2017

Developing an algorithmic trading strategy using supervised machine learning classification techniques and technical indicators

STUDENT: MAX FITZPATRICK

PROFESSOR: YVES JEGOUREL

M2 BANQUE, FINANCE & NEGOCE INTERNATIONAL | Université de Bordeaux

Summary

This research paper attempts to study the usefulness of popular technical indicators for predicting future price movements in S&P 500 equities. We train a unique random forest classifier on daily technical indicators derived from OHLC data from 01/01/2000 to 19/09/2014 for 75 S&P 500 equities and attempt to predict whether the stock will be up or down in price 30 trading days later. The classifier is incorporated into an algorithmic trading strategy which uses the up/down prediction as a buy/sell signal for the underlying security. Several strategies were created and tested, ranging from single equity strategies to directional long/short investment strategies.

The out of sample performance (01/01/2015 to 01/01/2017) is extremely promising and culminated in two long/short algorithms which beat the benchmark over the tested time-period while maintaining better risk metrics than a SPY buy and hold strategy. The profitability of these strategies simultaneously indicates that technical indicators do indeed have useful predictive power for future price movements, that a relevant feature set in conjunction with machine learning classification techniques can profitably predict price movements in stock markets, and that identifiable patterns and trends exist in financial markets.

Table of contents

Introduction 1
I) Model specifications 2
A) Feature selection 2
Delta / ROC 2
SMA10/20/100 2
Bollinger Bands 3
Stochastic oscillator 3
MACD 4
RSI 4
CCI 4
B) Classifier selection 5
K Nearest Neighbour (KNN) 5
Support Vector Machine classification (SVM) 6
Random Forest 9
Hybrid voting classifier 11
Results and final classifier selection 11
II) Trading algorithm development 13
A) Single equity strategy 14
B) Directional long/short strategies 15
Diversified long/short strategy 15
Confidence based long/short strategy 17
Conclusion 19
Annex 20
Annex I: Single equity strategy overview 20
Annex II: Diversified long/short performance metrics 21
Annex III: Confidence based long/short performance metrics 24
Annex IV: SPY buy and hold performance metrics for comparison 27
Bibliography 28
Source code 29
Tools used 29

Introduction

In market finance there are two main analysis styles, fundamental analysis and technical analysis. Fundamental analysis primarily seeks to examine accounting data, focusing on earnings, assets, quality, and strategic positioning, in an attempt to determine an equity's value by evaluating the underlying factors that affect the company's core business. In contrast, technical analysis is a methodology used for forecasting price levels or direction through the study of past market data, these indicators are generally mathematical transformations of previous price (OHLC) and/or volume data. Popular examples include moving averages, RSI, MACD, etc.

The efficacy and validity of both technical and fundamental analysis is disputed by the efficient market hypothesis which states that stock market prices are essentially unpredictable. Due to market efficiency current share prices already incorporate all relevant information. Furthermore, if we consider financial markets to be a random walk (a sequence of random variables) then it logically follows that the time series for any given equity can be considered to be a Martingale stochastic process for which, at a particular time (n) in the realised sequence, the expectation of the next value in the sequence ($n+1$) is equal to the present observed value even given knowledge of all prior observed values:

$$E(X_{n+1}|X_1, \dots, X_n) = X_n$$

Thus, the efficient market hypothesis and Martingale probability theory would indicate that technical analysis based on the examination of past data should be incapable of consistently inferring future price movements and generating alpha. However, if this is indeed true why do active management investment strategies and technical analysis continue to exist?

In order to study the efficacy of technical analysis and its associated indicators this paper attempts to create an algorithmic trading strategy which uses machine learning classification techniques with a range of technical indicators as inputs as a means of testing the predictive power of technical analysis. The majority of previous studies have focused on regression analysis where the researchers attempt to predict the future price of a security, we will instead be using classification in order to attempt to predict whether the price of a given stock will go up or down in the future.

I) Model specifications

In this first section we will be exploring our model specifications, namely our model inputs (features) and the different machine learning classifiers that we will be considering.

A) Feature selection

The inputs (features) for our classification model will be a range of popular and widely used technical indicators which we will then use to predict whether a stock's price will be up or down in 30 trading days. We aimed to use a variety of technical indicators which smooth price data, provide information about momentum, or indicate statistical variance. The technical indicators used in our feature set are listed below.

Delta / ROC

Delta or rate of change is simply the percentage change in the security's price over the previous x trading days, it is generally used as means of measuring momentum. Previous academic studies have found ROC to have significance when predicting future price movements.

$$Delta_t = \frac{P_t - P_{t-x}}{P_{t-x}}$$

$$where,$$

$$P_t \; is \; the \; price \; at \; time \; t$$

$$x \; is \; the \; length \; of \; the \; lookback \; period$$

SMA10/20/100

Our feature set contains 3 simple moving averages with a 10-day, 20-day, and 100-day lookback period. Simple moving averages (in addition to exponential moving averages and log transformations) are generally used to smooth price data and minimise noise. SMAs with different lookback periods are often used simultaneously in order to identify trends and trend reversals using SMA crossovers as trading signals.

$$SMA_n = \frac{1}{n}\sum_{i=1}^{n} P_i$$

$$where,$$

$$P_i \; is \; the \; price$$

$$n \; is \; the \; length \; of \; the \; lookback \; period$$

Bollinger Bands

Bollinger Bands are a tool invented by John Bollinger in the 1980s. Bollinger Bands can be used to measure the "highness" or "lowness" of the current price relative to previous trades. Bollinger Bands are essentially a volatility indicator. Bollinger Bands consist of a simple moving average enveloped by an upper and a lower band representing deviation from the mean. Our Bollinger bands use a 20-day lookback window, the formula is listed below:

$$B1 = 4\frac{MSD}{SMA}$$

$$B2 = \frac{P_t - SMA + 2MSD}{4MSD}$$

$$where,$$

$$SMA \; is \; a \; 20 \; day \; simple \; moving \; average$$

$$MSD \; is \; a \; 20 \; day \; rolling \; standard \; deviation$$

Stochastic oscillator

Stochastic Oscillator follows the speed or the momentum of the price. Generally, momentum changes before the price changes. It measures the level of the closing price relative to low-high range over a period of time. Our lookback period is 14 days.

$$SOk = 100 * \frac{(C - L14)}{(H14 - L14)}$$

$$where,$$

$$C \; is \; the \; current \; close$$

$$H14 \; is \; the \; highest \; high \; over \; the \; past \; 14 \; days$$

$$L14 \; is \; the \; lowest \; low \; over \; the \; past \; 14 \; days$$

MACD

MACD, short for moving average convergence/divergence, is a trading indicator created by Gerald Appel in the late 1970s. It supposedly reveals changes in the strength, direction, momentum, and duration of a trend in a stock's price. By comparing EMAs of different lengths, the MACD gauges changes in the trend of a stock. When the MACD goes below the signal line, it indicates a sell signal. When it goes above the signal line, it indicates a buy signal.

$$MACD = EMA_{12}(C) - EMA_{26}(C)$$

$$MACD\ signal\ line = EMA_{12}(MACD)$$

$$where,$$

$$C = closing\ price\ series$$

$$EMA_n = n\ day\ exponential\ moving\ average$$

RSI

RSI is a popular momentum indicator which determines whether the stock is overbought or oversold. A stock is said to be overbought when the demand unjustifiably pushes the price upwards. This condition is generally interpreted as a sign that the stock is overvalued and the price is likely to go down. A stock is said to be oversold when the price goes down sharply to a level below its true value. This is a result caused due to panic selling. RSI ranges from 0 to 100 and generally, when RSI is above 70, it may indicate that the stock is overbought and when RSI is below 30, it may indicate the stock is oversold.

$$RSI = 100 - \frac{100}{1 + RS}$$

$$RS = \frac{Average\ gain\ over\ the\ past\ 14\ days}{Average\ loss\ over\ the\ past\ 14\ days}$$

CCI

The commodity channel index (CCI) is essentially a momentum oscillator originally introduced by Donald Lambert in 1980 as a tool for identifying cyclical trends in commodity markets, it measures a security's variation from the statistical mean. It can also be used to identify overbought or oversold signals. Its use has become common in equity and currency markets despite its origins in commodity trading.

$$CCI_{20} = \frac{TP_t - SMA}{MSD}$$

$$TP_t = \frac{H_t + L_t + C_t}{3}$$

$$where,$$

$$SMA = 20\ day\ simple\ moving\ average\ of\ the\ close\ price$$

$$MSD = 20\ day\ moving\ standard\ deviation\ of\ the\ close\ price$$

$$H = high, L = low, C = close$$

Thus, our final feature set will contain 13 variables:

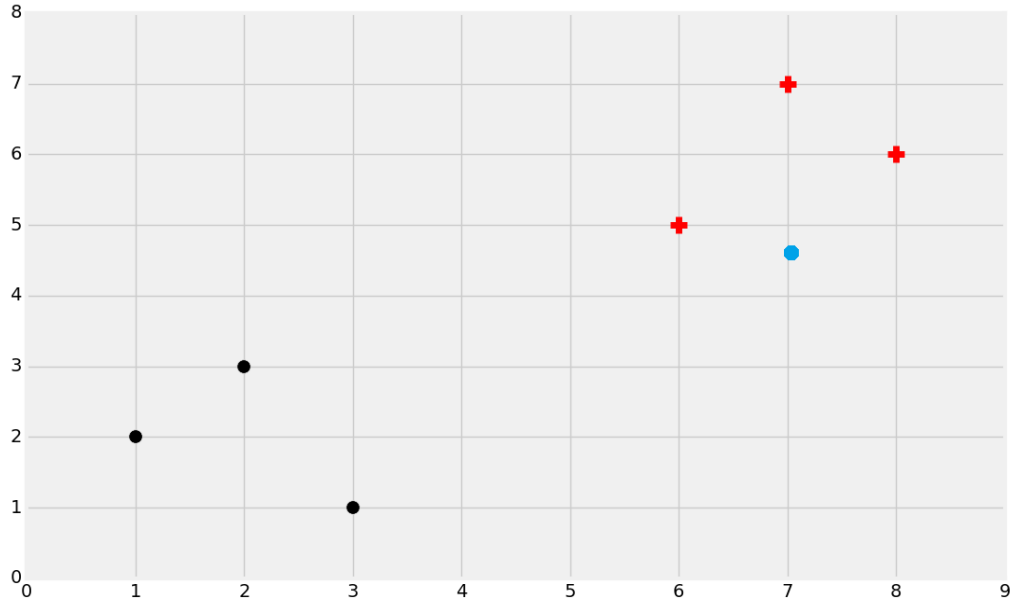
Delta	MA10	MA20	MA100	ROC
B1	B2	SOk	MACD	MACDsign
MACDdiff	RSI14	CCI		

B) Classifier selection

In this section we will explain the basic theory behind the machine learning classifiers that we will be evaluating before testing each one and selecting a classifier for our final model.

K Nearest Neighbour (KNN)

the KNN algorithm is a non-parametric supervised machine learning method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. In KNN classification an object is classified by a majority vote of its neighbours, with the object being assigned to the class most common among its *k* nearest neighbours (*k* is generally an odd positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of the single nearest neighbour. The KNN algorithm generally uses Euclidean distance as a means of computing the distances between data points depicted in the feature space.



As an example, the above graph depicts a two-dimensional feature space with two classes: black dots and red crosses. In this instance, if $k = 1$, the data point represented by the blue dot would be attributed to the "red cross" class because its nearest neighbour is a red cross.

