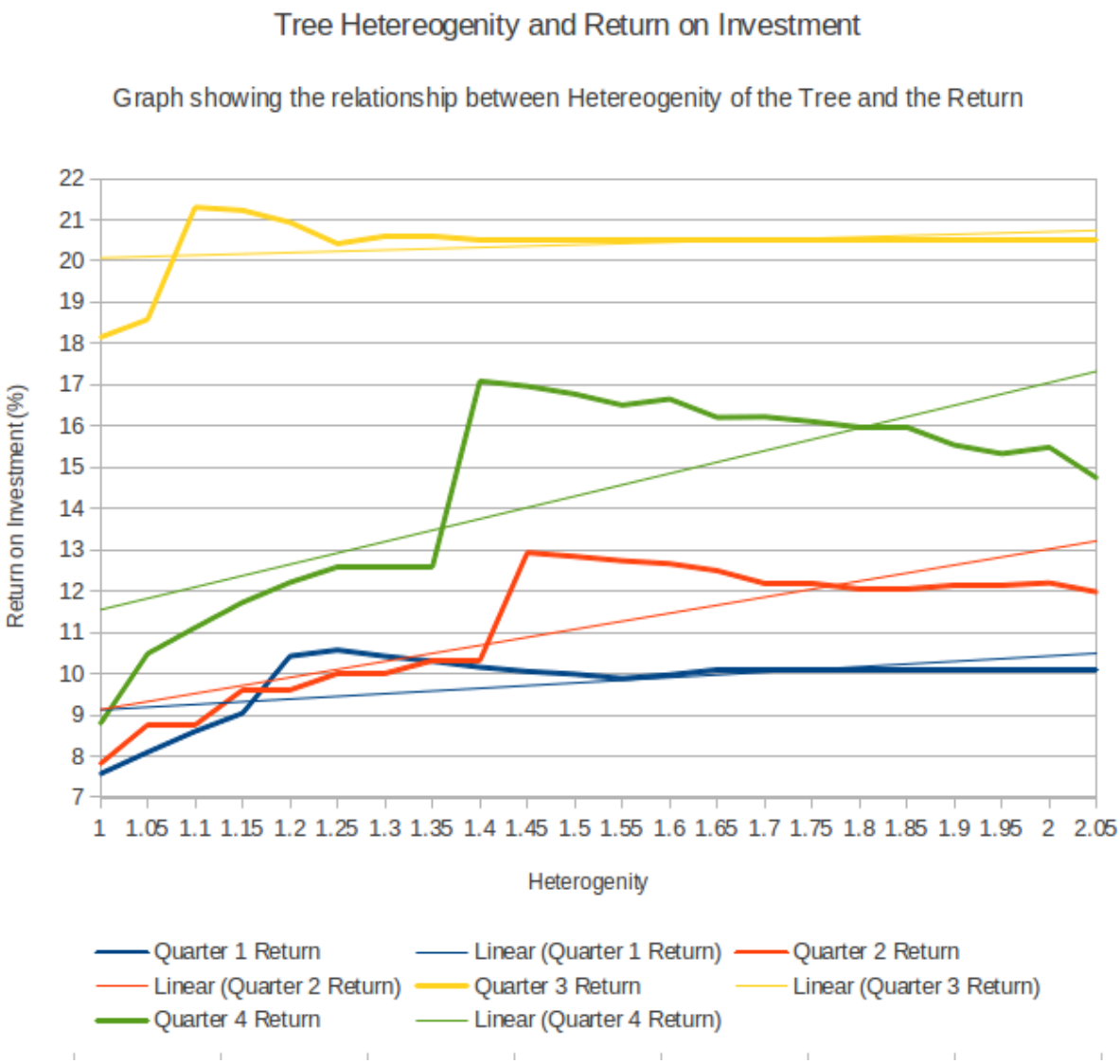
The average sizes of the decision trees

As compared with the Technical Analysis tree growth rates, the Fundamental analysis trees did grow very quickly as well however their growth was less aggressive and the trees took longer to converge onto their optimal sizes. The graph below shows the average tree size per iteration across all 30 simulations for each of the four quarters tested. It is interesting because it illustrates after how many iterations the size of the tree begins to converge on it's optimum (see the graph above). This could be helpful in designing future experiments, although the size of the tree is sensitive to certain operators like the growth mutation rate and the crossover probability.



The average heterogeneity of the decision trees

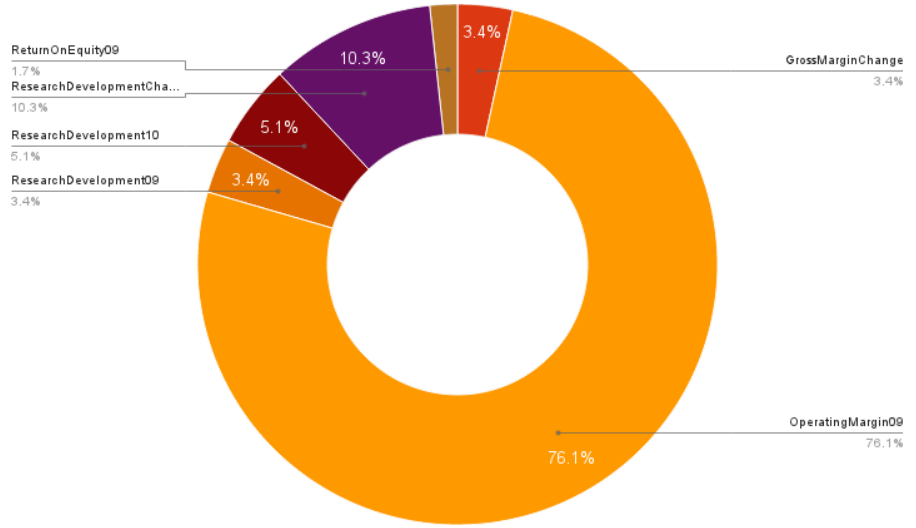
The graph below illustrates the relationship between the heterogeneity of the tree and the return on investment it produced. It is interesting because it indicates whether complex (highly heterogeneous) or simple (not heterogeneous) strategies perform better over time.



