LUND UNIVERSITY FINANCE SOCIETY EST 1991

TRADING & QUANTITATIVE RESEARCH REPORT

ASSET ALLOCATION

A machine learning strategy

In collaboration with:

OQAM

Analysts:
Niclas Wölner-Hanssen

Introduction

In collaboration with OQAM Asset Management, in this project we compare a machine learning (ML) asset allocation strategy against a traditional asset allocation strategy. The ML-strategy is targeted towards the stock market and is using the Random Forest algorithm as the allocation engine:

- Whenever the Random Forest predicts the stock market to *not* fall more than -2% the next week, we allocate towards the stock market.
- Whenever the Random Forest predicts the stock market to fall more than -2% the next week, we allocate towards the bonds market.

As benchmark to our ML strategy, we use a traditional 60/40 strategy. The 60/40 strategy used here is a weekly rebalanced 60% equities and 40% bonds allocation, a method similar to what is commonly advised to customers at commercial banks.

A word on backtest overfitting

Backtest overfitting is when your in-sample backtest is profiting from noise rather than signal. When your in-sample backtest is profiting from noise and not signal, the strategy you are trading on is constructed to trade on trading-signals that are too in-sample specific. The trading signals resulted in an outcome that by chance happened to be profitable in the in-sample period and is therefore likely to perform poorly out-of-sample (OOS). A classic example is when one changes trading rules in the in-sample period in order to maximize the Sharpe ratio for that time period.

Another kind of overfitting that occurs, but not by looking at a backtest, is when you are specifying your trading-strategy based on your memory of the market. You know today what the market has done in the past and create a strategy according to historical events that is too in-sample specific. We are calling this memory-overfitting.

Separating noise from signal in financial time series is a very difficult task and to avoid all overfitting is probably impossible. However, the opposite of being too specific is to be non-specific. In the strategy outlined in this report we are therefore going to follow the parsimony principle, also known as Occam's razor. To clarify, we rather want to have fewer parameters (non-complexity) explaining some of the variation in-sample than more parameters (complexity) that explains most of the variation in-sample.

Within the objective of constructing an asset allocation strategy based on Random Forest, to be somewhat successful, backtest overfitting prevention is the main goal. Therefore, in the strategy-making process we are to some extent going to follow the guidelines proposed by Professor Marcos Lopez de Prado in his book *Advances in Financial Machine Learning (2018)* which is mostly dedicated to this important topic. To quote Lopez de Prado, the main problem with ML in finance essentially is that:

"ML algorithms will always find a pattern, even if that pattern is a statistical fluke. "

Decision Tree & Random Forest

In machine learning lingo, a supervised model is a model where you specify what it is that you want to predict, the output ("Target"), and with what inputs ("Features"). These components are then inserted into an algorithm (a Decision Tree for example) to learn the mapping function from the input to the output.

Decision Tree

As *Figure 1: Decision Tree* implies, Decision Trees makes predictions by breaking down data into smaller subsets by asking questions about the data.

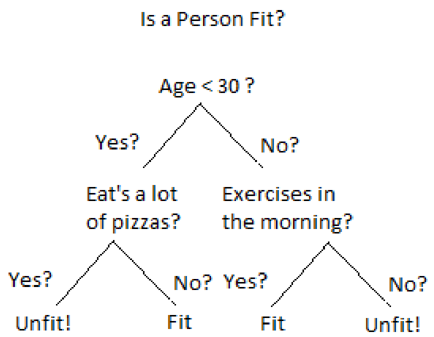


Figure 1: Decision Tree

Figure 1: Decision Tree is a simplified image of a Decision Tree (in appendix 1 is a more detailed version of a Decision Tree), where we try to predict if a person is fit or unfit by a if-else-then rule-system.

- If age < 30 and eats pizza a lot = Unfit.
- If age < 30 and does not eat pizza a lot = Fit.

… and so on.

This is the same procedure that is often times done in systematic trading:

- If stock price is above X & have reported Z in sales last quarter: Trade, because we predict up movement of the price.

- If stock price is above X, but have not reported Z in sales last quarter: Do not Trade, because we predict down movement of the price.