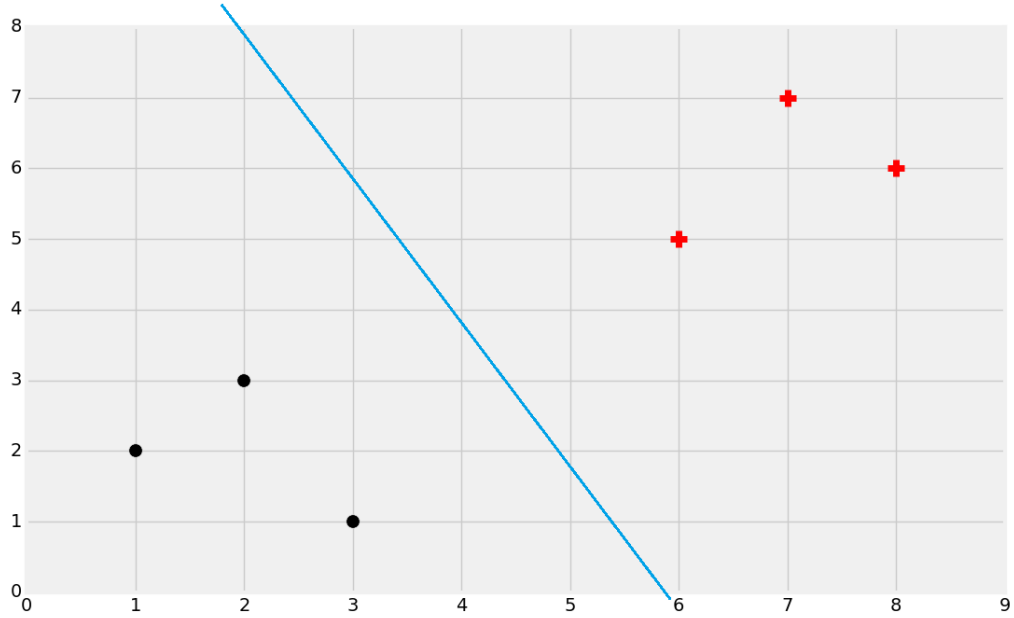
KNN is mathematically simple but computationally intensive, it works by computing the Euclidean distance between all points on a plane and assigning the majority class of the k nearest neighbours. Although the above example uses two-dimensional space, KNN can be scaled to n dimensions:

$$Euclidean\ distance = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

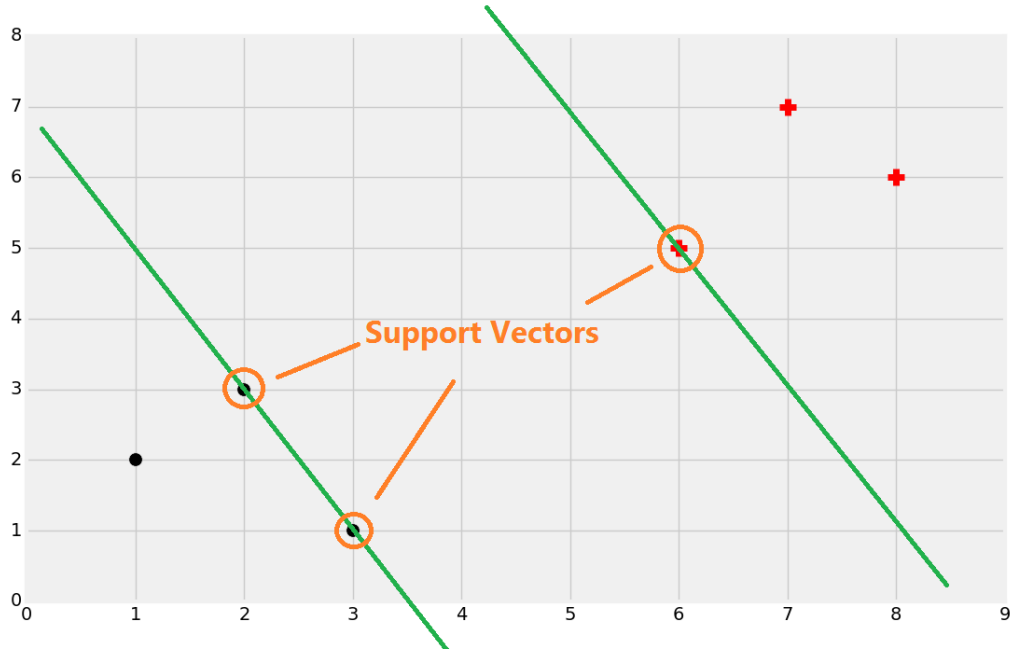
Support Vector Machine classification (SVM)

In this paper we used a linear Support Vector Machine and did not consider polynomial, rbf, or sigmoid kernels for generating hyperplanes.

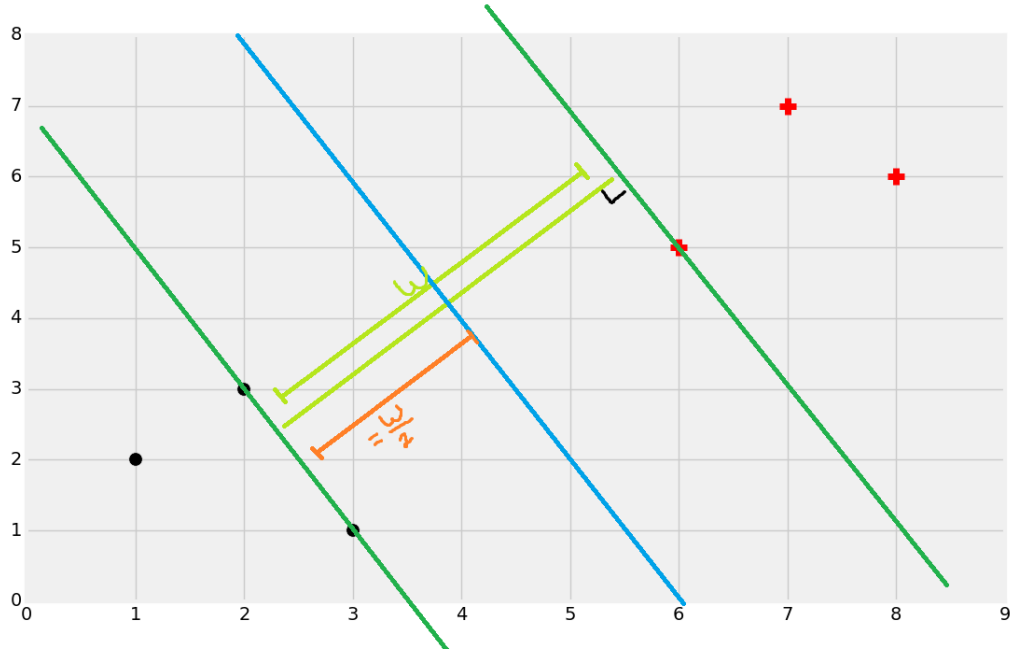
A support vector machine constructs a hyperplane in n dimensional space for n features, which can be used for classification. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class. In the case of support vector machines, a data point is viewed as a p-dimensional vector (a list of p feature inputs).



To use our previous two-dimensional example as an illustration, the bisecting blue line depicted above would be the hypothetical hyperplane, it allows us to linearly divide the data into two and constitutes the decision boundary for our classification. Any new data point falling to the right of the hyperplane would be classified as a "red cross" and any new data point falling to the left of the hyperplane would be classified as a "black dot." The hyperplane is constructed by identifying our support vectors, that is to say the data points which create an optimal distance between the different classes.



Once these support vectors have been identified through iteration, the hyperplane can be constructed in parallel at a distance of $\frac{w}{2}$ between the support vectors.



More formally, if we are given a training set of n observations in the following structure:

$$(\overrightarrow{x_1}, y_1), \dots, (\overrightarrow{x_n}, y_n)$$

Where $\overrightarrow{x_i}$ is a p-dimensional vector (depicting p features) and y_i is either 1 or -1 indicating the class to which the data point $\overrightarrow{x_i}$ belongs. We must construct a hyperplane that optimises the distance between the points $\overrightarrow{x_i}$ for which $y_i = 1$ and the group of $\overrightarrow{x_i}$ for which $y_i = -1$.

Thus, the optimal hyperplane is the set of points which satisfy the following constraints:

$$\vec{w} * \vec{x}_i + b \geq 1, for\ all\ x_i\ having\ the\ class\ 1$$

$$\vec{w} * \vec{x}_i + b \leq -1, for\ all\ x_i\ having\ the\ class\ -1$$

Which can be simplified to the following single constraint:

$$y_i(\vec{w} * \vec{x}_i + b) \geq 1, for\ any\ i = 1, \dots, n$$

Thus, solving for the optimal hyperplane is a Lagrange optimisation problem:

$$Minimize\ in\ (\vec{w}, b)$$

$$||w||$$

$$subject\ to: y_i(\vec{w} * \vec{x}_i + b) \geq 1, for\ any\ i = 1, \dots, n$$

Random Forest

Random forests belong to the ensemble family of machine learning classification techniques, a random forest is essentially a voting group of randomly generated decision trees which are each created from a different and random subsample of the training data. Random forests use bagging (Bootstrap Aggregating) to generate random subsamples to combine several weak learners (decision trees) into a single strong learner (forest).

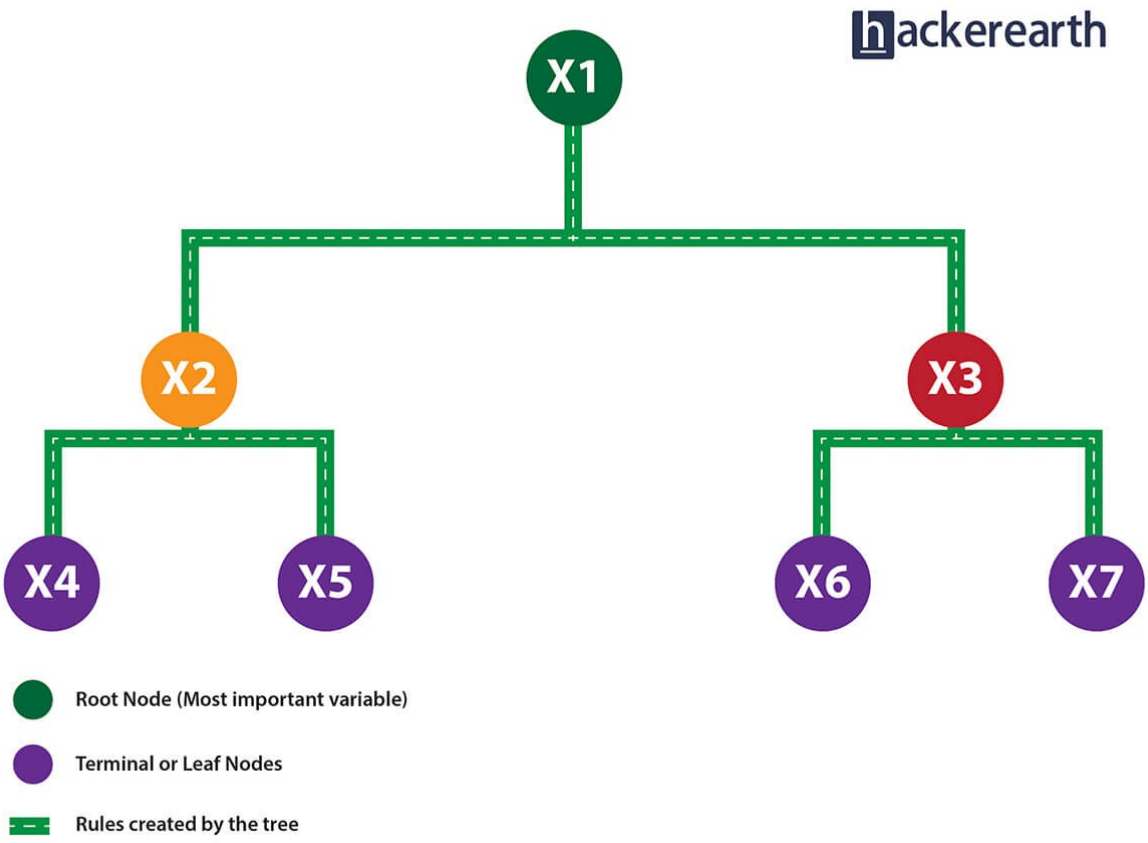
If we see our training data as a $n \times p$ matrix of n rows and p columns of which $p\text{-}1$ columns are features and the p^{th} column indicates the class of the observation:

$$training\ data = \begin{bmatrix} f_{A1} & f_{B1} & f_{C1} & f_{D1} & f_{p1} & C_1 \\ f_{A2} & f_{B2} & F_{C2} & f_{D2} & f_{p2} & C_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ f_{An} & f_{Bn} & f_{Cn} & f_{Dn} & f_{pn} & C_n \end{bmatrix}$$