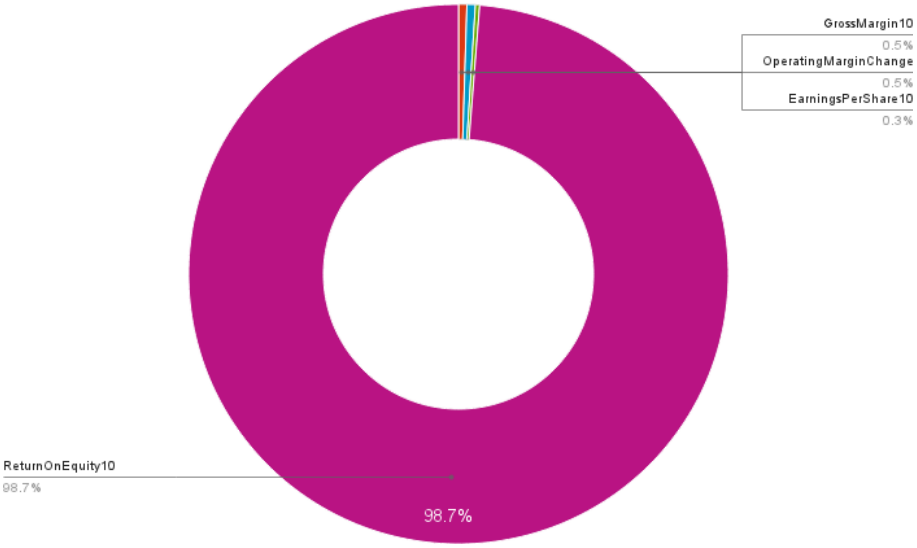
The indicators used most frequently in the decision trees

For each one of the quarters simulated, pie charts have been produced showing which indicators on average across all 30 simulations were most commonly found in the decision trees. This analysis is interesting because it could indicate which indicators are better at predicting stock returns over differing period of time.

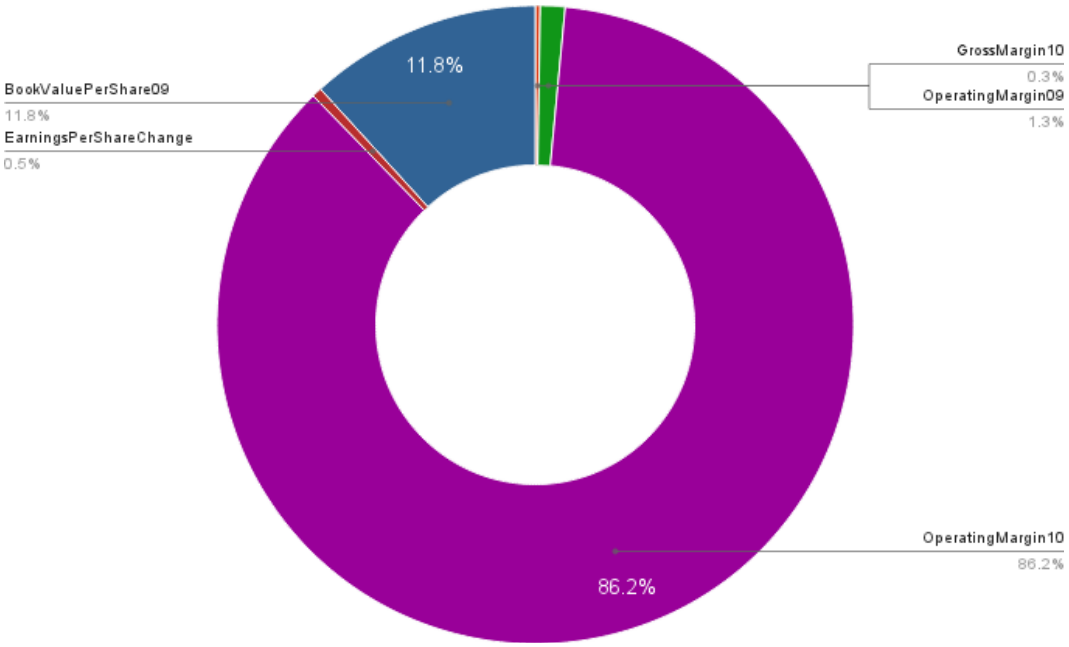
Quarter One most used fundamental indicators



