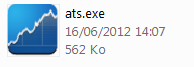
# Automated Trading System

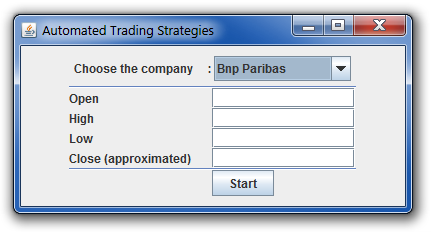
*Philippe REMY*

## Version 1.0

Double click on **ats.exe** to launch the application.



The main frame, similar to the screen shot below pops up.



At first you need to choose the underlying equity.

Open, High, Low, Close checkboxes must be filled before starting the process.

Before you click on start, you have to know that yahoo market data are retrieved from a past date to yesterday.

So the system cannot guess today prices.

At 17:30:00 (5min before the closure of the Paris SE).

|  |  |
| --- | --- |
| Type of price | Meaning |
| Open | Known |
| High | Mostly known (can be approximate) |
| Low | Mostly known (can be approximate) |
| Close | Unknown (need of approximation) |

Output is :

|  |  |
| --- | --- |
| Type of output | Value |
| True prediction count | Integer |
| False prediction count | Integer |
| Probability (True) | True / (False+True) |
| Expected value | Boolean |
| Expected price variation (min bound) | Float |

Example is :

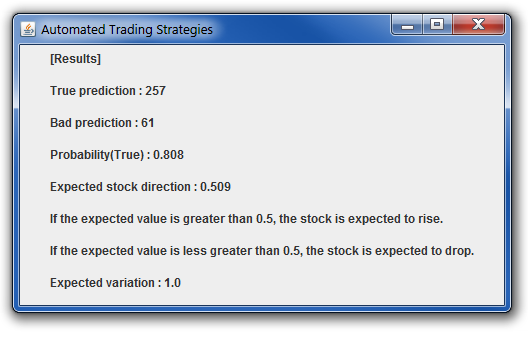
|  |  |
| --- | --- |
| Type of output | Value |
| True prediction count | 224 |
| False prediction count | 50 |
| Probability (True) | 0.817 |
| Expected value | 0 |
| Expected price variation (min bound) | 0.40 |

It means that the underlying mathematical model has a probability of 0.817 to emphasize true values among a given unknown set.

The expected value means that the system expects the stock to fall for tomorrow.

(0 for drop, 1 for rise)

Expected price variation is a a minima bound quantification of the equity price variation.



## Workflow

### Retrieval of market datas

[…]

“open”, “high”, “low”, “close”

574.46;576.62;566.7;576.16

574.52;578.48;570.38;572.16

571.24;573.5;567.26;571.53

571.0;574.62;569.55;574.13

[…]

### Addition of today prices (approximate prices)

[…]

“open”, “high”, “low”, “close”

574.46;576.62;566.7;576.16

574.52;578.48;570.38;572.16

571.24;573.5;567.26;571.53

571.0;574.62;569.55;574.13

[…]

**572.0;580.0;560.0;579.0**

### Binary transformation filter

Binary transformation is used to map the input set into a normalized and non-scattered space. How could the neural network know that going from 100 to 101 is the same as going from 1 to 1.01 ?

It is done as the following :

**For** each column

**If**(value[i+1] > value[i]) value[i] = 1

**Else** value[i] = 0

It is also known as the sign(derivative(x)) for each column.

[…]

0.0;0.0;1.0;0.0

1.0;1.0;1.0;1.0

1.0;1.0;1.0;1.0

0.0;1.0;0.0;1.0

1.0;1.0;1.0;0.0

0.0;0.0;0.0;1.0

[…]

### Splitting into two sets : training and testing

Default value is 70% for training set and 30% for testing set.

### Learning and minimizing the error on the testing set

Choosing the best neural network which has the minimal error rate for the sets.

### Retrieval of results and probabilities

Results on the error set may be alike :

|  |  |
| --- | --- |
| Computed Value | Expected Value |
| 1 | 1 |
| 0 | 1 |
| 1 | 1 |
| 0 | 0 |
| 0 | 0 |

True prediction : 4

Bad prediction : 1

Probability (True) : 0.80

### Removal of not optimized items

The pricing method for forecasting stock price is the first column known as the open prices colum.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Time | Open | High | Low | Close |
| i-3 | 1 | 0 | 1 | 0 |
| i-2 | 0 | 0 | 0 | 0 |
| i-1 | 1 | 1 | 1 | 0 |
| i | 1 | 1 | 1 | 1 |
| i+1 | **X (predicted value)** |  |  |  |

If(X = 1)

Open[i+1] > Open[i]

Else

Open[i+1] < Open[i]

Problem :

For computing Open[i+1], Open[i], High[i], Low[i], Close[i] (prices[i]) must be known (as well as lagged prices : prices[i-1], … prices[i-k]). The only moment when all the four variables are known is at the closure of Paris SE (17:35 CEST).

Second problem :

How worth overseeing X when you cannot buy at Open[i] to sell at Open[i+1] because you must wait for close[i] ?

Solution :

The space must be resized into a smaller space where you can assume an inequality between Open[i], Open[i+1] and Close[i].

For them model to be correct, these two following conditions have to be true :

* If the model finds X to be equal to 1 then the expectation is Open[i] < Open[i+1]. Close[i] must be lower than Open[i].

This way Close[i] < Open[i] < Open[i+1] and you known the quantity Open[i] – Close[i] which is the minima bound price variation.

* If X = 0, Close[i] must be greater than Open[i] so that Open[i+1] < Open[i] < Close[i].

You short at Close[i] and you buy at Open[i+1] which is lower than Open[i].

### Building new results and probabilities

Having considered these new conditions, the new results space [known as the optimized set] is recalculated.

### Running process with today prices and check if trade is optimized

Calculate the expected Open[i+1] for the last today prices. If it satisfies the above conditions, trade is allowed, else forbidden.

## Identified risks

The biggest risk is to estimate the right close price 1-5 min before.

If the equity closes at its highest or at its lowest, it may bring extra approximation for estimating (high,close) or (low,close) values.