For each decision tree in the forest of B (user defined) decision trees, the algorithm will randomly select a subset of k rows and P columns where $P < p$ for each decision tree, generally for classification: $P = \sqrt{p}$

$$decision\ tree\ 1 = \begin{bmatrix} f_{B14} & f_{D14} & C_{14} \\ f_{B2} & F_{D2} & C_2 \\ \vdots & \vdots & \vdots \\ f_{B25} & f_{D25} & C_{25} \end{bmatrix}$$

$$decision\ tree\ 2 = \begin{bmatrix} f_{E7} & f_{B7} & f_{G7} & C_7 \\ f_{E32} & f_{B32} & F_{G32} & C_{32} \\ \vdots & \vdots & \vdots & \vdots \\ f_{E17} & f_{B17} & f_{G17} & C_{17} \end{bmatrix}$$



Decision trees are then fitted from these subsamples, the data is partitioned using a sequence of if/else rules which divide the $k \times P$ subsample into distinct groups. The root node (X1) is assigned the feature which provides the highest homogeneity in child nodes (X2, X3). For a split to take place, the Gini coefficient for a child node should be less than that for the parent node. Splits will continue to take place until this is no longer true. Once a classification needs to be made, the result is the mode of the classifications provided by all of the decision trees in the forest.

Random forests present many advantages, they are robust to overfitting due to random feature selection, robust to insignificant features, and output a certainty score bound between 0 and 1 for any given classification. However, the main drawback of random forests is that they are somewhat of a "black-box" in that they do not provide interpretable results. At best a random forest can provide the importance of each feature, but it cannot offer any real insight into how its decisions are reached. It should be noted that although random forests are randomly generated, they can be replicated by specifying the seed given to the algorithm's random number generator.

Hybrid voting classifier

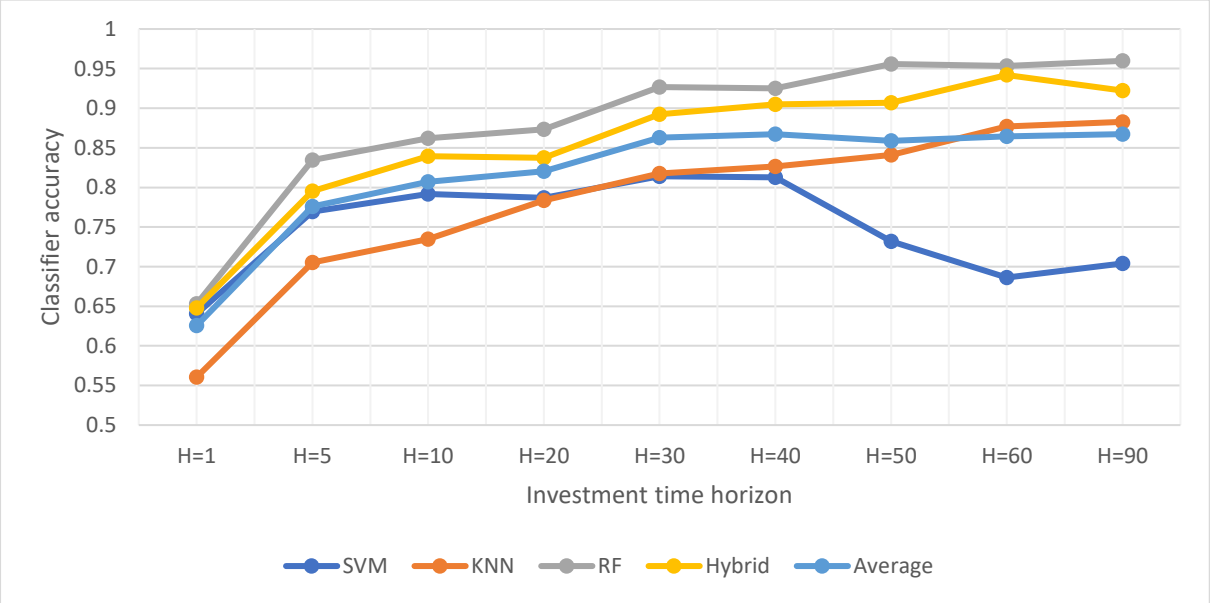
The final classifier tested is a hybrid of the KNN, SVM, and random forest. All three classifiers will be used simultaneously on a classification problem and the most common "vote" between the three classifiers will be retained.

Results and final classifier selection

Testing was performed on a dataset pulled from Quandl containing daily Apple (AAPL) OHLC data from 01/01/2000 to 19/09/2014, the dataset contains 3602 observations. This OHLC data was then used to generate our technical indicator feature set, a class of +1 was assigned if apple's stock price increased over the next 30 days and a class of -1 was assigned if apple's stock price decreased over the next 30 days. Thus, it is a binary classification problem.

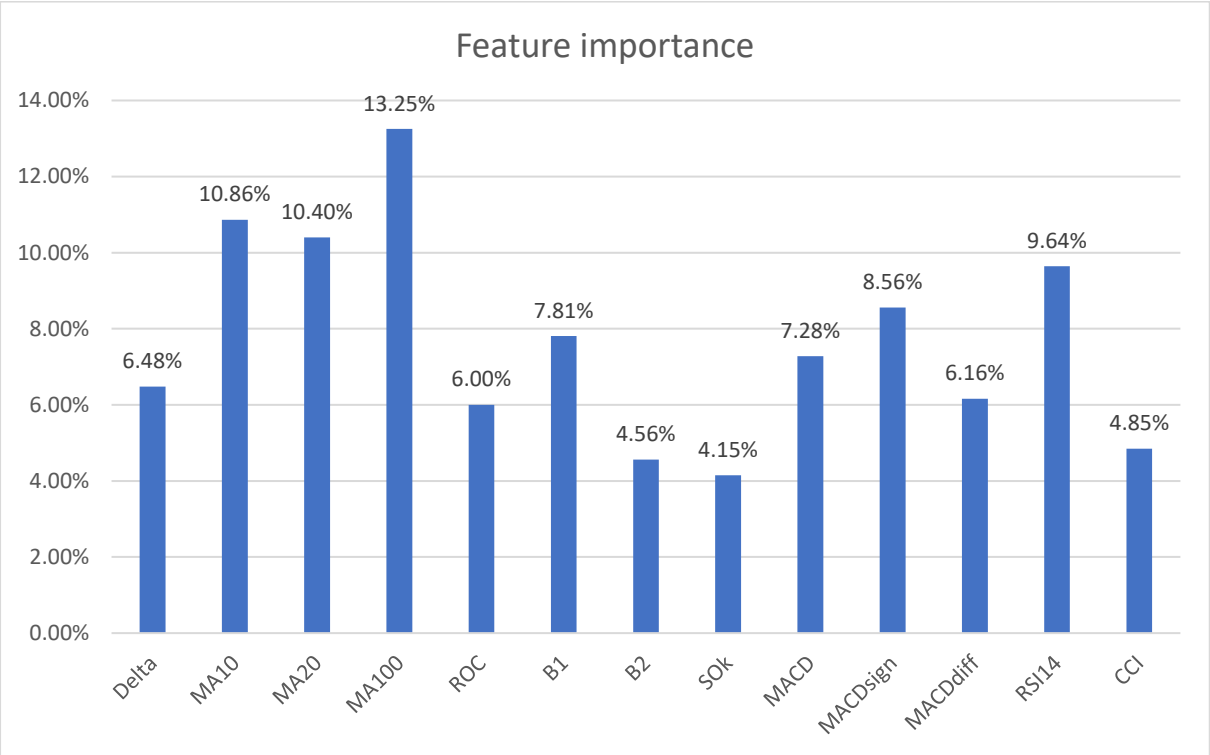
Regarding cross-validation, a randomly selected 67% of the dataset was used for training and the remaining 33% of the dataset was used for validation.

Regarding parameter optimisation, due to computational limitations very little optimisation was undertaken and we did not perform a comprehensive grid-search.



Classifier	Parameters
KNN	K = 5
SVM	Kernel = linear
Random Forest	Trees = 1000 Criterion = Gini coefficient Max depth = none
Hybrid voting	Weight = uniform

As we can see in the above graph, the random forest outperformed the other classifiers for all time horizons and we decided to proceed with this classifier. Furthermore, the random forest can be run in parallel and outputs a confidence score for its classifications which we will take advantage of in our final trading algorithm. Regarding the investment time horizon, we selected a holding period of 30 days as this allowed an acceptable amount of trades during the testing period while ensuring high classifier accuracy.



We can see that the classifier attributes the most importance to the moving averages, however there would appear to be no insignificant features in the dataset.

Now that we have identified our features and the classifier that will be used, in the next section we will use these to generate a buy/sell signal that will be incorporated into an algorithmic trading strategy.

II) Trading algorithm development

Algorithm development and out-of-sample backtesting were conducted on quantopian.com. *Quantopian* is a quant hedge fund which provides a development platform and free data to prospective developers, the best performing algorithms are then licenced by *Quantopian* and allocated funds for live trading.

Development takes place in a Python environment, backtesting and analysis specifically use the Zipline and Pyfolio libraries which were developed and open-sourced by *Quantopian*, data manipulation and classification are primarily performed using the Pandas, Numpy, and SKlearn libraries.

For our training algorithms the random forest classifiers are trained on daily data from 01/01/2000 to 19/09/2014 (n=3602), performance is then evaluated out-of-sample from 01/01/2015 to 01/01/2017.

Commissions are set to: $0.0075 per share with a minimum transaction cost of $1, the first fill will incur at least the minimum commission, and subsequent fills will incur additional commission. As a reference point, a retail *Interactive Brokers* account has a commission model of $0.005 per share and a minimum transaction cost of $1.

The slippage model used is: a maximum of 2.5% of the volume for any given minute and a price impact constant of 0.1 is used. From *Quantopian's* API documentation:

"For example: if the backtest is running in one-minute bars, and you place an order for 60 shares; then 1000 shares trade in each of the next several minutes; and the volume limit is 2.5%; then your trade order will be split into three orders (25 shares, 25 shares, and 10 shares). The price impact constant (0.1) defines how large of an impact your order will have on the backtester's price calculation. The slippage is calculated by multiplying the price impact constant by the square of the ratio of the order to the total volume. In our previous example, for the 25-share orders, the price impact is 0.1 * (25/1000) * (25/1000), or 0.00625%. For the 10-share order, the price impact is 0.1 * (10/1000) * (10/1000), or .001%."

Our universe was composed of 75 different S&P 500 equities which were chosen based on the availability of sufficient data for training (listed prior to 01/01/2000) and based on certain volume criteria so as to not be overly impacted by slippage while trading. A comprehensive list of the used equities is available in Annex I.