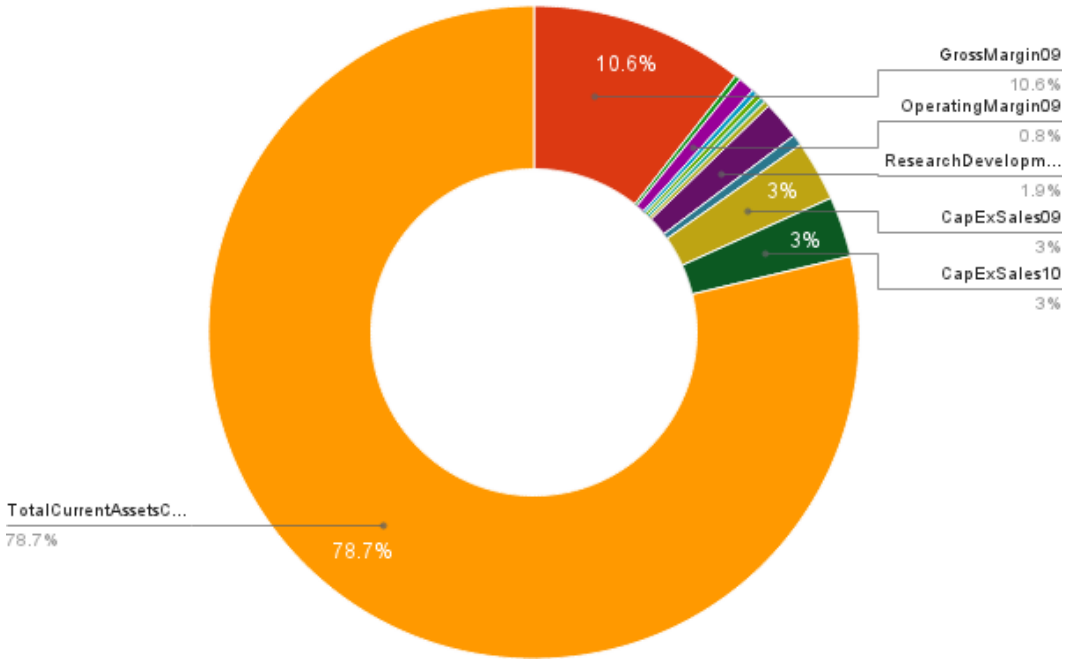
Quarter Two most used fundamental indicators



Quarter Three most used fundamental indicators



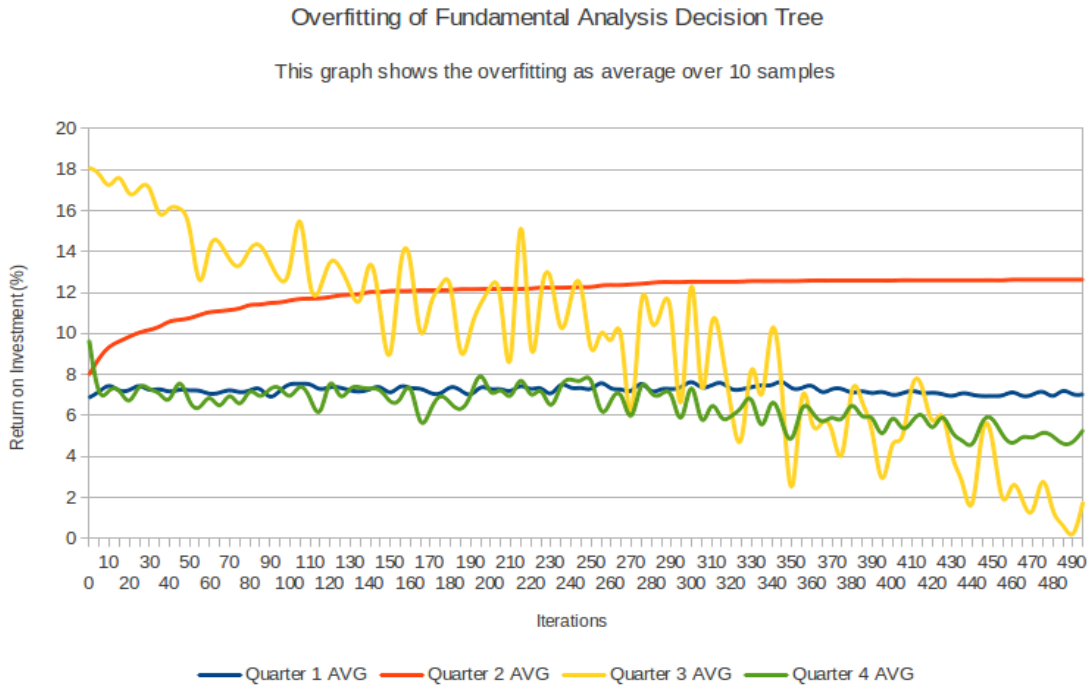
Quarter Four most used fundamental indicators



