MTH 9899 Machine Learing Final Project, May 2015

Technical Trading Strategies Using Linear Regression, Random Forest and Support Vector Machine

Master of Financial Engineering Program, Baruch College

Group Member

- Weiyi Chen, weiyi.chen@baruchmail.cuny.edu
- Haotian Wu, haotian.wu@baruchmail.cuny.edu
- Yifan Cui, yifan.cui@baruchmail.cuny.edu
- Chengbin Wang, chengbin.wang@baruchmail.cuny.edu

1. Introduction

1.1 Abstract

For this project, we intend to use different machine learning algorithms to modify traditional technical strategies, and backtest with research in cycle to find a good algorithm.

Detailed code and installation guide can be found in the following link in github

https://github.com/MTH-9899-Machine-Learning/ML-algotrade (https://github.com/MTH-9899-Machine-Learning/ML-algotrade)

Machine Learning algorithms:

- Linear Regression, http://scikit-learn.org/dev/modules/linear_model.html (http://scikit-learn.org/dev/modules/linear_model.html)
- Random Forests, http://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees (http://scikit-learn.org/stable/modules/ensemble.html#forests-of-randomized-trees)
- Support Vector Machine, http://scikit-learn.org/stable/modules/svm.html (http://scikit-learn.org/stable/modules/svm.html)

Technical Strategies:

- Momentum, http://stockcharts.com/school/doku.php?id=chart_school:trading_strategies:moving_momentum (http://stockcharts.com/school/doku.php?id=chart_school:trading_strategies:moving_momentum)
- Relative Strength Index (RSI), http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:relative_strength_index_rsi (http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:relative_strength_index_rsi)

1.2 Data Set and Software: Quantopian

Quantopian is our main research and backtesting platform, we research algorithmic trading ideas and explore curated financial data with the assistance of machine learning algorithm in this hosted research environment. Quantopian supports flexible data access, custom plotting, and post-hoc analysis on backtest.

For convenience to verify our machine learning algorithms by cheating ('cheat' is like using today's data to forecast today), we also support local access to data from Yahoo, with event study and backtesting tools from Python module QuantSoftware Toolkit. In concludes,

Data input

- Quantopian, or Yahoo
- Equity Universe: adjusted close of S&P500 index equity component
- Frequency: daily

Software

- Quantopian, a platform to build, test, and execute trading algorithms.
- QuantSoftware Toolkit, a modified Python-based open source software framework designed to support portfolio construction and management

1.3 Programming Language

Python. The requirement of this course suggests Matlab, but we request for using Python because our preferred platform Quantopian is using Python and the Python module QSTK we will be using is also in Python.

The main module we will be using is scipy-learn (http://scikit-learn.org/stable/supervised_learning.html#supervised-learning (http://scikit-learn.org/stable/supervised_learning.html#supervised-learning)), which contains implemented Machine Learning algorithms.

1.4 Books and Papers reference

- Alpaydin, Ethem. Introduction to machine learning. MIT press, 2014.
- Aronson, David. Evidence-based technical analysis: applying the scientific method and statistical inference to trading signals. Vol. 274. John Wiley & Sons, 2011.

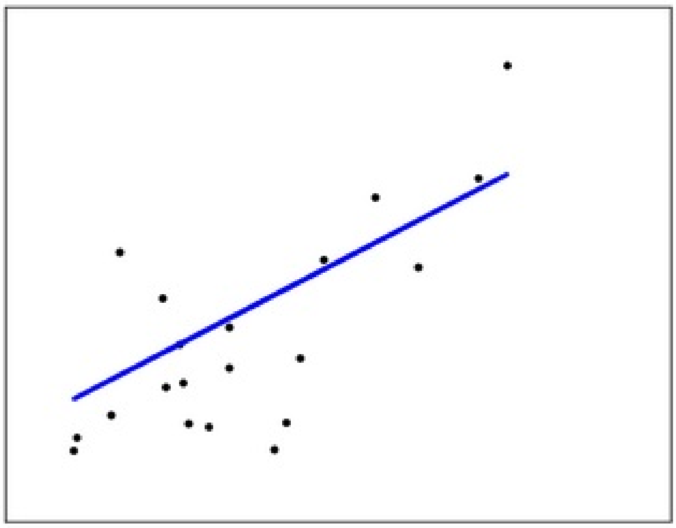
2. Background Knowledge

2.1 Machine Learning algorithms

2.1.1 Linear Regression (OLS)

LinearRegression fits a linear model with coefficients w = (w_1, ..., w_p) to minimize the residual sum of squares between the observed responses in the dataset, and the responses predicted by the linear approximation.