A) Single equity strategy

Our first trading strategy involves training a random forest classifier for a single equity and then trading that equity from 01/01/2015 to 01/01/2017 based on the trading signals outputted by the classifier. Initial capital is set to $100,000. The algorithm works as follows:

1. Declare global variables
2. Import training data from Google drive
3. Train the random forest classifier
4. The first position is taken on 02/01/2017 and the portfolio rebalances every 30 trading days.
5. Every 30 trading days, the algorithm fetches the last 100 days of OHLC data for the relevant equity and calculates the technical indicators used in our feature set.
6. The technical indicators for the previous day are input into the random forest classifier and it will output a 1 if it predicts that the price will be up in 30 days or a -1 if it predicts that the price will be down in 30 trading days. Thus, a 1 indicates that a long position should be taken and a -1 indicates that a short position should be taken.
7. The portfolio allocates 100% of its capital to the relevant position and holds that position until the next rebalance in 30 trading days.

This strategy was created and tested individually for the 75 different equities in our universe.

A comprehensive performance review can be seen in Annex I, the table below indicates the average performance over the 75 equities. The benchmark used is the SPDR S&P 500 ETF Trust ($SPY) which tracks the performance of the S&P 500 index.

Strategy return	SPY return	Buy & Hold return	Alpha	Beta
10.31%	13.39%	11.28%	0.05	0.05

Sharpe ratio	Sortino ratio	Volatility	Max drawdown %
0.18	0.3	0.25	-31.07%

We can see that, on average, the random forest classifier and technical indicator feature set are capable of generating positive alpha while remaining market neutral. However, the strategy fails to

beat the benchmark or a simple buy and hold strategy and is plagued by high volatility and drawdown. Furthermore, 100% of the portfolio's capital is invested into a single equity and there is no diversification which severely impacts the algorithm in the case of an incorrect prediction. These factors obviously significantly impact the strategy's Sharpe ratio, which is a measure of risk adjusted return. It should also be noted that although the benchmark had a cumulative return of 13.39% over the backtesting period, if one had allocated 100% of portfolio capital ($100,000) to the SPY on 02/01/2015 then the actual cumulative return, net of commissions and slippage, would be 13.15%. Similarly, the average buy and hold return, net of commissions and slippage, would be lower than 11.28%.

Furthermore, after an examination of the log outputs, it became clear that the algorithm was often trading based off of weak certainty. The algorithm, by design, would trade at every rebalancing period regardless of whether the outputted class had a certainty of 0.51 or 0.99. We feel that this, coupled with the lack of diversification, heavily impacted results and it led to the development of two directional long/short strategies which trade either all of the universe's equities or a subsample of equities based on the classifier's confidence.

B) Directional long/short strategies

Diversified long/short strategy

The first directional long/short strategy was created in order to allow a high level of diversification in an attempt to improve both returns and risk adjusted returns. A separate classifier is trained for each equity in our universe and a long or short position is taken in each equity depending on the prediction of its classifier.

Considering the increased amount of securities being traded, starting capital was increased to $1,000,000. Furthermore, due to memory limitations the training data for each equity was trimmed from 3602 observations to 3500 observations.

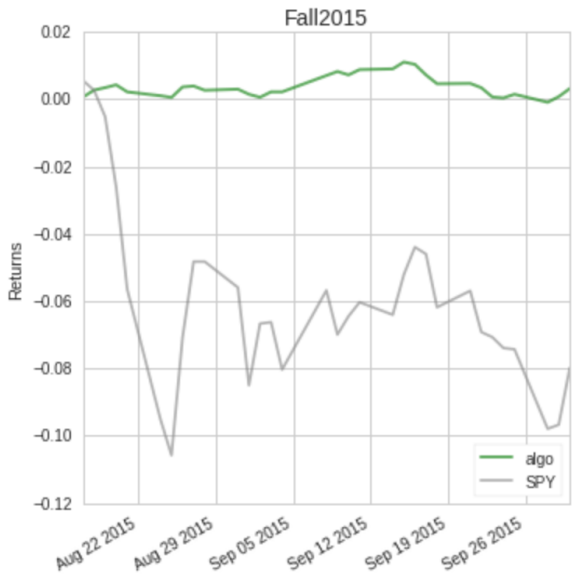
The trading algorithm works as follows:

1. Declare global variables
2. Import all training data from dropbox, this is a single csv file which contains the training set for each of the 75 equities in order. Equity 1 is columns 1-14, equity 2 is columns 15-28, etc.

3. Loop through the training data, train and store a separate random forest classifier for each equity on its relevant training data.
4. The first position is taken on 05/01/2015 and the portfolio rebalances every 30 trading days
5. When rebalancing the portfolio, the algorithm initially loops through every equity in our universe, pulls its OHLC data for the last 100 days, calculates the technical indicators for our feature set and inputs the previous day's observations into its classifier, each classifier then outputs a prediction.
6. The algorithm computes the portfolio weights ($\pm \frac{1}{number\ of\ shortlisted\ stocks}$) based on the predictions for the new positions and readjusts its allocations accordingly.

As we can see in the image below, the implementation of diversification lead to improvements across the board. The long/short strategy managed to beat the benchmark while maintaining impressive risk ratios and minimal volatility and drawdown, in addition to remaining market neutral.

Further investigation of the algorithm's performance reveals low correlation to Fama-French beta factors, a positive mean for monthly returns, a stable annual return for both 2015 and 2016, and a probability of making a profitable trade that it is greater than 0.5. Furthermore, the algorithm successfully preserved capital and proved to be robust during the two stress events that took place during the backtesting period.



A more comprehensive offering of performance metrics is available in Annex II

Confidence based long/short strategy

The second directional long/short strategy was created to ensure that all trades satisfied a minimum confidence threshold in an attempt to increase returns and minimise the volatility and drawdown exhibited by the single equity strategies. We chose 92.5% as our confidence threshold. In light of the increased amount of securities being traded, starting capital was increased to $1,000,000. Furthermore, due to memory limitations the training data for each equity was trimmed from 3602 observations to 3500 observations.

The trading algorithm works as follows:

1. Declare global variables
2. Import training data from dropbox, this is a single csv file which contains the training set for each of the 75 equities in order. Equity 1 is columns 1-14, equity 2 is columns 15-28, etc.
3. Loop through the training data, train and store a separate random forest classifier for each equity on its relevant training data.
4. The first position is taken on 05/01/2015 and the portfolio rebalances every 30 trading days
5. When rebalancing the portfolio, the algorithm loops through every equity in our universe, pulls its OHLC data for the last 100 days, calculates the technical indicators for our feature

set and inputs the previous day's observations into its classifier, each classifier then outputs a prediction.

6. Each stock's classifier will output a confidence score indicating its confidence of taking a long position, these confidence scores are then appended to a dataframe.
7. The algorithm searches the dataframe then takes any stock for which the confidence of a long position is >0.925 or <0.075 (indicating over 92.5% confidence of a short position) and adds it to our final trading shortlist.
8. The algorithm computes the portfolio weights ($\pm \frac{1}{number\ of\ shortlisted\ stocks}$) based on the predictions for the new positions and readjusts its allocations accordingly.

As the above image indicates, this strategy represented a marked improvement over our previous single equity strategies, successfully increasing returns in excess of the benchmark and more than halving volatility and drawdown. As a consequence of these improved metrics our Sharpe and Sortino ratios increased relative to the single equity strategies, indicating an improved risk adjusted return. However, although our absolute performance improved relative to the previous long/short strategy, this algorithm exhibited much higher volatility and drawdown and its risk metrics are justifiably lower.

The confidence based strategy is capable of generating impressive returns but our universe isn't big enough to ensure adequate diversification and risk management. On average, the algorithm's portfolio only contained 6 equities, and at one point only contained 2 equities, which heavily contributed to its volatility. That being said, the confidence long/short algorithm exhibited less volatility and drawdown than a SPY buy and hold strategy over the same time period. Furthermore, the confidence long/short algorithm exhibited a positive mean monthly return, a probability of making a profitable trade that it is greater than 0.5, and successfully navigated the "Fall 2015" and "New Normal" stress events that took place during the backtest period.

A more comprehensive offering of performance metrics is available in Annex III.

Conclusion

Through the development of these trading algorithms we have shown that profitable and alpha generating strategies are possible when using a feature set composed solely of technical indicators. This would imply that technical indicators are indeed relevant for predicting future price movements. The predictive power of these indicators implies that momentum and statistical variance are not necessarily integrated into a stock's current price. This conclusion brings into question the efficient markets hypothesis in addition to the presumption that financial markets are essentially a random walk and that stock prices can be thought of as a Martingale stochastic process. It indicates that detectable, cyclical, and recurrent patterns and trends exist in financial markets. Furthermore, we have shown that random forest classifiers exhibit significant predictive capacity for financial time series.

Moreover, the long/short trading algorithms that we developed can be enhanced further with more time and computational resources, the proposed additions would be the following:

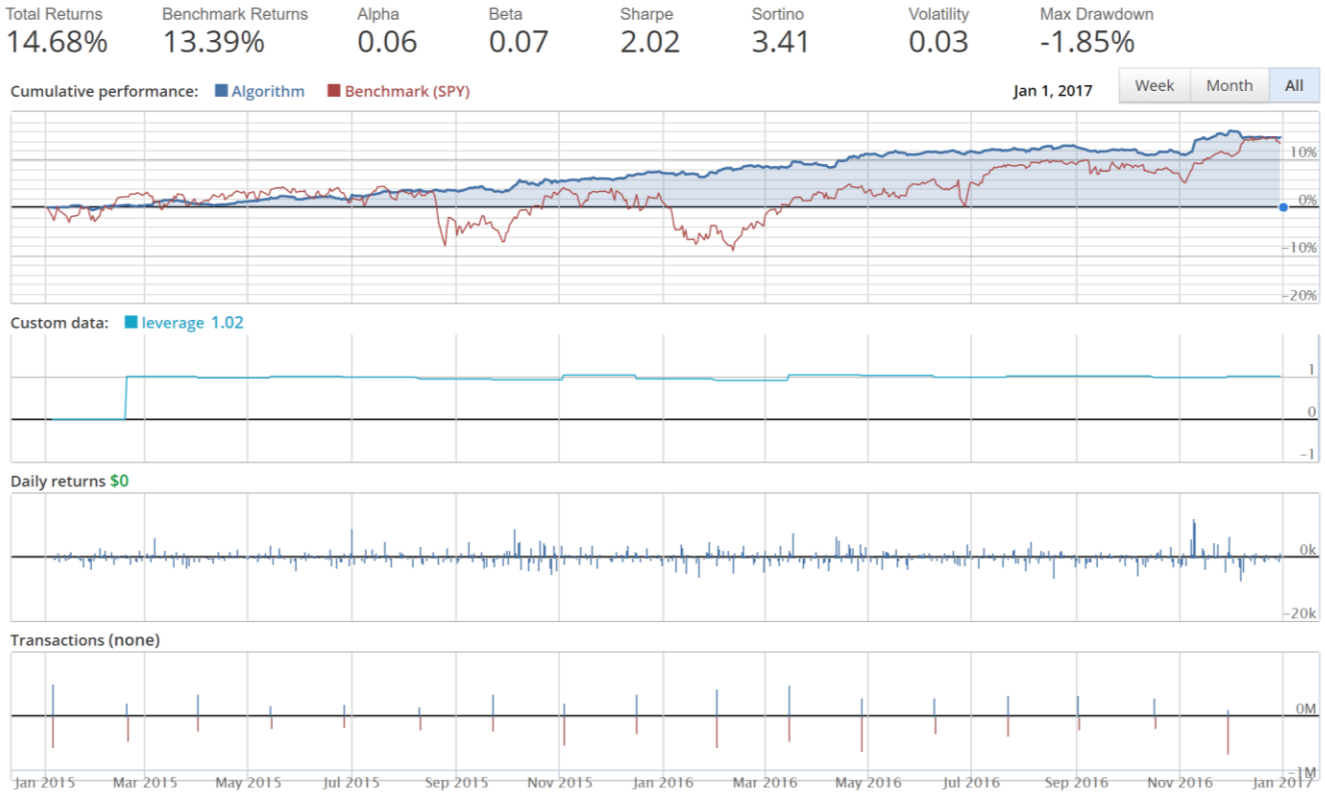
- A comprehensive grid search for parameter optimisation for each random forest classifier
- Retraining of the classifier to incorporate new observations at each rebalancing period
- Increasing the size of the universe
- Increasing the size of the datasets
- Increasing the size of the feature sets
- Implementing trailing stop-losses
- Dynamic universe selection: selecting stocks based on recent volume and excluding stocks which have earnings reports during the holding period in order to minimise slippage and volatility
- Calculating portfolio weights as a function of VaR or through Markowitz portfolio optimisation