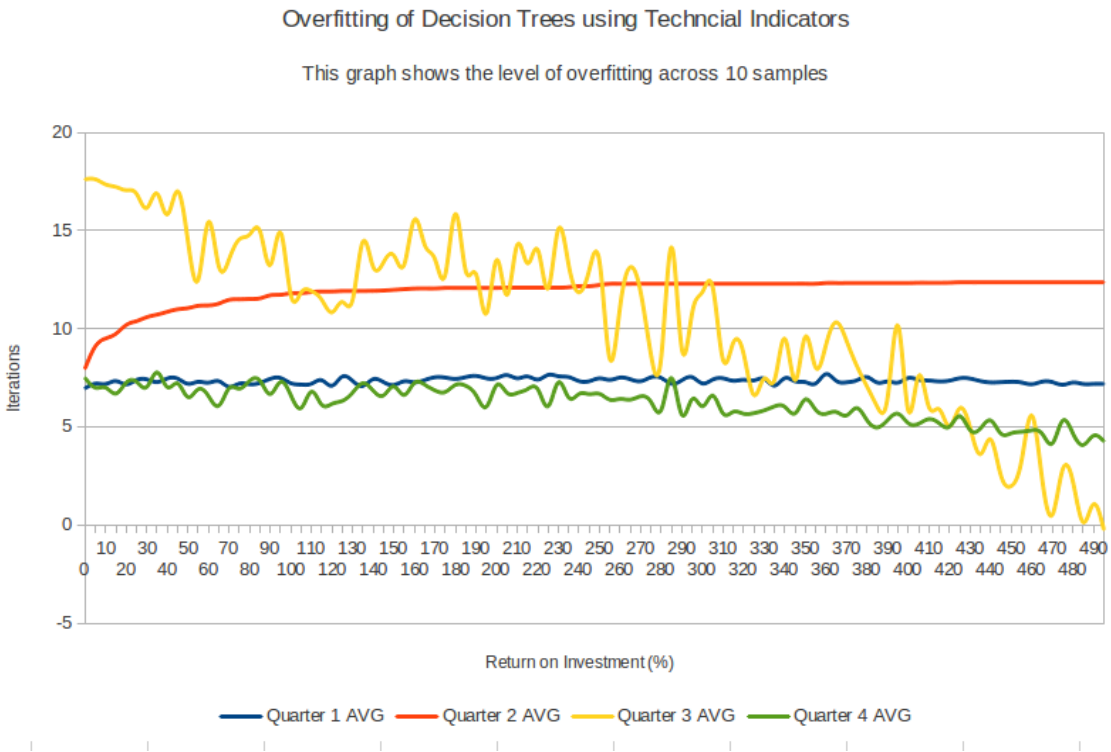
Additional Comment - The problem with Overfitting

Overfitting is one of the fundamental challenges faced by many Artificial Intelligence algorithms, especially those in machine learning. Genetic Programming is not excluded from that list and it the algorithm can overfit the trees it evolves to the problem being solved. This means that the trees cannot be generalized and used on unseen problems.

The severity of overfitting normally increases with the number of generations the algorithm is run for. In this research study overfitting was a concern and many of the decision trees evolved did overfit the periods for which they were evolved. However, to what extent this is a concern for fixed period investment strategies, is still unclear. Below are two graphs illustrating the average performance of decision trees evolved for the second quarter of 2011 in the third and fourth quarter. As you will see the overfitting worsens over time.



The above graph illustrates that for the first and fourth fiscal quarter, the effect of overfitting by decision trees using Technical Indicators resulted in near to average market returns (6 - 10%) for the Security Analysis Decision Tree. After approximately 270 iterations, this worsened to below market returns. In the case of the third quarter overfitting was not bad for the first 100 iterations as the decision tree was still able to produce returns above the market return (-18%), however after 100 iterations this performance deteriorates rapidly toward the market return.



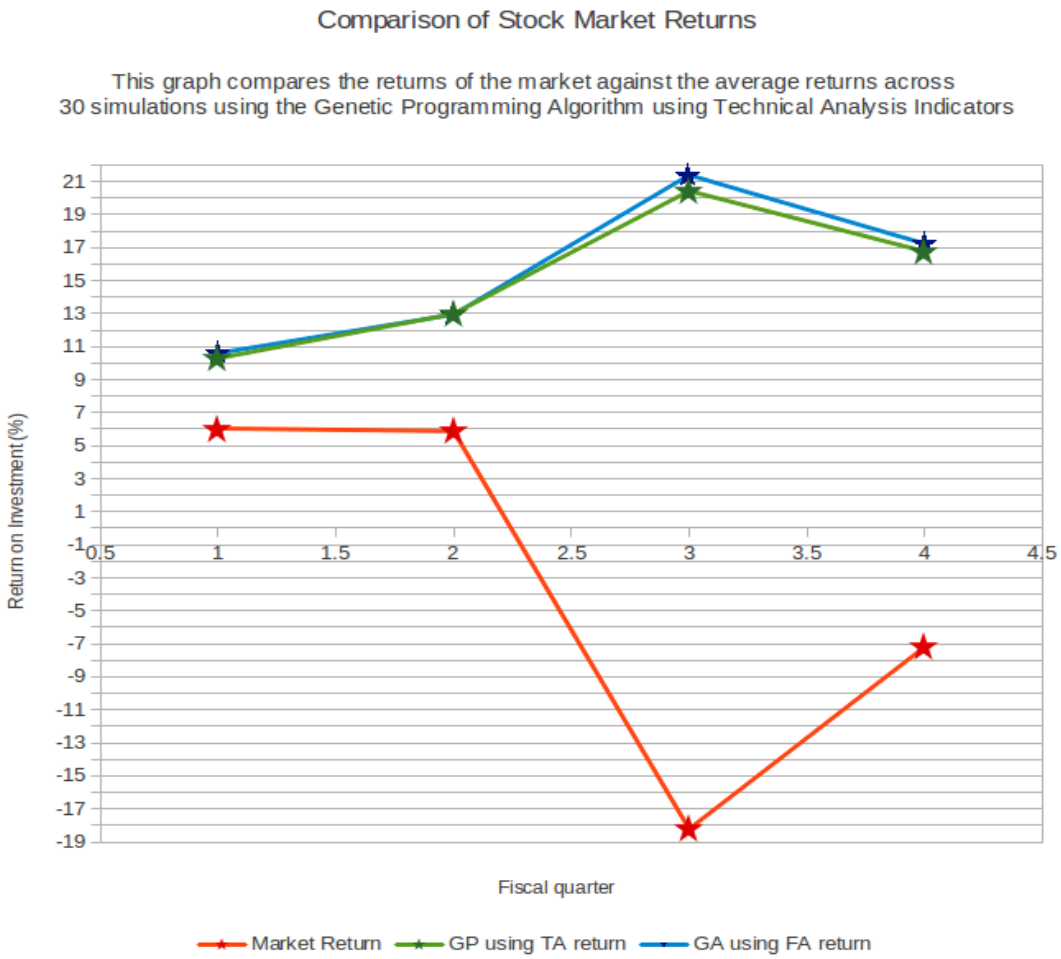
The above graph illustrates that for the first and fourth fiscal quarter, the effect of overfitting by decision trees using Technical Indicators resulted in near to average market returns (6 - 10%) for the Security Analysis Decision Tree. After approximately 200 iterations, this worsened to below market returns. In the case of the third quarter overfitting was not bad for the first 300 iterations as the decision tree was still able to produce returns above the market return (-18%), however after 300 iterations this performance deteriorates rapidly toward the market return.