What would have been very useful is to have the iterable power of a machine, and based on this iteration the machine creates trading rules. This could then either objectively confirm already established trading rules, enlighten the investor with new ideas or work as a standalone data-driven strategy. This is exactly what we were aiming to do here. The tricky part is the quality of those strategies in terms of backtest overfitting.

If not regulated, Decision Trees will keep asking questions until perfect fit is reached. Decision Trees are therefore easily overfitted to the period it is trained on. This is where the Random Forest comes in which is considered to be less prone to overfit than the Decision Tree.

Random Forest

The Random Forest Classifier is a ensemble method in which its prediction is based on a multitude of Decision Trees**,** depicted in *Figure 2: Random Forest*. Specifically, the predicted class in a Random Forest is determined by the class with highest mean probability over all the Decision Trees included in the Random Forest[2].

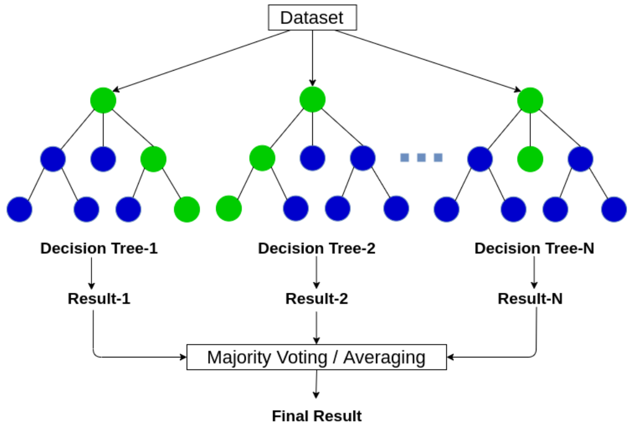


Figure 2: Random Forest

These Decision Trees within the Random Forest have two random components in them which results in a slightly different prediction for each tree. Firstly, the trees make use of bootstrapped samples: some samples are randomly drawn with replacement, which means that one sample can be used multiple times (marked with blue in appendix 1).

2. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Random Forest (continued) & Target

Secondly, when splitting the data, only a random subset of the features is used (marked with green in appendix 1).

In the process of making predictions, each Tree first decides which feature and what feature value should be on the top of the tree, the root node. It makes this decision based on how pure the subsamples are in terms of the target labels. The "purity" or "impurity" is calculated using entropy[3].

The difference between entropy before and after a split is the information gain. The feature with highest information gain is the most important feature for the prediction, it splits the data the best and is going to be the feature on the top of the tree.

In a Random Forest the information gain for each feature is calculated as the average information gain for each feature over all trees. Random Forest can therefore not only be used for prediction but for the purpose of determine which features have most predictive power on the target. This is something we are going to exploit later.

After deciding which feature and on what feature value to split the data on first, another round of information gain calculations over the features are made to further divide the subsamples. Just like in the single Decision Tree case, if not regulated these trees within the Random Forest keeps on splitting the data until perfect purity. However, unlike in the simple Decision Tree case, because of the random components, these trees do not have perfect fit to the real data and is therefore less prone to overfit compared to the simple Decision Tree.

Target

As proxy for the stock market we use SPDR S&P 500 ETF Trust, "SPY" and for the bonds market: Vanguard Total Bond Market Index Fund ETF, "BND".

We use a weekly timeframe. We choose one day of the week to be our rebalance day. On this day we make our prediction 5 days ahead and position ourselves 100% in SPY if we predict that SPY is *not* going to fall more than -2.0%, otherwise we allocate 100% towards BND. We hold our positions until the next week's rebalance day and make new predictions on that day and position ourselves accordingly and so on.

We use weekly returns based on Adjusted Close from finance.yahoo.com from 2008-08-11 to 2021-01-29. Finally, we backtest our strategy for all weekdays and calculate an equal weighted average to be our final result. Due to the stochastic nature of returns, we could be lucky or unfortunate just by choosing Wednesday as rebalance day instead of Tuesday.

In *Table 1: SPY Returns Table* we see for example that the cumulative returns when rebalancing on Mondays is twice that of rebalancing on Fridays. *Table 2: SPY Correlations table*, shows the correlations between the weekly returns for different rebalance days of the week on SPY. To have a more robust backtest we therefore use all weekdays as rebalance days and calculate an average.