$$\min_{w} {\|Xw - y\|_2}^2$$

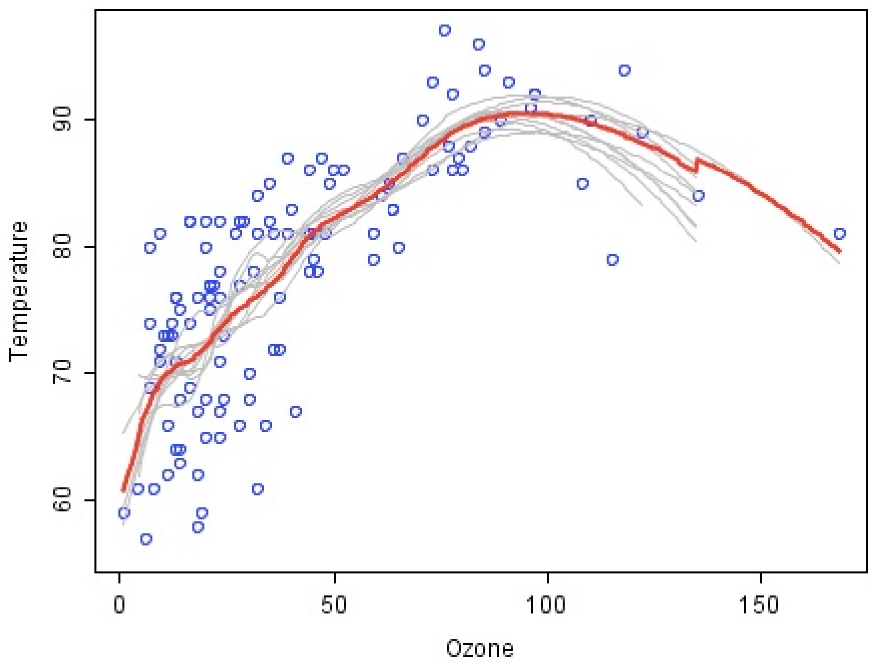


In [ ]:

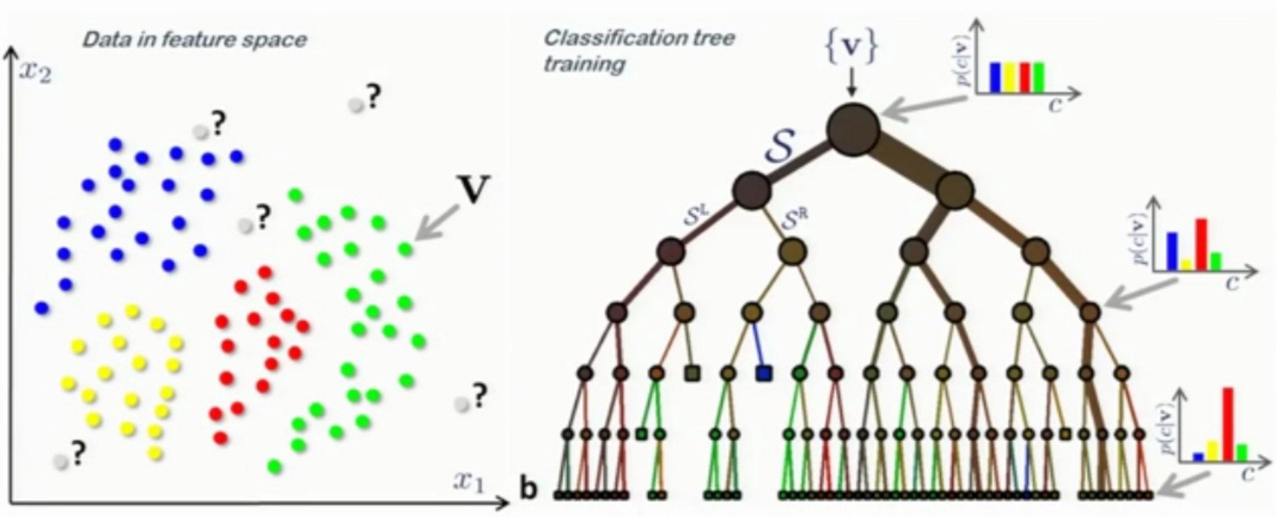
```
>>> from sklearn import linear_model
>>> clf = linear_model.LinearRegression()
>>> clf.fit ([[0, 0], [1, 1], [2, 2]], [0, 1, 2])
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
>>> clf.coef_
array([ 0.5,  0.5])
```