Additional Comment - Technical Analysis? Or Fundamental Analysis?

In this comment I will be comparing the performance of genetic programming evolved security analysis decision trees that use technical indicators against genetic programming evolved security analysis decision trees that use fundamental indicators.

Performance against the market using Technical & Fundamental Indicators

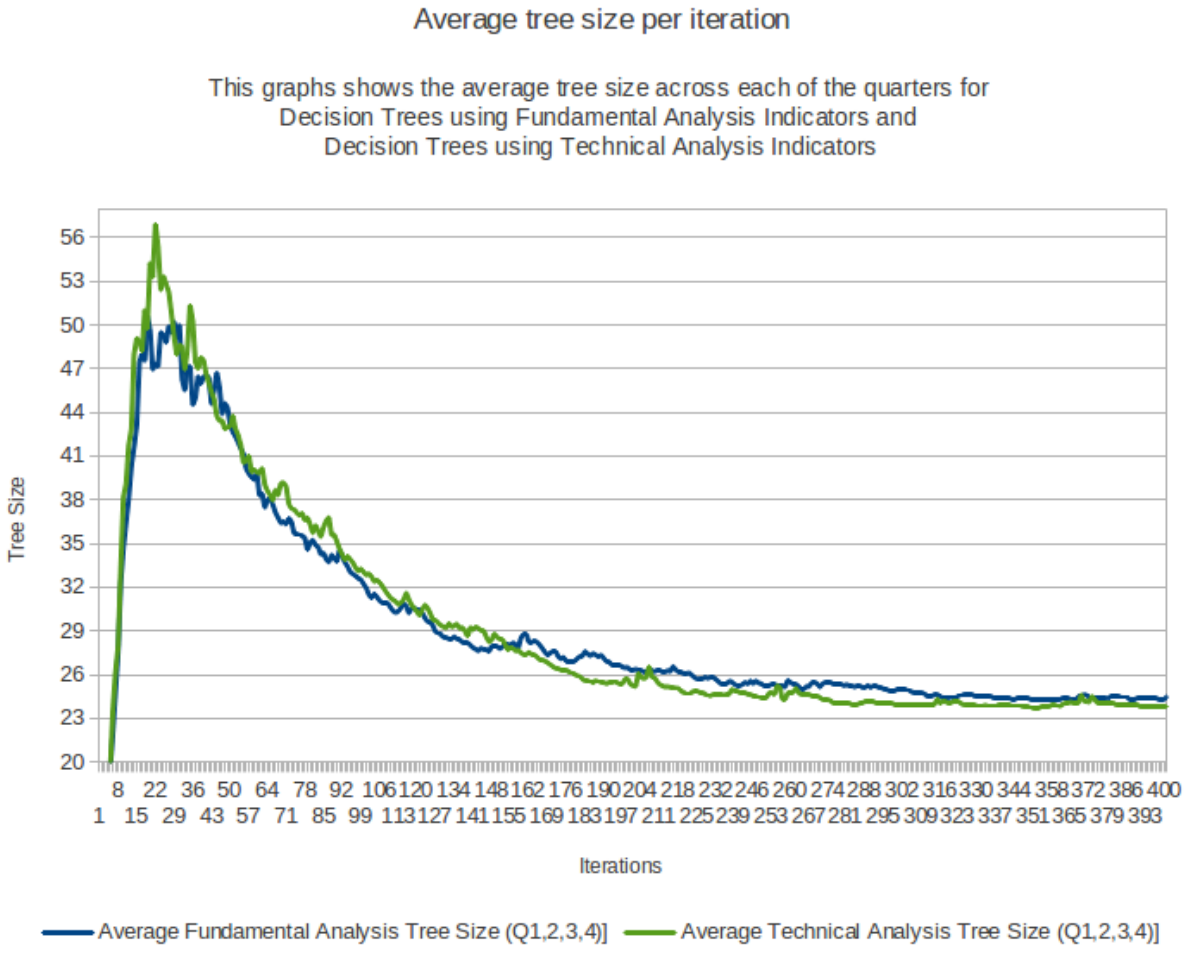
	Q 1	Q 2	Q 3	Q 4
Average Market Returns	6.04	5.88	-18.21	-7.19
GP using Technical Analysis Average ROI	10.29	12.98	20.42	16.76
GA using Fundamental Analysis Average ROI	10.60	12.94	21.39	17.23



The average tree sizes evolved by the Genetic Programming algorithm that used technical indicators and the one that used fundamental indicators were very similar.