Our target is classified as:

- Label 1: when the weekly return of SPY is less than -2.0 %.
- Label 0: when the weekly return of SPY is greater than -2.0%.

SPY	CAGR	Ann. Vol	Cumulative Ret.	MDD
Monday	12.51%	19.95%	302.61%	-46.32%
Tuesday	9.95%	19.99%	206.89%	-43.94%
Wednesday	10.45%	16.61%	223.59%	-37.99%
Thursday	8.93%	17.66%	174.79%	-39.75%
Friday	7.9%	19.14%	145.73%	-49.51%

Table 1: SPY Returns table

SPY - Corr	Monday	Tuesday	Wednesday	Thursday	Friday
Monday	1.0	0.67	0.53	0.38	0.08
Tuesday	0.67	1.0	0.76	0.58	0.32
Wednesday	0.53	0.76	1.0	0.75	0.51
Thursday	0.38	0.58	0.75	1.0	0.69
Friday	0.08	0.32	0.51	0.69	1.0

Table 2: SPY Correlations table

3. https://en.wikipedia.org/wiki/Information_gain_in_decision_trees

Next steps & Drivers

Threshold value of the target classifications

We want to be in the stock market as much as possible, but we want to avoid the bigger drawdowns. The research made during this project also suggests that a threshold of 0% compared to -2%, given the noisy characteristics of many of the features used (which we are discussing later) also increase backtest overfitting. However, more research has to be made on the subject.

Next steps

Having overfitting in mind and the Machine Learning tendency to easily find any pattern, our approach is the following:

1. Select a big universe of features from our trading intuition.
2. Use a Random Forest Nr.1 to select a fraction of those features that have most predictive power on SPY.
3. Take those features and insert them into a Random Forest Nr. 2 to make predictions on SPY.
4. Allocate according to the prediction.

In this way we minimize our memory-overfitting in that we do not select the features that is used, rather we let the Random Forest select the features. We are also having a parsimony principle approach in that we only use a subset of the features we generate. In the same vein, having to many noisy variables in the Random Forest could result in the case that, by chance, some features gets time-specific better feature importance than other more stable features which could result in that the Random Forest neglects the more stable features in favour of the more noisy features.

Drivers

On an aggregate level, when there is a increase in demand relative to supply in financial markets, the price increases. Likewise, when there is an increase in supply relative to demand, the price decreases. On the individual level, generally, when a investor buys an asset, her or she have the belief of a price increase on that asset within some future time period.

On the other side of the trade there is a seller. Generally, that seller have the opposite belief of the buyer of a future price increase or have found a more attractive investment elsewhere to invest in with the cash received from selling the current asset. Therefore, on the aggregate level, one could say that when there are more buyers than sellers, there are more investors believing in a future price increase than there are investors believing in a future price decrease on the underlying asset.

With risk-averse investors, if the underlying asset is relatively risky and buyers are winning, this would indicate a larger risk appetite in the markets. As long as there is risk appetite in the markets, the market will more likely increase in value than the opposite.

In line with our mission to avoid the bigger drawdowns, when collecting drivers for SPY our reasoning is as follows:

- If the market indicates a willingness to take on risk → buyers of SPY will win → SPY will increase.
- If the market is not willing to take on risk → sellers of SPY will win → SPY will decrease.

The choosen drivers for SPY are listed on the next page in *Table 3: Selected Drivers.*

LINC
LUND UNIVERSITY FINANCE SOCIETY EST 1991

Drivers (Continued)

Yahoo Ticker	Single stocks:
AAPL	Apple
AMZN	Amazon
GOOGL	Google
MSFT	Microsoft

Yahoo Ticker	Stock Indicies (ETFs):
SLY	Small Cap
SPY	S&P 500
DIA	DOW

Yahoo Ticker	Sector Indicies (ETFs):
XLY	Consumer Discretionary
XLP	Consumer Staples
XLF	Financial
XLI	Industrial
XME	Materials
XLK	Technology
XLE	Energy

Yahoo Ticker	MISC:
^VIX	VIX
XLY-XLP	Consumer Beta
HG=F	Copper (Future)
SHY	1-3y Bonds (ETF)
GLD	Gold (ETF)
DX-Y.NYB	USD index

Table 3: Selected Drivers

Motivation

Single Stocks – SPY is an ETF which tracks the S&P 500 index. S&P 500 is a market-value-weighted index of 500 large companies in the United States. Single stocks such as Amazon, Google, Apple and Microsoft who all have relatively large weight on the index is therefore included. What happens on these markets should therefore affect SPY on a weekly time frame.