Annex

Annex I: Single equity strategy overview

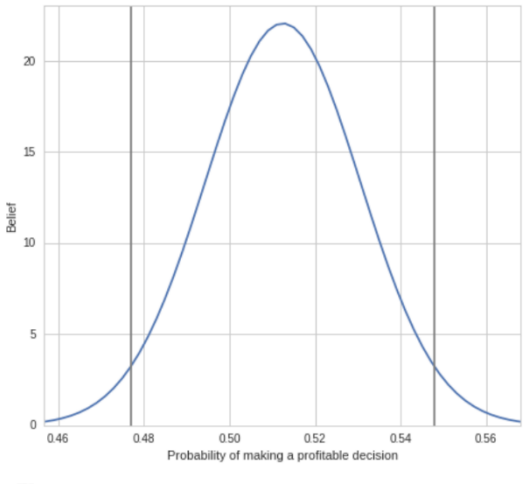
Ticker	Name	Sector	Strategy return	Buy and hold return	SPY return	Return vs SPY	Return vs buy and hold	Alpha	Beta	Sharpe	Sortino	Volatility	Max drawdown
AAPL	Apple Inc.	Information Technology	-10.50%	9.05%	13.39%	-23.89%	-19.55%	0.02	-0.62	-0.11	-0.16	0.24	-28.39%
ABT	Abbott Laboratories	Health Care	2.62%	-10.66%	13.39%	-10.77%	13.28%	0.05	0.48	0.17	0.24	0.22	-25.73%
ADM	Archer-Daniels-Midland C	Consumer Staples	20.45%	-7.14%	13.39%	7.06%	27.59%	0.12	-0.17	0.49	0.75	0.25	-27.60%
ADP	Automatic Data Processir	Information Technology	-7.23%	29.38%	13.39%	-20.62%	-36.61%	-0.09	0.5	-0.11	-0.16	0.18	-27.20%
ADSK	Autodesk Inc	Information Technology	-2.61%	23.20%	13.39%	-16.00%	-25.81%	0.07	-0.18	0.12	0.18	0.32	-36.09%
AET	Aetna Inc	Health Care	-0.66%	42.23%	13.39%	-14.05%	-42.89%	0.16	-0.59	0.12	0.17	0.26	-37.22%
AIG	American International G	Financials	-36.80%	20.89%	13.39%	-50.19%	-57.69%	-0.17	-0.3	-1	-1.29	0.21	-45.33%
AMGN	Amgen Inc	Health Care	36.25%	-3.98%	13.39%	22.86%	40.23%	0.18	0.51	0.72	1.04	0.26	-18.52%
AXP	American Express Co	Financials	15.31%	-17.69%	13.39%	1.92%	33.00%	0.15	0.81	0.43	0.59	0.23	-37.58%
BAC	Bank of America Corp	Financials	1.77%	27.29%	13.39%	-11.62%	-25.52%	0.06	-0.08	0.17	0.25	0.28	-42.28%
BAX	Baxter International Inc.	Health Care	25.23%	15.17%	13.39%	11.84%	10.06%	0.24	-0.82	0.47	1.24	0.35	-36.74%
BBY	Best Buy Co. Inc.	Consumer Discretionary	72.90%	16.52%	13.39%	59.51%	56.38%	0.33	0.03	0.95	1.55	0.36	-27.94%
BDX	Becton Dickinson	Health Care	-21.40%	22.99%	13.39%	-34.79%	-44.39%	0.02	-0.99	-0.56	-0.81	0.18	-30.94%
BLL	Ball Corp	Materials	12.76%	11.78%	13.39%	-0.63%	0.98%	0.03	0.73	0.37	0.58	0.24	-18.07%
CA	CA, Inc.	Information Technology	-12.17%	11.65%	13.39%	-25.56%	-23.82%	0.03	-0.96	-0.22	-0.33	0.2	-29.10%
CAG	Conagra Brands	Consumer Staples	-20.77%	14.41%	13.39%	-34.16%	-35.18%	0.04	-1	-0.25	-0.44	0.3	-37.16%
CI	CIGNA Corp.	Health Care	-6.27%	29.91%	13.39%	-19.66%	-36.18%	0.01	-0.04	0.02	0.02	0.27	-40.85%
CLX	The Clorox Company	Consumer Staples	-11.25%	21.33%	13.39%	-24.64%	-32.58%	-0.05	0.02	-0.3	-0.4	0.16	-28.84%
CSCO	Cisco Systems	Information Technology	-20.43%	16.00%	13.39%	-33.82%	-36.43%	0.01	-0.96	-0.41	-0.55	0.22	-33.61%
CSX	CSX Corp.	Industrials	-41.65%	4.16%	13.39%	-55.04%	-45.81%	-0.25	0.38	-0.87	-1.16	0.27	-50.17%
DIS	The Walt Disney Compan	Consumer Discretionary	-17.99%	13.77%	13.39%	-31.38%	-31.76%	0	-0.98	-0.4	-0.6	0.2	-27.14%
DOV	Dover Corp.	Industrials	21.51%	9.78%	13.39%	8.12%	11.73%	0.06	0.91	0.5	0.74	0.26	-31.04%
ECL	Ecolab Inc.	Materials	13.68%	14.85%	13.39%	0.29%	-1.17%	0.05	0.36	0.42	0.59	0.2	-17.65%
EMN	Eastman Chemical	Materials	-21.10%	4.25%	13.39%	-34.49%	-25.35%	-0.08	0.04	-0.28	-0.37	0.28	-57.57%
EMR	Emerson Electric Compan	Industrials	-31.17%	-2.81%	13.39%	-44.56%	-28.36%	-0.17	0.41	-0.72	-0.97	0.23	-45.97%
FDX	FedEx Corporation	Industrials	-35.62%	8.82%	13.39%	-49.01%	-44.44%	-0.23	0.5	-0.86	-1.07	0.23	-52.94%
GIS	General Mills	Consumer Staples	-22.74%	23.10%	13.39%	-36.13%	-45.84%	0.01	-1.01	-0.67	-0.92	0.17	-34.57%
GPC	Genuine Parts	Consumer Discretionary	-12.42%	-5.30%	13.39%	-25.81%	-7.12%	-0.04	0.77	-0.25	-0.34	0.19	-30.08%
GPS	Gap Inc.	Consumer Discretionary	-21.28%	-42.62%	13.39%	-34.67%	21.34%	-0.07	-0.08	-0.13	-0.19	0.38	-59.41%
GWW	Grainger (W.W.) Inc.	Industrials	16.30%	-4.90%	13.39%	2.91%	21.20%	0.1	0.63	0.46	0.67	0.21	-20.70%
HON	Honeywell Int'l Inc.	Industrials	-1.18%	21.66%	13.39%	-14.57%	-22.84%	0.08	-0.56	0.06	0.09	0.18	-20.49%
HRB	Block H&R	Financials	0.10%	-27.44%	13.39%	-13.29%	27.54%	0.1	0.49	0.16	0.22	0.31	-43.86%
INTC	Intel Corp.	Information Technology	-3.53%	6.39%	13.39%	-16.92%	-9.92%	0.04	-0.54	0.04	0.06	0.23	-30.32%
ITW	Illinois Tool Works	Industrials	-24.22%	35.22%	13.39%	-37.61%	-59.44%	-0.12	0.01	-0.68	-0.94	0.18	-36.42%
JNJ	Johnson & Johnson	Health Care	-11.92%	16.70%	13.39%	-25.31%	-28.62%	-0.08	0.32	-0.35	-0.48	0.15	-20.27%
JWN	Nordstrom	Consumer Discretionary	-30.87%	-31.93%	13.39%	-44.26%	1.06%	0	0.95	-0.38	-0.49	0.34	-51.35%
KEY	KeyCorp	Financials	44.13%	37.79%	13.39%	30.74%	6.34%	0.11	0.54	0.79	1.21	0.28	-32.57%
LB	L Brands Inc.	Consumer Discretionary	-13.26%	-19.52%	13.39%	-26.65%	6.26%	0	0.54	-0.14	-0.19	0.26	-41.48%
LLY	Lilly (Eli) & Co.	Health Care	-10.37%	12.43%	13.39%	-23.76%	-22.80%	-0.07	0.5	-0.08	-0.12	0.26	-38.74%
LMT	Lockheed Martin Corp.	Industrials	71.55%	37.65%	13.39%	58.16%	33.90%	0.2	0.5	1.68	2.84	0.17	-9.06%
LNC	Lincoln National	Financials	68.74%	19.84%	13.39%	55.35%	48.90%	0.34	-0.16	0.99	1.47	0.32	-42.04%
LOW	Lowe's Cos.	Consumer Discretionary	-13.15%	6.69%	13.39%	-26.54%	-19.84%	-0.05	0.02	-0.23	-0.32	0.21	-31.84%
LUV	Southwest Airlines	Industrials	15.17%	19.71%	13.39%	1.78%	-4.54%	0.02	0.69	0.38	0.54	0.31	-30.91%
MAT	Mattel Inc.	Consumer Discretionary	18.10%	-0.63%	13.39%	4.71%	18.73%	0.1	0.78	0.42	0.65	0.3	-32.34%
MDT	Medtronic plc	Health Care	2.85%	2.77%	13.39%	-10.54%	0.08%	0.01	0.85	0.17	0.23	0.2	-18.99%
MMC	Marsh & McLennan	Financials	-5.57%	23.25%	13.39%	-18.96%	-28.82%	-0.07	0.5	-0.1	-0.13	0.16	-18.24%
MRO	Marathon Oil Corp.	Energy	-2.71%	-35.78%	13.39%	-16.10%	33.07%	0.17	0.56	0.26	0.37	0.56	-61.80%
MSFT	Microsoft Corp.	Information Technology	13.09%	41.18%	13.39%	-0.30%	-28.09%	-0.05	0.68	0.37	0.55	0.25	-17.13%
MU	Micron Technology	Information Technology	41.37%	-37.41%	13.39%	27.98%	78.78%	0.25	-0.39	0.6	0.9	0.49	-42.28%
NEM	Newmont Mining Corpor	Materials	232.70%	81.99%	13.39%	219.31%	150.71%	0.59	0.3	1.51	2.44	0.47	-25.47%
NKE	Nike	Consumer Discretionary	-11.09%	8.19%	13.39%	-24.48%	-19.28%	-0.04	0.13	-0.15	-0.21	0.22	-38.86%
NOC	Northrop Grumman Corp	Industrials	62.99%	63.50%	13.39%	49.60%	-0.51%	0	1	1.42	2.3	0.18	-10.59%
NTAP	NetApp	Information Technology	-11.71%	-10.79%	13.39%	-25.10%	-0.92%	-0.01	-0.09	-0.03	-0.04	0.32	-47.83%
NUE	Nucor Corp.	Materials	35.94%	29.53%	13.39%	22.55%	6.41%	0.26	-0.37	0.67	0.93	0.3	-40.09%
NWL	Newell Brands	Consumer Discretionary	40.74%	21.45%	13.39%	27.35%	19.29%	0.17	0.2	0.81	1.23	0.25	-26.67%
ORCL	Oracle Corp.	Information Technology	62.98%	-11.95%	13.39%	49.59%	74.93%	0.23	-0.69	1.34	2.19	0.2	-9.57%
OXY	Occidental Petroleum	Energy	-15.04%	-4.05%	13.39%	-28.43%	-10.99%	-0.05	0.29	-0.18	-0.26	0.26	-29.55%
PCAR	PACCAR Inc.	Industrials	64.08%	0.30%	13.39%	50.69%	63.78%	0.29	-0.26	1.13	1.72	0.24	-22.05%
PH	Parker-Hannifin	Industrials	-18.20%	13.57%	13.39%	-31.59%	-31.77%	-0.05	-0.22	-0.32	-0.45	0.23	-38.11%
PKI	PerkinElmer	Health Care	-11.54%	20.62%	13.39%	-24.93%	-32.16%	0.07	-0.88	-0.18	-0.25	0.22	-26.86%
PX	Praxair Inc.	Materials	27.90%	-4.76%	13.39%	14.51%	32.66%	0.14	0.31	0.78	1.12	0.18	-13.51%
SHW	Sherwin-Williams	Materials	-4.14%	4.53%	13.39%	-17.53%	-8.67%	-0.02	0.56	0.02	0.03	0.23	-29.46%
SLB	Schlumberger Ltd.	Energy	-17.81%	3.39%	13.39%	-31.20%	-21.20%	-0.05	-0.35	-0.27	-0.38	0.25	-36.27%
SNA	Snap-On Inc.	Consumer Discretionary	4.12%	29.11%	13.39%	-9.27%	-24.99%	-0.01	0.36	0.2	0.3	0.2	-22.14%
STI	SunTrust Banks	Financials	-5.35%	37.03%	13.39%	-18.74%	-42.38%	-0.15	0.8	0.03	0.04	0.27	-41.04%
SWK	Stanley Black & Decker	Consumer Discretionary	-24.67%	24.43%	13.39%	-38.06%	-49.10%	0	-0.96	-0.63	-0.86	0.19	-34.65%
SYY	Sysco Corp.	Consumer Staples	46.25%	46.79%	13.39%	32.86%	-0.54%	0.06	0.69	1.1	1.95	0.19	-11.58%
TGT	Target Corp.	Consumer Discretionary	-9.45%	1.16%	13.39%	-22.84%	-10.61%	0	-0.97	-0.13	-0.19	0.21	-21.64%
TJX	TJX Companies Inc.	Consumer Discretionary	-12.70%	12.37%	13.39%	-26.09%	-25.07%	0.03	-0.99	-0.23	-0.32	0.2	-24.44%
TXT	Textron Inc.	Industrials	0.57%	15.76%	13.39%	-12.82%	-15.19%	0.03	0.03	0.14	0.21	0.25	-27.01%
UNH	United Health Group Inc.	Health Care	62.41%	63.77%	13.39%	49.02%	-1.36%	0	1	1.2	1.86	0.22	-12.70%
WFC	Wells Fargo	Financials	24.93%	6.56%	13.39%	11.54%	18.37%	0.15	-0.38	0.64	0.96	0.21	-16.05%
WMB	Williams Cos.	Energy	171.97%	-20.13%	13.39%	158.58%	192.10%	0.69	-0.47	1.22	1.91	0.52	-35.27%
WMT	Wal-Mart Stores	Consumer Staples	-3.81%	-14.87%	13.39%	-17.20%	11.06%	0.01	0.12	0	0	0.19	-18.38%
XOM	Exxon Mobil Corp.	Energy	38.14%	4.56%	13.39%	24.75%	33.58%	0.18	-0.05	0.9	1.31	0.2	-14.07%
Total		**75**	**10.31%**	**11.28%**	**13.39%**	**-3.08%**	**-0.97%**	**0.054533**	**0.0492**	**0.176267**	**0.304667**	**0.252133333**	**-31.07%**