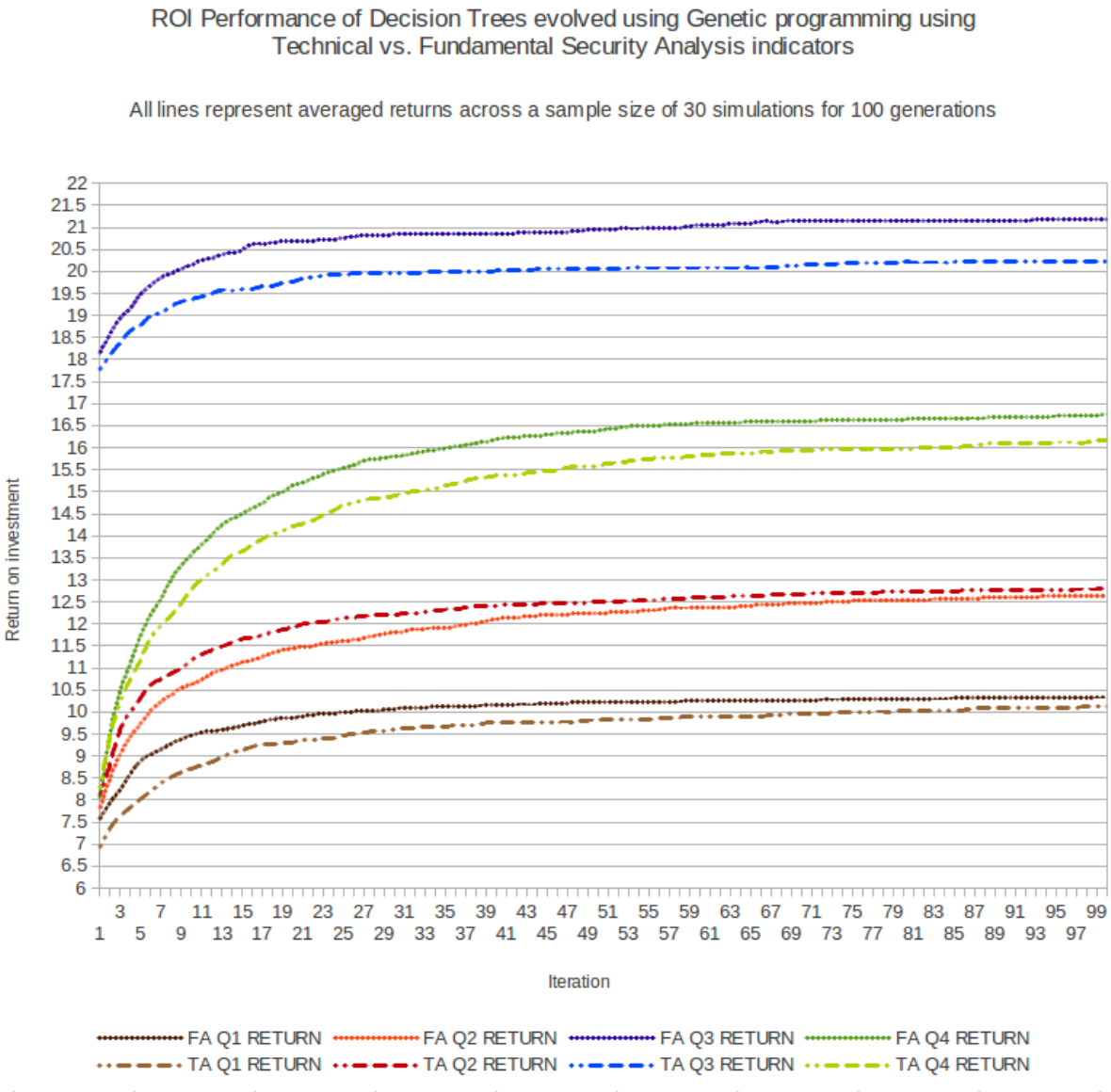
One observable difference was that the size to which decision trees using technical analysis indicators grew was larger than the size to which decision trees using fundamental indicators grew. On average after 20 to 30 iterations, trees using technical indicators contained 56 investment decision (leaf) nodes whereas trees using fundamental analysis contains 50 investment decision (leaf) nodes.

Another observable difference was that over time the Genetic Programming algorithm was able to find increasingly smaller trees that did not reduce the return on investment of the strategy regardless of whether technical or fundamental indicators were being used. This implies that a good strategy for security analysis does not necessarily need to be complex.



The graph below shows the average return on investment made by the security analysis decision trees across the thirty simulations for the 62 Information Technology stocks listed on the S&P500. This graphs tells a very interesting story.