Single Indices – Small cap stocks have historically higher volatility than large cap stocks**. T**heoretically, if buyers win on the riskier small cap market there should be a willingness to also take on risk for the less riskier large cap stocks going forward.

Sector Indices – These 500 companies within SPY are also divided into business segments. The Financial, Technology and Consumer Discretionary sector are the three largest weighted cyclical sectors of SPY. Given their size, price changes on these indices should affect the major index. Their high beta also makes them more riskier in that they tend to be more volatile than other sectors within SPY.

Therefore, if buyers win on these markets, theoretically this would indicate a willingness to take on risk going forward. Constituents in the Industrial & Materials sectors are companies early in the value chain, theoretically if buyers win here the market has faith on the economy going forward.

MISC – Of interest is also the Volatility Index, VIX, which is an index derived from prices on S&P 500 options reflecting the markets expectations on the 30-day forward volatility of S&P 500. Increased expected volatility would theoretically increase uncertainty and therefore decrease the risk appetite.

USD index: foreign investors in SPY with bearish outlook on the USD might have a bearish sentiment towards buying USD denominated stocks because of the currency risk.

Copper: base metal heavily used in electronic devices and in the construction sector. Theoretically, if buyers win here the market has faith on the economy going forward.

SHY: ETF that tracks an index of 1-3 Year US Treasury bonds. While the index tracks the price of the underlying bonds, it also tracks the corresponding interest yields. However the relationship is inverse. The interest yield affects business margins in that their cost of debt increases with the interest paid on their loans. The interest yield could also work as a proxy for the effectiveness of QE (Quantitative Easing) which is a expansionary monetary policy where the FED aim to lower interest rates with longer maturity. Lower interest rates on the other hand could theoretically increase stock buy-backs, since CEO's and CFO's then have a cheap way to increase stock value by decreasing the amount of stocks outstanding.

Features & Feature Importance

Features

We then refine our 20 drivers using Technical-Analysis (TA) transformations. After the TA-transformations our 20 drivers results in a ML-ready feature universe of 154 features. The computations are described in *Table 4: TA-transformations.*

The discussion in the drivers section about increased risk appetite, when buyers are winning on risky assets, is lagged information. We know only after the upward move that there are more believers on future upward movement of the price than downward movement of the price. That sentiment could change the next day or next week. However, based on behavioural theory, that upward movement could lead to further upward movement.

Except for Bollinger Bands, which is a mean reverting strategy, most of the TA-computations are momentum based. With these momentum computations we try to quantify human behavioural tendencies such as "fear of missing out" and "herding". In a financial markets context what this means is that investors tend to gravitate towards assets that have already performed well in the past due to the fear of missing out on further price increases.

Random Forest Nr.1 – Feature Importance

In this section we are aiming to select only the most important features of the 154 features that makes up our feature universe. A couple of feature importance methods that professor Lopez De Prado mentions in his book *Advances in Financial Machine Learning (2018)* are:

Mean Decrease Impurity (MDI) – The MDI is another name for the built-in feature importance calculation mentioned on page 3 in the Random Forest section. However, the MDI is biased towards features with many unique values which means that the feature importance score is automatically going to be higher for features with more values.

Mean Decrease Accuracy (MDA) – MDA uses permutation where it first fits a model with all features, based on some scoring function like accuracy or recall, then permutates each feature, one-by-one and the difference between the validation set score before and after the feature has been permutated makes up the feature importance for that feature. A problem with the MDA method is that if feature X and feature Z are correlated, when permutating feature X, the difference in score before and after permutating feature X will be zero (since feature Z is intact), although feature X could have significant predictive power to the target.