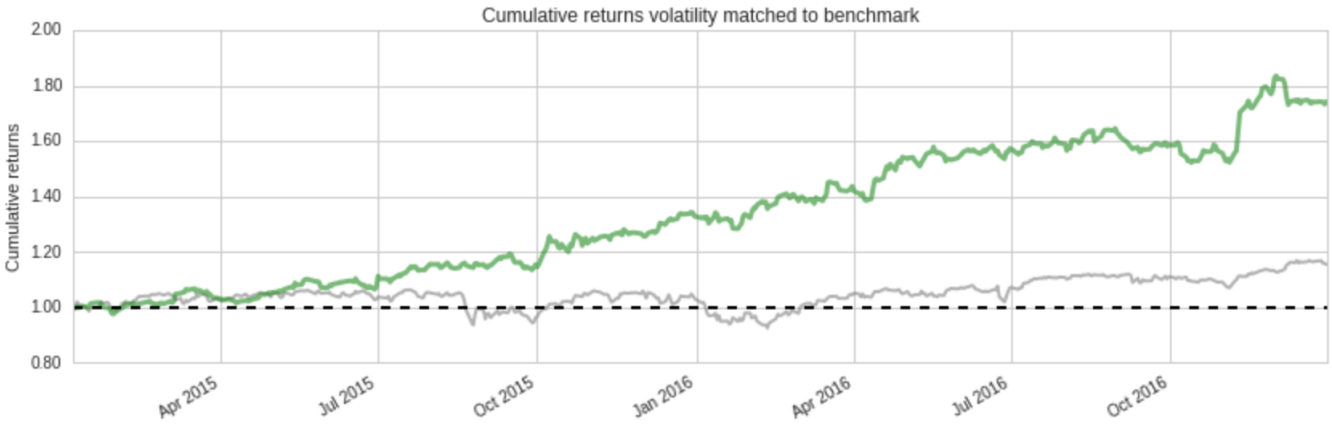
Annex II: Diversified long/short performance metrics

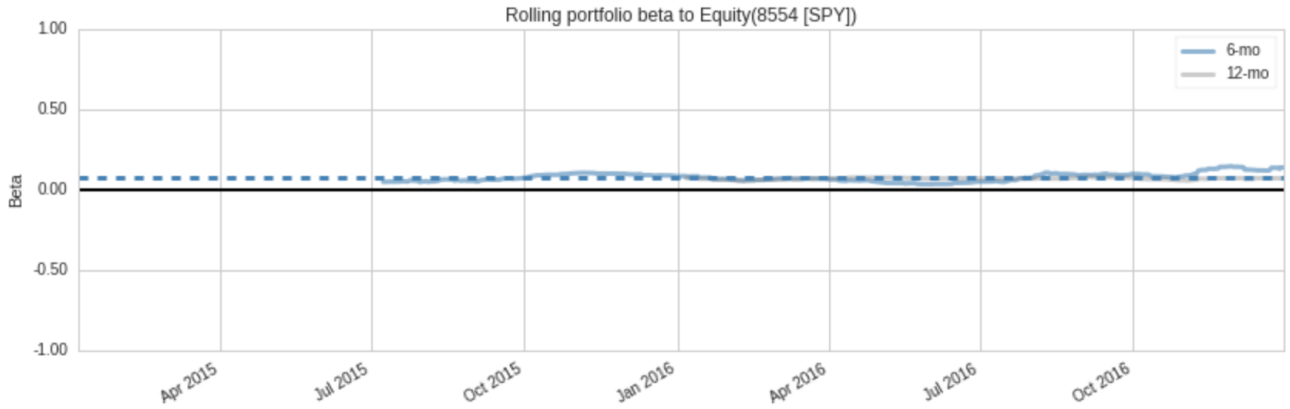


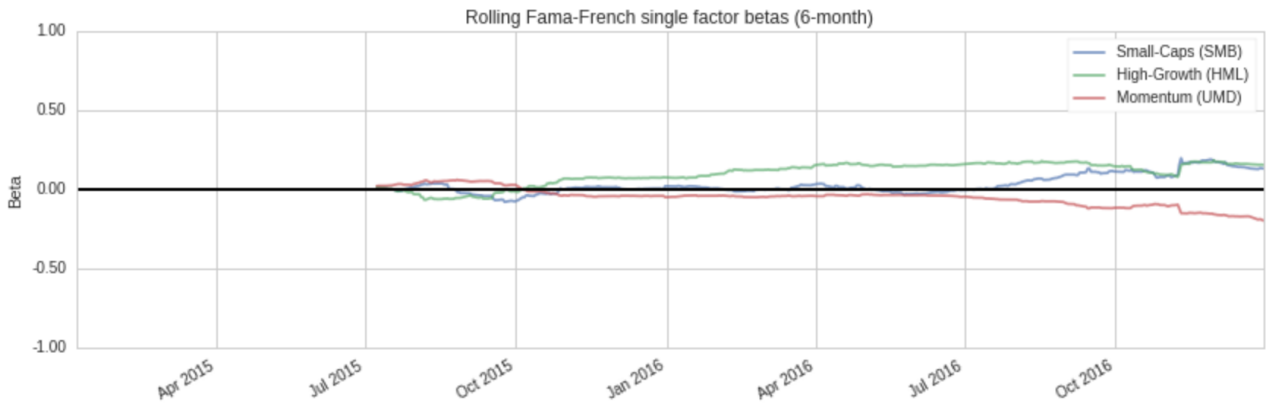
Performance statistics	Backtest
annual_return	0.07
cum_returns_final	0.15
annual_volatility	0.03
sharpe_ratio	2.02
calmar_ratio	3.84
stability_of_timeseries	0.97
max_drawdown	-0.02
omega_ratio	1.41
sortino_ratio	3.41
skew	0.71
kurtosis	2.83
tail_ratio	1.29
common_sense_ratio	1.38
gross_leverage	0.99
information_ratio	-0.00
alpha	0.06
beta	0.07

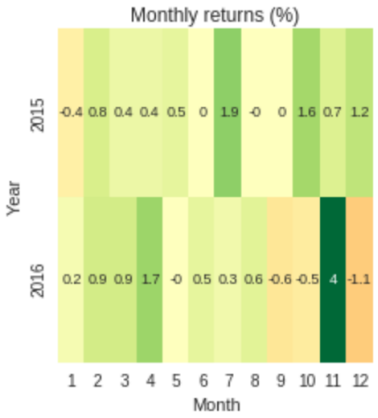
Top 10 positions of all time	max
MRO-5035	2.02%
NEM-5261	2.00%
MU-5121	1.93%
MSFT-5061	1.92%
INTC-3951	1.86%
ADSK-67	1.85%
NTAP-13905	1.84%
CI-1539	1.84%
LUV-4589	1.84%
TXT-7674	1.83%

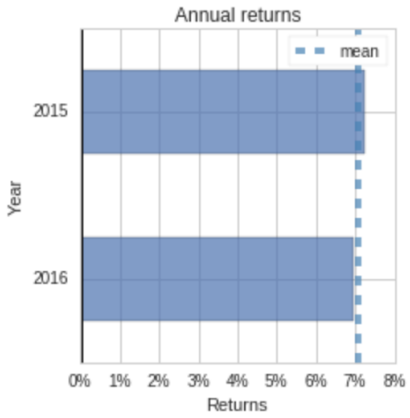


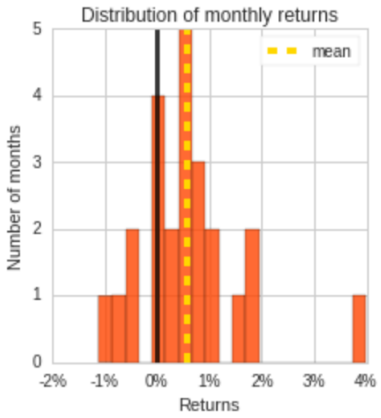

Cumulative returns volatility matched to benchmark
Cumulative returns
2.00
1.80
1.60
1.40
1.20
1.00
0.80
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016