Despite being prone to overfitting we can observe that in three out of the four time periods simulated, decision trees using fundamental indicators outperformed decision trees using technical indicators. In addition to this, the graph shows that as the investment duration increases, the relative performance of fundamental analysis against technical analysis increases. This implies that fundamental analysis is better for long term security analysis, which is consistent with the literature on the topic.



Conclusions

Performance against the market conclusions

The genetic programming algorithm has great potential to evolve new strategies for security analysis and investment management provided that better functions for calculating fitness can be derived. Throughout this research study we have seen that decision trees evolved using Genetic Programming are able to produce stock classifications that beat the average market return consistently. This is true for decision trees evolved to use technical indicators as well as decision trees evolved to use fundamental indicators.

Security analysis conclusions

Optimal sizes for decision trees

For decision trees evolved using technical indicators the optimal tree size lies between 25 and 31. That is, a decision tree optimized to produce high returns will have between 25 and 31 investment decisions (terminal nodes). Based on the properties of a binary tree therefore, we can estimate that an optimal decision tree using technical analysis will have between 49 and 60 investment rules (functional nodes).

For decision trees evolved using fundamental indicators the optimal tree size lies between 28 and 34. That is, a decision tree optimized to produce high returns will have between 28 and 34 investment decisions (terminal nodes). Based on the properties of a binary tree therefore, we can estimate that an optimal decision tree using fundamental analysis will have between 54 and 66 investment rules (functional nodes).

These two conclusions are interesting because they imply that optimal strategies for security analysis that make use of fundamental indicators will, on average, be slightly more complex than those that make use of technical indicators.

Optimal levels of heterogeneity for decision trees

The levels of heterogeneity found in the decision trees in this research study were correlated with the size of that tree. That means that as the trees get larger, they become more and more diverse. This is an intuitive conclusion and makes sense. This statement is supported by the fact that the graphs showing the effects of tree heterogeneity on return mirror the graphs showing the effects of tree size on return.

Conclusions pertaining to the indicators used

It is difficult to draw conclusions regarding which indicators were most indicative of future stock market returns since we do not know what proportion of the indicator selection was as a result of overfitting and what proportion was as a result of true genetic evolution. That having been said, we can conclude that following indicators could bear a strong relationship to future stock market returns because they were employed substantially by the Genetic Programming algorithm in the best security analysis decision trees:

Fundamental Indicators

1. Operating Margins - A ratio used to measure a company's pricing strategy and operating efficiency
2. Gross Margin - A company's total sales revenue minus its cost of goods sold, divided by the total sales revenue, expressed as a percentage.
3. Book Value Per Share - A measure used by owners of common shares in a firm to determine the level of safety associated with each individual share after all debts are paid accordingly.
4. Research and Development - Investigative activities that a business chooses to conduct with the intention of making a discovery that can either lead to the development of new products or procedures, or to improvement of existing products or procedures.
5. Return on Equity - The amount of net income returned as a percentage of shareholders equity. Return on equity measures a corporation's profitability by revealing how much profit a company generates with the money shareholders have invested.
6. Total Current Assets - A balance sheet account that represents the value of all assets that are reasonably expected to be converted into cash within one year in the normal course of business.

Technical Indicators

1. Simple Moving Averages - A simple, or arithmetic, moving average that is calculated by adding the closing price of the security for a number of time periods and then dividing this total by the number of time periods. Short-term averages respond quickly to changes in the price of the underlying, while long-term averages are slow to react.
2. Volume Price Trends - A technical indicator consisting of a cumulative volume line that adds or subtracts a multiple of the percentage change in share price trend and current volume, depending upon their upward or downward movements
3. Price Movements (%) - This indicator calculated the percentage price movement from the date that the securities were either bought or shorted (3 Jan 2011) until the current 2011 quarter being tested.

Conclusions regarding whether to use Technical or Fundamental Indicators

Whilst fundamental indicators did perform better on average than fundamental indicators in the decision trees evolved in this research study, both performed relatively well when compared with the average market returns. Therefore I am inclined to conclude that a strategy for security analysis should consider taking into account both technical analysis as well as fundamental analysis. That having been said, more research needs to be done on other types of indicators beyond just the basic fundamental and technical indicators used in this research study.

Recommended Further Research

The genetic programming algorithm has great potential in evolving new strategies for performing security analysis. This warrants further, more detailed research into the topic. What this research study has done successfully is break new ground and provide a proof of concept upon which future research and development should occur. Below is a list of additional research topics that could be researched to provide a more complete picture on how to use Genetic Programming in Finance.

1. Conduct an investigation into different types of fitness functions that would reduce the effects of overfitting. This investigation would ideally identify aggregate fitness functions that are measured at multiple point in time such as weekly or monthly. This investigation should also consider more complex fitness functions that incorporate risk models such as Jensen's Alpha or Value at Risk (VaR) as well as more complex fitness functions from the Computer Science discipline such as Canary fitness functions.
2. Extend the research study to cover different categories of financial indicators including Quantitative Indicators, Sentiment Indicators and Macroeconomic Indicators. Additional indicators in technical analysis and fundamental analysis should also be investigated including the newer technical indicators and qualitative fundamental indicators.
3. The scope of the research study could be extended in terms of sectors and economies. Similar research studies could be performed for sectors outside of the Information Technology stocks on the S&P500 or even outside of the US in Asian, African and European markets.
4. Additionally, the scope could be extended to look at producing security analysis decision trees to classify more than just equities (stocks). Research could be done to evolve decision trees for buying bonds, commodities and derivatives.