Feature prefix	Description
MOM_5_10_cross	If price 5-day SMA is above 10-day SMA (true=1,false=-1) (On all drivers Ex. VIX)
MOM_10_20_cross	If price 10-day SMA is above 20-day SMA (true=1,false=-1) (On all drivers Ex. VIX)
MOM_10_cross	If price is above 10-day SMA (true=1,false=-1) (On all drivers Ex. VIX)
MOM 10/21 %	Percentage change between 10-day SMA and 20-day SMA on price (On all drivers Ex. VIX)
BB_signal - bollinger band	If price above 2 std from 20-day SMA = -1, if price below 2 std from 20-day SMA = 1, else 0. (On all drivers Ex. VIX)
frac MOM 10/21 %	Percentage change between 10-day SMA and 20-day SMA on fractional differenced prices.(On all drivers Ex. VIX)
frac MOM 5/21 signal	Percentage change of the 5-day SMA and 21-day SMA on fractional differenced series.(On all drivers Ex. VIX)
inst_B 1,2,3,4w	The number of weeks out that both the volume and price has increased since last week on the single stocks. -1 if False, 1 if True)
VIX:	VIX < 20 = 0, 20 < VIX < 30 = -1, 30 < VIX < 40 = -2, VIX > 40 = -3

Table 4: TA-transformations

Prediction

Single Feature Importance (SFI) – The SFI computes the feature importance by running a Random Forest with only one feature at a time and then look at the mean of the validations set scores. The main problem with SFI is that feature X might have predictive value only in combination with feature Z, which SFI does not account for with its isolation.

Our method – We use a combination of the above methods, we rank the features on all measures for each weekday and then sum those rankings and then select the 20 best performing features. These 20 features are then used as inputs for the prediction model each weekday.

Random Forest Nr.2 - Prediction

We use Random Forest a second time, with the purpose of predicting SPY with the features selected from the first Random Forest model.

Feature transformations

As mentioned, the built-in MDI feature importance calculation within the Random forest is biased towards features with many values. Since our features have mixed number of unique values, before inserting the features into the second Random Forest to predict SPY, we use a one-hot-encoding transformation that creates one feature column for each value on that feature and label it to either 0 or 1. For example: our VIX feature column have the values [1,-1,-2,-3] this would result in 4 columns, the -3 value get a 1 if true, 0 if false. With our one-hot-encoding all features have the same number of unique values and we overcome this problem with different number of feature values.

Cross-validation training and Hypertuning

We train on the period 2008-08 to 2018-06 using a 10-fold walk-forward cross-validation that takes overlapping samples into account. We therefore remove the first 5 samples in each validation set, since we have features that are calculated on information in the training set and also removing the last sample in each training set, since the target label is based on information in the validation set.

The method is visualized in *Figure 3: Cross-validation* where we can see that there is gaps between the training sets and validation sets.

At the end of the training period, the model has created 1000 trading rules, one for every Decision Tree. The predicted class from the Random Forest is again the class with the highest mean probability of all **t**rees. The structure of these individual **t**rees, the depth of the **t**ree, how many features to consider for each split, is determined by which settings that on the aggregate tree level have the highest mean recall score (percentage correctly predicted 1:s) on the 10 validation sets.

Black-box

The reason we use Random Forest instead of a simple Decision Tree is to prevent overfitting. However, it comes with the cost of not having a intuitive formula of the trading strategy. We have a model with 1000 trading rules and the Random Forest is making its prediction based on a average of all of these. Although it is possible to extract the 1000 rules, it could be cumbersome to present on a investor relations meeting.

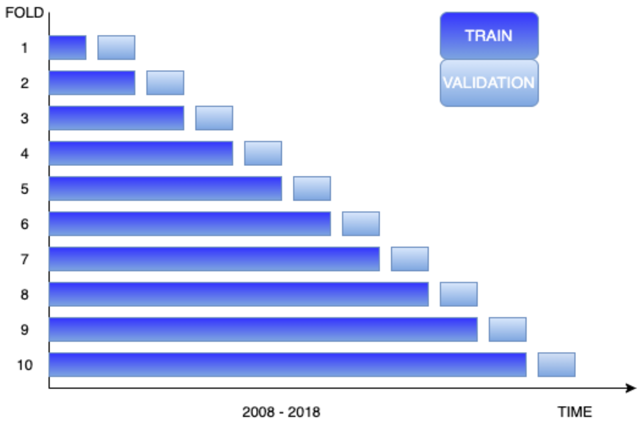


Figure 3: Cross-validation

Backtest

Algo RF: Feature selection and prediction using Random Forest. Whenever the model predicts SPY to have a return that is worse than -2% next week it allocates 100% to BND (bonds), otherwise it allocates 100% to SPY (stocks). *Note:* transaction costs is not taken into account.