2.1.2 Random Forests

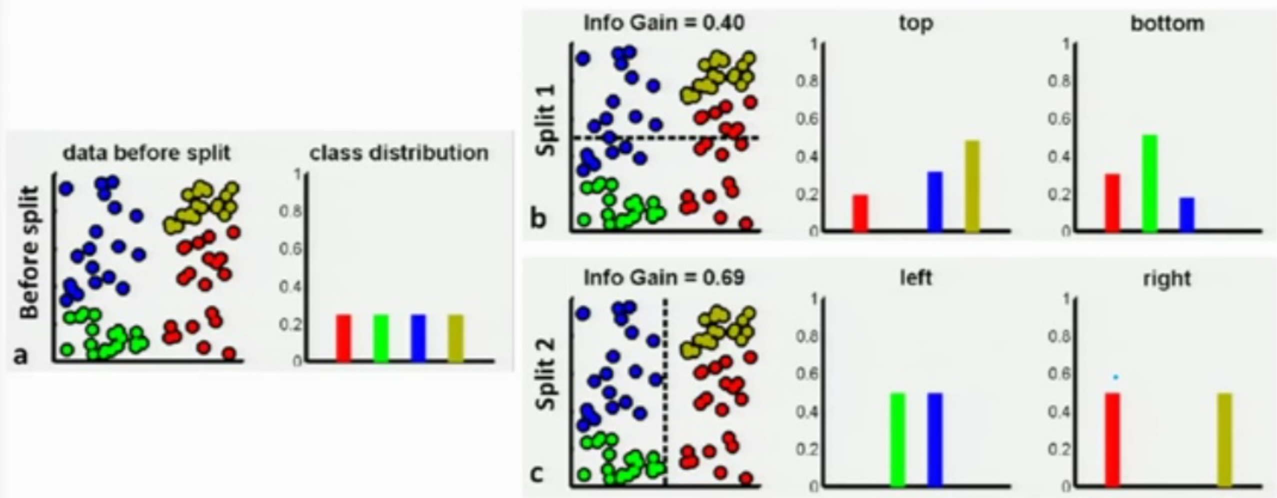
ensemble learning method (Unsupervised learning) the driving principle is to build several estimators independently and then to average their predictions. On average, the combined estimator is usually better than any of the single base estimator because its variance is reduced.



Temperature
Ozone
60
70
80
90
0
50
100
150



Data in feature space
Classification tree training
x_2
x_1
V
{v}
S
S^L
S^R
$p(c|\mathbf{v})$
c
b



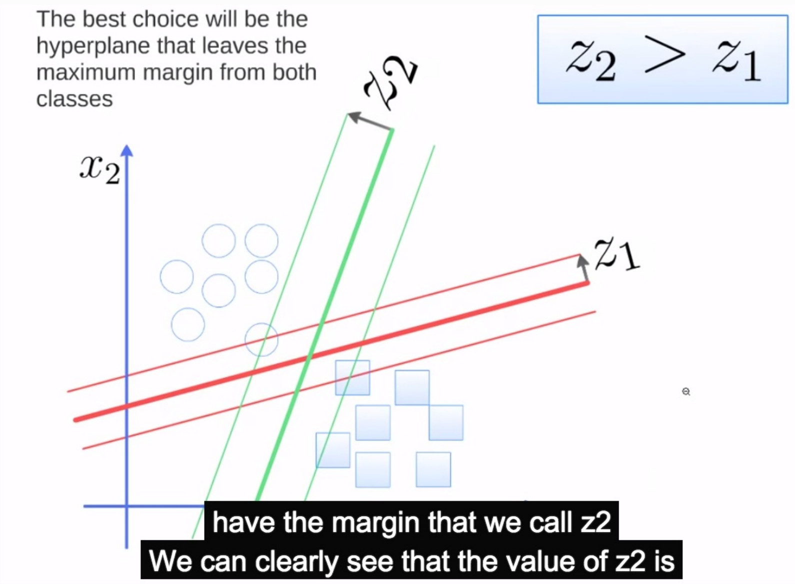
data before split
class distribution
Before split
a
Info Gain = 0.40
top
bottom
Split 1
b
Info Gain = 0.69
left
right
Split 2
c