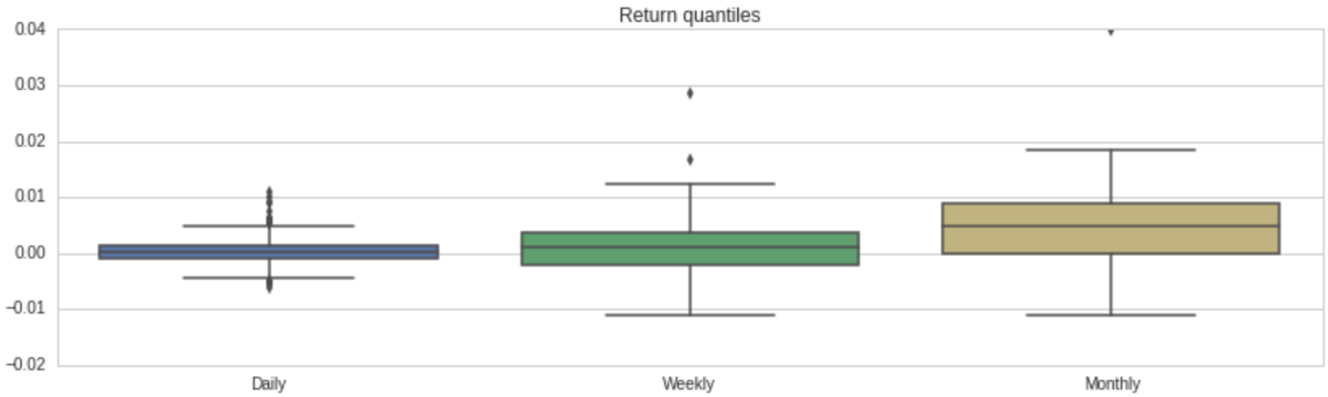

Rolling portfolio beta to Equity(8554 [SPY])
Beta
1.00
0.50
0.00
-0.50
-1.00
6-mo
12-mo
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016

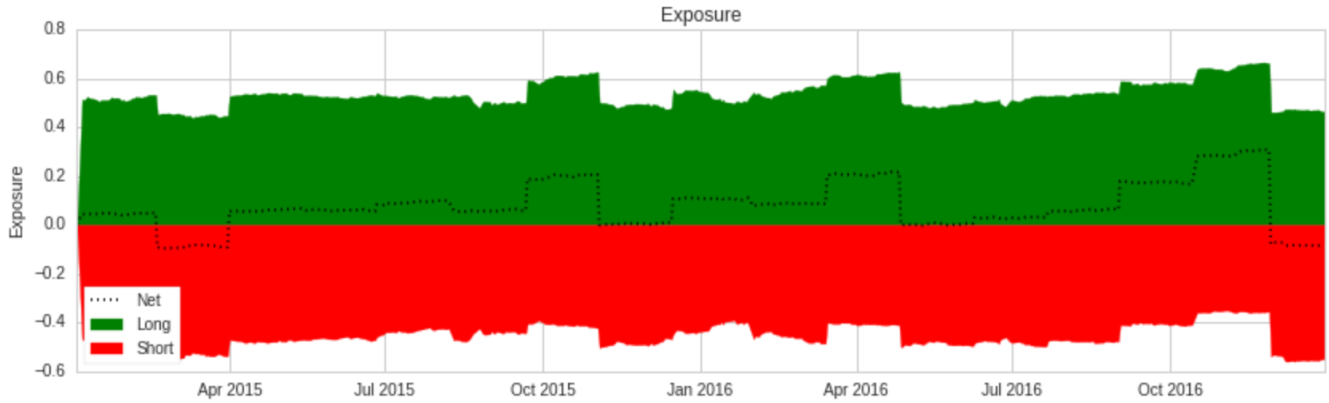

Rolling Fama-French single factor betas (6-month)
Beta
1.00
0.50
0.00
-0.50
-1.00
Small-Caps (SMB)
High-Growth (HML)
Momentum (UMD)
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016


Monthly returns (%)
Year
2015
-0.4 0.8 0.4 0.4 0.5 0 1.9 -0 0 1.6 0.7 1.2
2016
0.2 0.9 0.9 1.7 -0 0.5 0.3 0.6 -0.6 -0.5 4 -1.1
1 2 3 4 5 6 7 8 9 10 11 12
Month


Annual returns
mean
Year
2015
2016
0% 1% 2% 3% 4% 5% 6% 7% 8%
Returns


Distribution of monthly returns
mean
Number of months
0 1 2 3 4 5
-2% -1% 0% 1% 2% 3% 4%
Returns

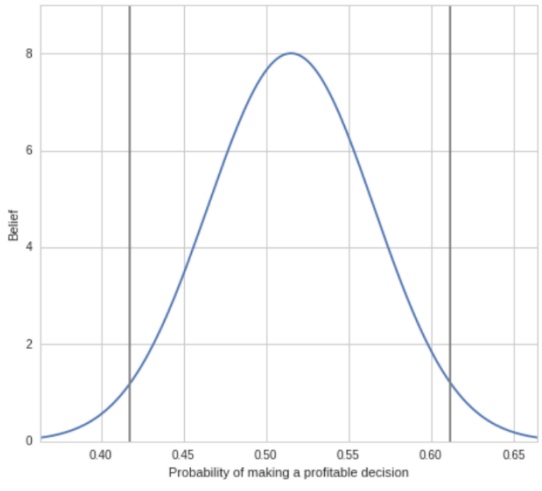

Return quantiles
0.04
0.03
0.02
0.01
0.00
−0.01
−0.02
Daily
Weekly
Monthly

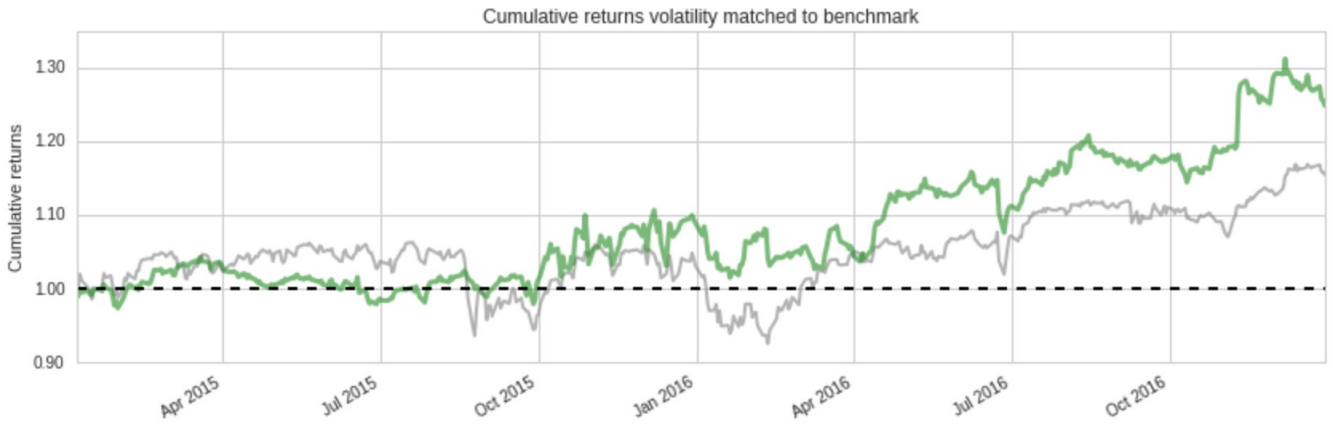

Exposure
Exposure
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6
Net
Long
Short
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016

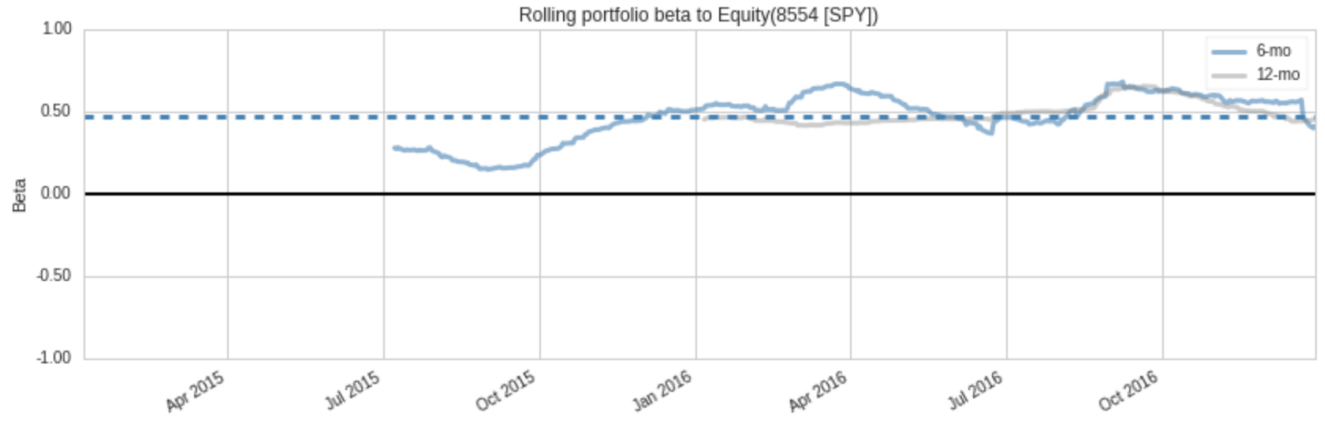
Annex III: Confidence based long/short performance metrics

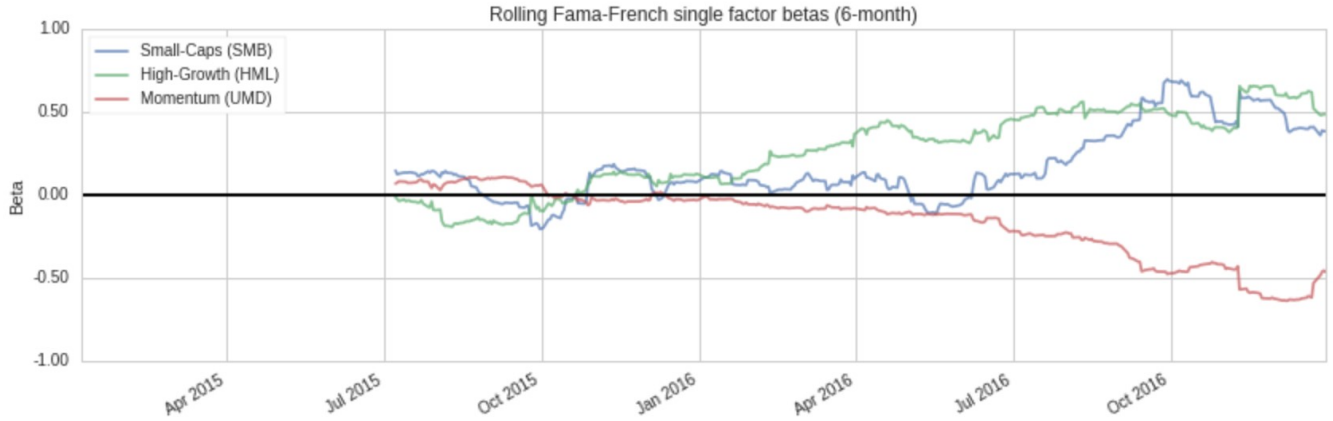
Performance statistics	Backtest
annual_return	0.10
cum_returns_final	0.22
annual_volatility	0.13
sharpe_ratio	0.84
calmar_ratio	1.39
stability_of_timeseries	0.82
max_drawdown	-0.07
omega_ratio	1.17
sortino_ratio	1.24
skew	0.20
kurtosis	5.78
tail_ratio	1.08
common_sense_ratio	1.19
gross_leverage	0.99
information_ratio	0.02
alpha	0.07
beta	0.45