60/40: Weekly rebalanced 60% SPY and 40% BND allocation.

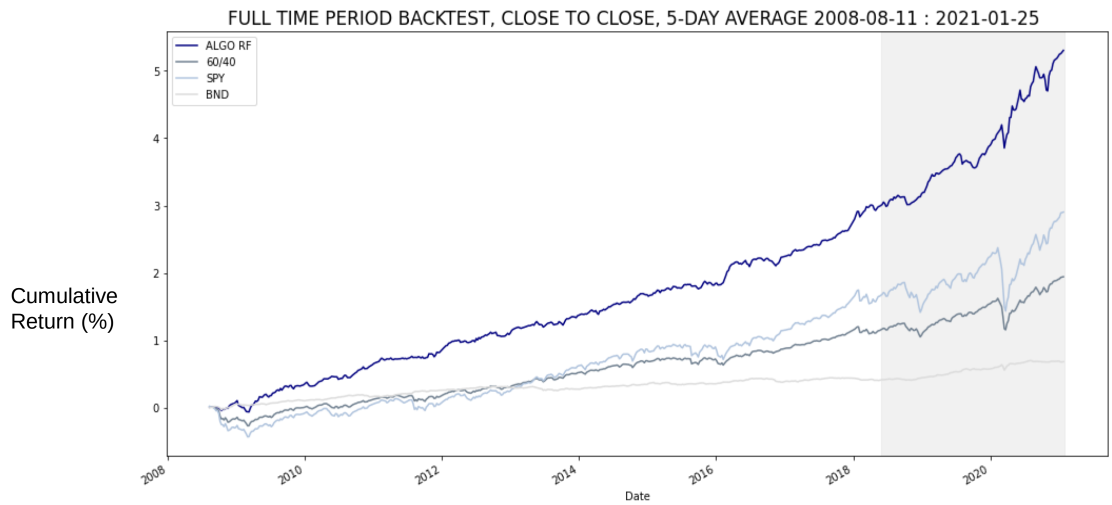


Figure 4: Full-period backtest

5-DAY AVERAGE FULL PERIOD

	CAGR	Ann.Vol	Cumulative Ret.	MDD	Sharpe Ratio	Sortino Ratio	Calmar Ratio
ALGO RF	15.46%	11.34%	530.27%	-17.74%	1.34	2.23	1.00
60/40	8.98%	11.30%	194.48%	-28.93%	0.82	1.14	0.31
SPY	11.46%	18.46%	290.43%	-45.71%	0.68	0.95	0.25
BND	4.24%	4.32%	68.47%	-6.84%	1.00	1.45	0.65