There are many other possibilities in the field of Computational Finance that are worth investigating. If you are interested in any of the aforementioned topics or would like to brainstorm more topics, my details are included on the front page of this report.

-- Thank You --

Appendices

Appendix A - Full list of companies

The list below shows the 62 of the 65 companies that were used in determining the fitness of each of the decision trees evolved in this assignment. Due to incomplete data or data inconsistencies three companies had to be removed from the sample including: ADI - Analog Devices, CSCO - Cisco Systems and VRSN - Verisign.

Ticker	**Company Name**
AAPL	Apple Inc.
ACN	Accenture plc
ADBE	Adobe Systems Inc
ADP	Automatic Data Processing
ADSK	Autodesk Inc
AKAM	Akamai Technologies Inc
ALTR	Altera Corp
AMAT	Applied Materials Inc
AMD	Advanced Micro Devices
BMC	BMC Software
BRCM	Broadcom Corporation
CA	CA, Inc.
CRM	Salesforce.com
CSC	Computer Sciences Corp.
CTSH	Cognizant Technology Solutions
CTXS	Citrix Systems
DELL	Dell Inc.
EA	Electronic Arts
EBAY	eBay Inc.
EMC	EMC Corp.
FFIV	F5 Networks

FIS	Fidelity National Information Services
FISV	Fiserv Inc
GOOG	Google Inc.
HPQ	Hewlett-Packard
HRS	Harris Corporation
IBM	International Bus. Machines
INTC	Intel Corp.
INTU	Intuit Inc.
JBL	Jabil Circuit
JDSU	JDS Uniphase Corp.
JNPR	Juniper Networks
KLAC	KLA-Tencor Corp.
LLTC	Linear Technology Corp.
LRCX	Lam Research
LSI	LSI Corporation
MA	Mastercard Inc.
MCHP	Microchip Technology
MOLX	Molex Inc.
MSFT	Microsoft Corp.
MU	Micron Technology
NFLX	NetFlix Inc.
NTAP	NetApp
NVDA	Nvidia Corporation
ORCL	Oracle Corp.
PAYX	Paychex Inc.
QCOM	QUALCOMM Inc.
RHT	Red Hat Inc.

SNDK	SanDisk Corporation
STX	Seagate Technology
SYMC	Symantec Corp.
TDC	Teradata Corp.
TEL	TE Connectivity Ltd.
TER	Teradyne Inc.
TSS	Total System Services
TXN	Texas Instruments
V	Visa Inc.
WDC	Western Digital
WU	Western Union Co
XLNX	Xilinx Inc
XRX	Xerox Corp.
YHOO	Yahoo Inc.

Appendix B - Full List of Fundamental Indicators used

Indicator	Meaning of the Indicator	Time Periods
Gross Margin %	A company's total sales revenue minus its cost of goods sold, divided by the total sales revenue, expressed as a percentage.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Operating Margin %	A ratio used to measure a company's pricing strategy and operating efficiency.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Earnings Per Share USD	The portion of a company's profit allocated to each outstanding share of common stock. Earnings per share serves as an indicator of a company's profitability.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Book Value Per Share USD	A measure used by owners of common shares in a firm to determine the level of safety associated with each individual share after all debts are paid accordingly.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
SG&A	'Selling, General & Administrative Expense - SG&A' is reported on the income statement, it is the sum of all direct and indirect selling expenses and all general and administrative expenses of a company.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
R&D	Investigative activities that a business chooses to conduct with the intention of making a discovery that can either lead to the development of new products or procedures, or to improvement of existing products or procedures. Research and development is one of the means by which business can experience future growth by developing new products or processes to improve and expand their operations.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Return on Assets %	An indicator of how profitable a company is relative to its total assets. ROA gives an idea as to how efficient management is at using its assets to generate earnings. Calculated by dividing a company's annual earnings by its total assets, ROA is displayed as a percentage.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Return on Equity %	The amount of net income returned as a percentage of shareholders equity. Return on equity measures a corporation's profitability	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two

	by revealing how much profit a company generates with the money shareholders have invested.	
Return on Invested Capital %	A calculation used to assess a company's efficiency at allocating the capital under its control to profitable investments. The return on invested capital measure gives a sense of how well a company is using its money to generate returns.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Cap Ex as a % of Sales	Funds used by a company to acquire or upgrade physical assets such as property, industrial buildings or equipment. This type of outlay is made by companies to maintain or increase the scope of their operations. These expenditures can include everything from repairing a roof to building a brand new factory.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Total Current Assets	A balance sheet account that represents the value of all assets that are reasonably expected to be converted into cash within one year in the normal course of business. Current assets include cash, accounts receivable, inventory, marketable securities, prepaid expenses and other liquid assets that can be readily converted to cash.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Total Current Liabilities	A company's debts or obligations that are due within one year. Current liabilities appear on the company's balance sheet and include short term debt, accounts payable, accrued liabilities and other debts.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Current Ratio	A liquidity ratio that measures a company's ability to pay short-term obligations.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two
Quick Ratio	An indicator of a company's short-term liquidity. The quick ratio measures a company's ability to meet its short-term obligations with its most liquid assets. The higher the quick ratio, the better the position of the company.	This indicator was extracted for 2009 and 2010 and the YoY% change was calculated between the two