In [ ]:

```
>>> from sklearn.ensemble import RandomForestClassifier
>>> X = [[0, 0], [1, 1]]
>>> Y = [0, 1]
>>> clf = RandomForestClassifier(n_estimators=10)
>>> clf = clf.fit(X, Y)
```

2.1.3 Support Vector Machine

Supervised methods



Minimizing $\vec{\omega}$ is a nonlinear optimization task, solved by the Karush-Kuhn-Tucker (KKT) conditions, using Langrange multipliers λ_i

$$\vec{\omega} = \sum_{i=0}^{N} \lambda_i y_i \vec{x}_i$$

Goal:

minimize $||w||^2$ and have low capacity

In [ ]:

```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.0, degree=3,
gamma=0.0, kernel='rbf', max_iter=-1, probability=False, random_state=None,
shrinking=True, tol=0.001, verbose=False)
```

2.2 Technical Strategies

2.2.1 Momentum

Strategy

Buy Signal:

1. Moving averages show a bullish trading bias with 20-day SMA trading above the 150-day SMA.
2. Stochastic Oscillator moves below 20 to signal a pullback.
3. MACD-Histogram moves into positive territory to signal an upturn after the pullback.

Sell Signal:

1. Moving averages show a bearish trading bias with the 20-day SMA trading below the 150-day SMA.
2. Stochastic Oscillator moves above 80 to signal a bounce.
3. MACD-Histogram moves into negative territory to signal a downturn after the bounce.

(a) Simple Moving Average (SMA)

Moving averages are trend-following indicators that lag price. This means the actual trend changes before the moving averages generate a signal. Many traders are turned off by this lag, but this does not make them totally ineffective. Moving averages smooth prices and provide chartists with a cleaner price plot, which makes it easier to identify the general trend.

This strategy employs two moving averages to define the trading bias. The bias is bullish when the shorter-moving average moves above the longer moving average. The bias is bearish when the shorter-moving average moves below the longer moving average. While chartists can use any combination of moving averages, this article uses the 20-day SMA and the 150-day SMA. The example below shows Baxter International (BAX) moving from a bullish trading bias to a bearish trading bias as the 20-day SMA moved below the 150-day SMA in mid August.

(b) Stochastic Oscillator

The second part of this trading strategy uses the Stochastic Oscillator to identify correction. As a bound oscillator that fluctuations between 0 and 100, the Stochastic Oscillator is ideal for spotting short-term pullbacks or bounces. A move below 20 signals a pullback in prices, while a move above 80 signals a bounce in prices.

$$
\begin{aligned}
\%K &= 100[(C - L14)/(H14 - L14)] \\
C &= \textit{the most recent closing price} \\
L14 &= \textit{the low of the } 14 \textit{ previous trading sessions} \\
H14 &= \textit{the highest price traded during the same } 14 \textit{ day period.} \\
\%D &= 3 - \textit{period moving average of } \%K
\end{aligned}
$$

The theory behind this indicator is that in an upward-trending market, prices tend to close near their high, and during a downward-trending market, prices tend to close near their low. Transaction signals occur when the %K crosses through a three-period moving average called the "%D".

(c) MACD-Histogram

The third part of this trading strategy uses the MACD-Histogram to identify upturns and downturns in prices. The MACD-Histogram measures the difference between MACD and its signal line. The indicator is positive when MACD is above its signal line and negative when MACD is below its signal line. The MACD-Histogram turns positive when prices turn up and turns negative when prices turn down.

2.2.2 Relative Strength Index (RSI)

Definition

Relative Strength Index (RSI) is a momentum oscillator that measures the speed and change of price movements. RSI oscillates between zero and 100. Traditionally, and according to Wilder, RSI is considered overbought when above 70 and oversold when below 30.

Calculation

```
              100
RSI = 100 - --------
             1 + RS
```

RS = Average Gain / Average Loss

To simplify the calculation explanation, RSI has been broken down into its basic components: RS, Average Gain and Average Loss. This RSI calculation is based on 14 periods, which is the default suggested by Wilder in his book. Losses are expressed as positive values, not negative values.