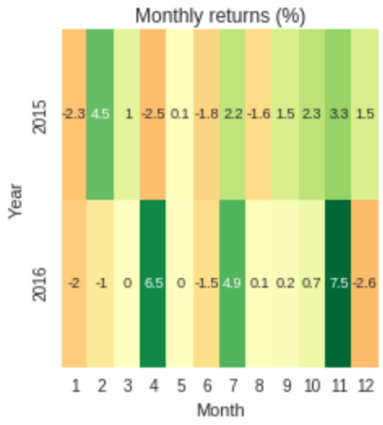
Top 10 positions of all time	max
HON-25090	52.07%
KEY-4221	50.64%
NWL-5520	35.08%
LMT-12691	33.46%
CSCO-1900	30.54%
DOV-2262	26.91%
GIS-3214	26.83%
NOC-5387	26.36%
AIG-239	25.29%
ADP-630	18.51%



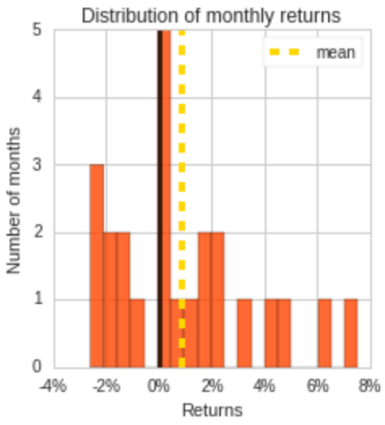

Cumulative returns volatility matched to benchmark
Cumulative returns
1.30
1.20
1.10
1.00
0.90
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016


Rolling portfolio beta to Equity(8554 [SPY])
Beta
1.00
0.50
0.00
-0.50
-1.00
6-mo
12-mo
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016

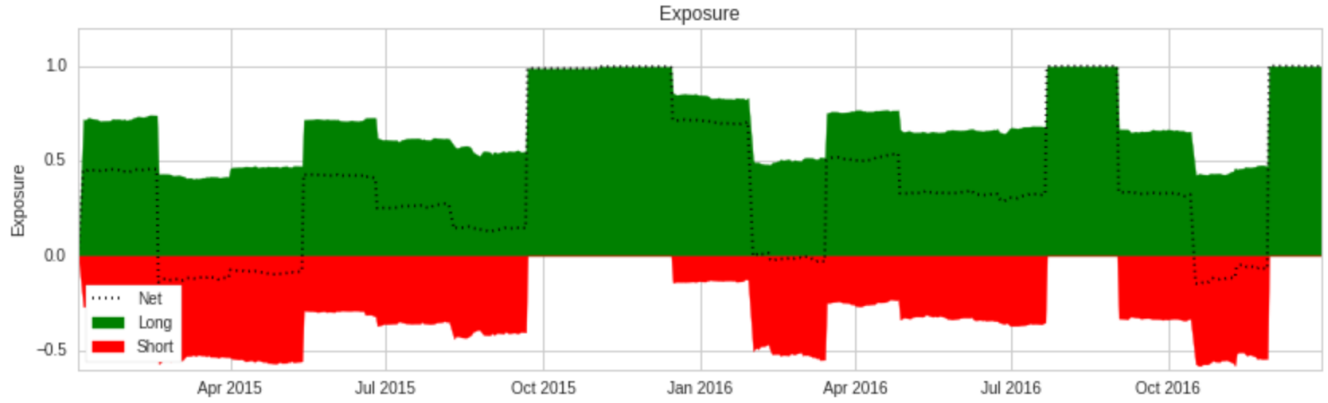

Rolling Fama-French single factor betas (6-month)
Beta
1.00
0.50
0.00
-0.50
-1.00
Small-Caps (SMB)
High-Growth (HML)
Momentum (UMD)
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016


Monthly returns (%)
Year
2015
2016
-2.3 4.5 1 -2.5 0.1 -1.8 2.2 -1.6 1.5 2.3 3.3 1.5
-2 -1 0 6.5 0 -1.5 4.9 0.1 0.2 0.7 7.5 -2.6
1 2 3 4 5 6 7 8 9 10 11 12
Month


Annual returns
mean
Year
2015
2016
0% 2% 4% 6% 8% 10% 12% 14%
Returns

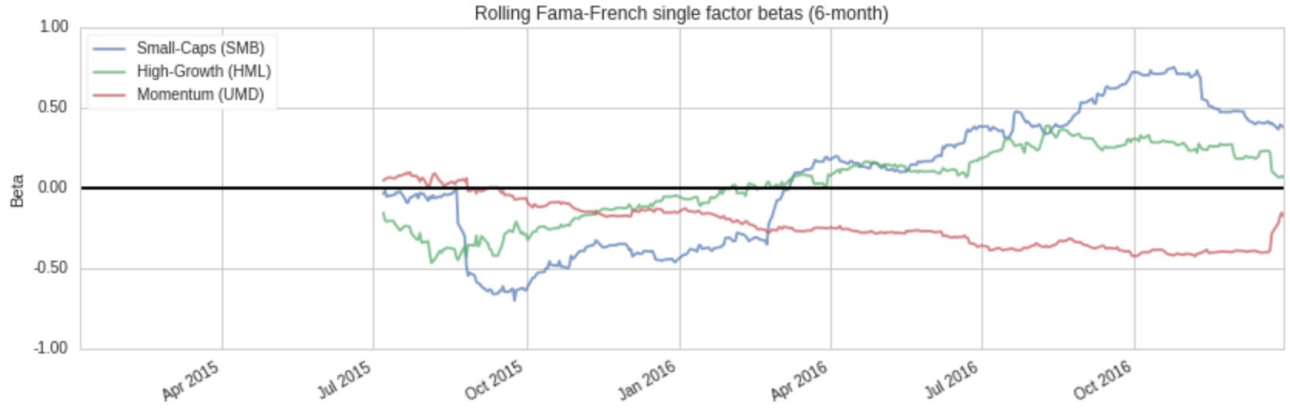

Distribution of monthly returns
mean
Number of months
0 1 2 3 4 5
-4% -2% 0% 2% 4% 6% 8%
Returns

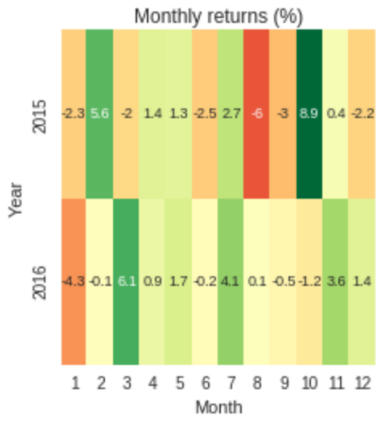

Return quantiles
0.08
0.06
0.04
0.02
0.00
−0.02
−0.04
Daily
Weekly
Monthly

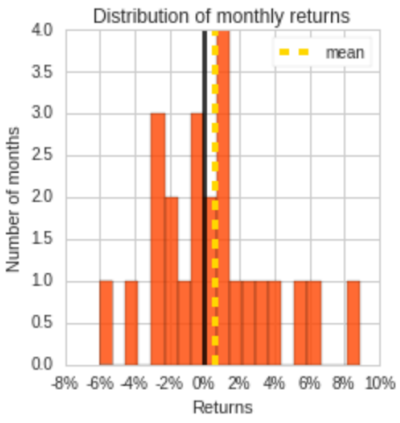

Exposure
Exposure
1.0
0.5
0.0
−0.5
Net
Long
Short
Apr 2015
Jul 2015
Oct 2015
Jan 2016
Apr 2016
Jul 2016
Oct 2016

Annex IV: SPY buy and hold performance metrics for comparison

Performance statistics	Backtest
annual_return	0.06
cum_returns_final	0.13
annual_volatility	0.14
sharpe_ratio	0.51
calmar_ratio	0.50
stability_of_timeseries	0.35
max_drawdown	-0.13
omega_ratio	1.09
sortino_ratio	0.71
skew	-0.28
kurtosis	2.34
tail_ratio	1.05
common_sense_ratio	1.11
gross_leverage	0.98
information_ratio	-0.01
alpha	-0.00
beta	0.99







Bibliography

- Barr, D., Mani, G., "Using Neural Nets to Manage Investments", AI Expert, February, 1994, 16 –21.
- Breiman, L., "Random Forests", Machine Learning, 45, 2001, 5-32.
- Chakraborty, K., Mehrotra, K., Mohan, C. K., Ranka, S., "Forecasting the Behaviour of Multivariate Time Series Using Neural Networks", Neural Networks, 5(2), 1992, 961 – 970
- Grudnitski, G., Osborn, L., "Forecasting S & P and Gold Futures Prices: An Application of Neural Networks", The Journal of Futures Markets, 13 (6), 1993,631 – 643, 1993.
- Hodgson, A., Nicholls, D., "The impact of index futures markets on Australian share market volatility", Journal of Business Finance and Accounting, 8, 1991, 267–80.
- Huerta R., Corbacho F., Elkan C., "Nonlinear Support Vector Machines can systematically identify stocks with high and low future returns", Algorithmic Finance 2, 2013, 45-58.
- Kamruzzaman, J., Sarker, R. A., "ANN-Based Forecasting of Foreign Currency Exchange Rates", Neural Information Processing - Letters and Reviews, 3 (2), 2004.
- Kim, K. J., "Financial time series forecasting using support vector machines", Neurocomputing, 55, 2003, 307 – 319.
- Kimoto, T., Asakawa, K., Takeoka, M., "Stock Market prediction system with modular neural networks", In: proceedings of the IEEE International Joint Conference on Neural Networks. San Diego, California, 2, 1990, 11-16.
- Kumar and Thenmozhi., "Forecasting stock index movement: A comparison of Support Vector Machines and Random Forest", 2006.
- Larivière, B., Poel, D. V. D., "Predicting Customer Retention and Profitability by Using Random Forests and Regression Forests Techniques", Working Paper, Department of Marketing, Hoveniersberg 24, 9000 Gent, Belgium, 2004.
- Maberly, E. D., "The informational content of the interday price change with respect to stock index futures", Journal of Futures Markets, 6, 1986, 385–395.
- C. Man-Chung, W. Chi-Cheong and L. Chi-Chung. "Financial time series forecasting by neural network using conjugate gradient learning algorithm and multiple linear regression weight initialization," Computing in Economics and Finance, vol. 61, 2000

- Refenes, A. N., Zapranis, A. S., Francis, G., "Stock Performance Modeling Using Neural Networks: Comparative Study With Regressive Models", Neural Networks, 7(2), 1994, 375-388.
- Wu, Y., & Zhang, H., "Forward premiums as unbiased predictors of future currency depreciation: A nonparametric analysis", Journal of International Money and Finance 16, 1997, 609–623.
- Yao, J., Chew, L. T., and Poh, H. L., "Neural networks for technical analysis: a study on KLCI", International Journal of Theoretical and Applied Finance, 2(2), 1999, 221-241

Source code

The source code, in addition to the data used, is available on my GitHub:

https://github.com/max-fitzpatrick

Tools used

- Anaconda 4.6
- Python 3.5
    - Zipline
    - Pyfolio
    - Alphalens
    - Pandas
    - Numpy
    - SKlearn
    - Matplotlib
- Visual Studio 2017
- Jupyter iPython notebook
- Microsoft Excel