The very first calculations for average gain and average loss are simple 14 period averages.

$$\begin{aligned} Average\ Gain &= [(previous\ Average\ Gain)x13 + current\ Gain]/14. \\ Average\ Loss &= [(previous\ Average\ Loss)x13 + current\ Loss]/14. \end{aligned}$$

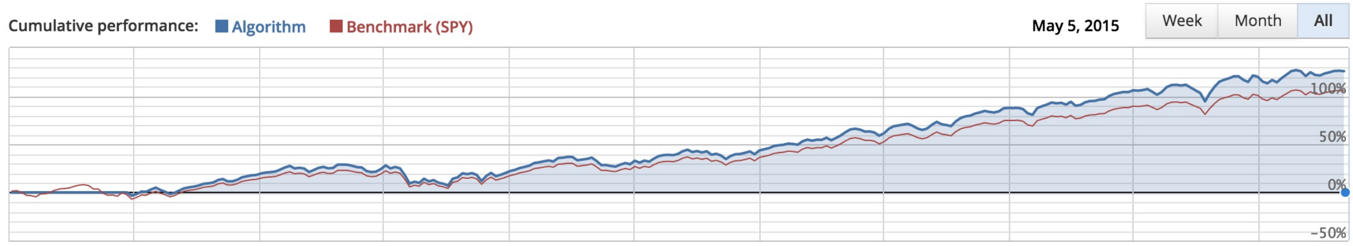
Signals can also be generated by looking for divergences, failure swings and centerline crossovers. RSI can also be used to identify the general trend.

3. Results

3.1 Momentum

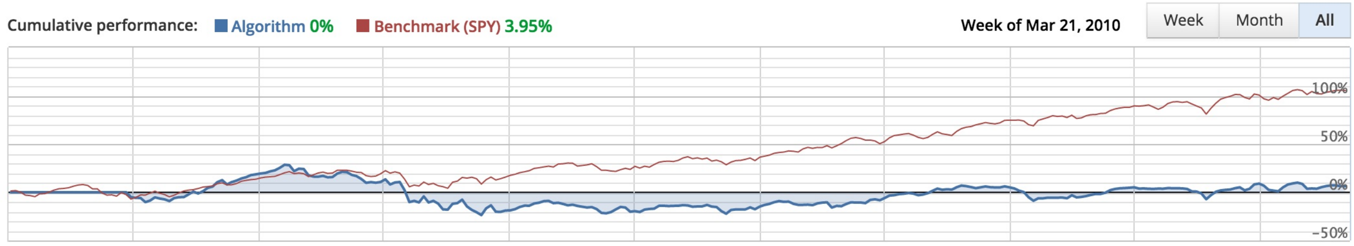
Linear Regression

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
124.2%	104%	0.05	0.94	1.32	1.67	0.24	0.16	21.1%



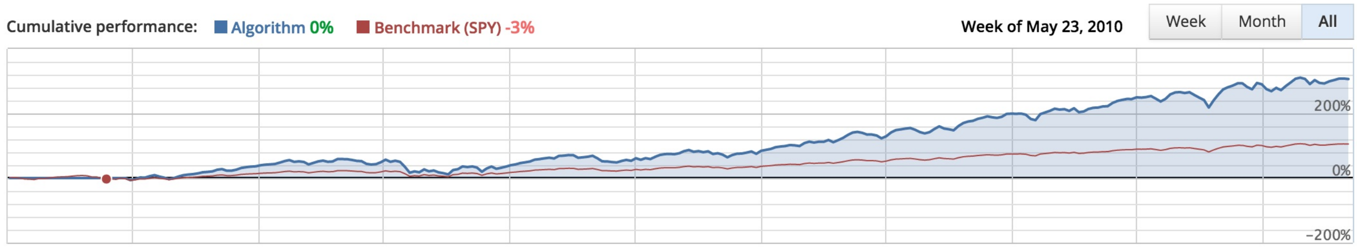
Random Forest

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
5.4%	104%	-0.18	0.98	-0.06	-0.06	-0.84	0.22	42%



Support Vector Machine

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
301.2%	104%	0.25	1.71	1.87	2.37	1.27	0.29	36.1%

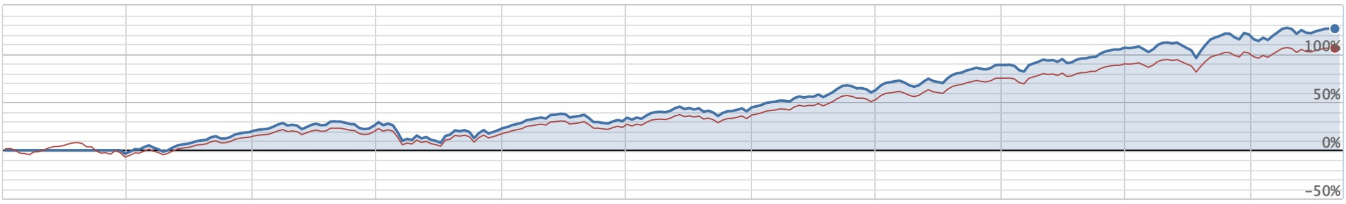


3.2 Relative Strength Index (RSI)

Linear Regression

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
126.89%	106.5%	0.05	0.94	1.34	1.69	0.24	0.16	21.1%

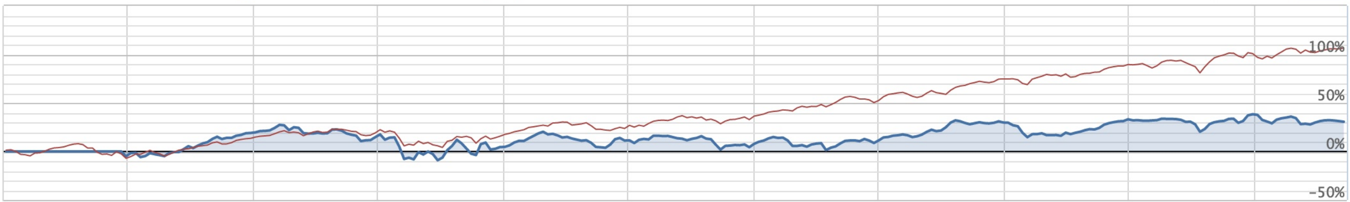
Cumulative performance: Algorithm 125.92% Benchmark (SPY) 105.7% Week of May 3, 2015 Week Month All



Random Forest

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
30.6%	106.5%	-0.14	0.96	0.16	0.19	-0.65	0.22	33.5%

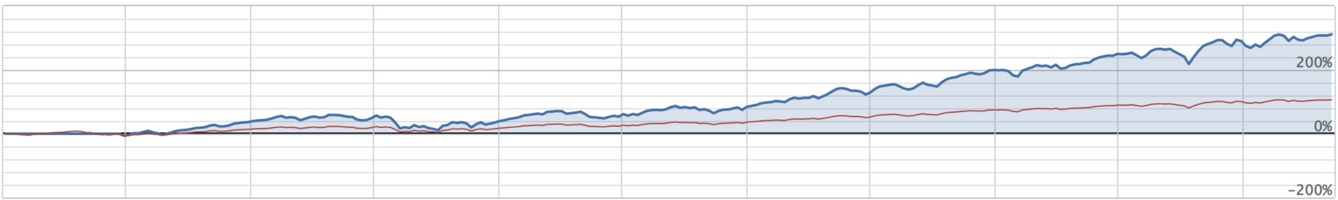
Cumulative performance: Algorithm 30.6% Benchmark (SPY) 105.15% Week Month All



Support Vector Machine

Total Returns	Benchmark Returns	Alpha	Beta	Sharpe	Sortino	Information Ratio	Volatility	Max Drawdown
311.2%	106.5%	0.26	1.71	1.93	2.44	1.32	0.29	36.1%

Cumulative performance: Algorithm 311.2% Benchmark (SPY) 105.15% May 11, 2015 Week Month All



3.3 Momentum with RSI

Linear Regression

Random Forest

Support Vector Machine

4. Conclusion

Sharpe Ratio of different learing algorithms with different trading strategies

Sharpe Ratio	Momentum	RSI	Momentum & RSI
Linear Regression	1.32	1.34	1.35
Random Forest	-0.06	0.16	1.93
Support Vector Machine	1.87	1.93	1.93

Learing Algorithms:

Support Vector Machine > Linear Regression > Random Forests

Trading Strategies:

Momentum + RSI > RSI > Momentum