Table 5: Full-period backtest

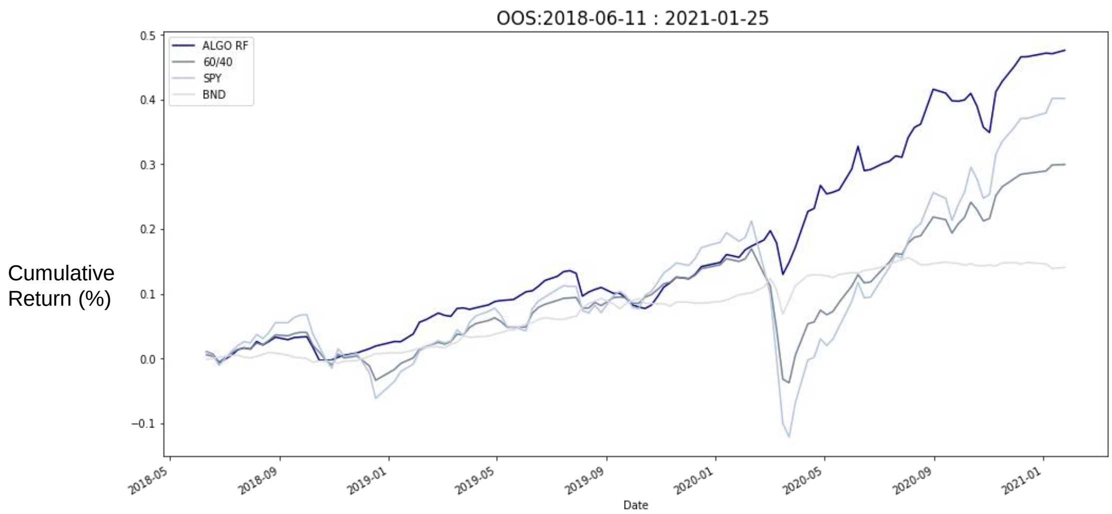


Figure 5: Out-of-sample backtest

5-DAY AVERAGE OUT-OF-SAMPLE

	CAGR	Ann. Vol	Cumulative Ret.	MDD	Sharpe Ratio	Sortino Ratio	Calmar Ratio
ALGO RF	19.04%	11.69%	59.48%	-8.63%	1.56	2.62	2.43
60/40	12.30%	13.30%	36.19%	-20.05%	0.95	1.32	0.62
SPY	15.31%	20.77%	46.14%	-29.93%	0.79	1.1	0.51
BND	6.64%	5.32%	18.68%	-6.17%	1.28	1.85	1.14

Table 6: Out-of-sample backtest

Results

Backtests are what they are, historical simulations of the past. We can visually see in *Figure 4: Full-period backtest* & *Figure 5: Out-of-sample backtest* on the previous page that the ML-strategy is outperforming the 60/40 for this specific period. The backtest tells us that our strategy is not failing us so far, but there is no guarantee that it is not going to fail us in the future.

Table 5: Full-period backtest on the previous page shows the performance metrics for the 5-day average of all weekdays for the full period. In *Table 6: out-of-sample backtest* the 5-day average OOS *Cumulative Return* was 59.5% for ALGO RF vs. 36.2% for 60/40. The *Compound Annual Growth Rate (CAGR),* was 19.04% for ALGO RF vs. 12.30% for 60/40. *Annual Volatility* was 11.69% for ALGO RF vs. 13.3% for 60/40. The *Maximum Drawdown (MDD)* measures downside risk and is the largest observed percentage drop from a peak (top) to trough (bottom) until a new peak is found during the period.

It was -8.63% for ALGO RF vs -20.05 % for 60/40. The *Annualized Sharpe Ratio*, risk-adjusted return, was 1.56 for ALGO RF vs. 0.95 for 60/40. Unlike Sharpe Ratio, which is based on the standard deviation of both positive and negative returns, the *Annualized Sortino Ratio* only considers the standard deviation of the negative returns. Sortino Ratio was 2.62 for ALGO RF vs. 1.32 for 60/40. The *Calmar Ratio* is just like the Sharpe- and Sortino Ratio a risk-adjusted measurement, however instead of dividing the annual return on statistical averages it uses the MDD in the denominator. The Calmar Ratio for ALGO RF was 2.43 vs. 0.62 for 60/40. *Note*: The Risk-free rate is assumed to be 0 and all metrics, except Cumulative Returns and MDD, are annualized compounded with 50 weeks.

Predictions

Table 7: 5-day average predictions shows the 5-day equal weighted %-correct predictions. To gauge how in-sample specific the strategy is, we can look at the difference between correctly predicted target labels on the period the strategy has been trained on and the OOS period.

If we have a lot higher prediction rate on the training period, we have a strategy that is too specific to the training period.

What we want to see is an equal correct percentage prediction on the two periods. The average predicted 1's was 62% for the FULL period and 61% for the OOS. For the 0's it was 62% on FULL period and 59% on the OOS. This indicates that we have succeeded with making the Random Forest selecting a strategy from the training period that is not too in-sample specific, since similar correctly predicted target labels could be found on the period it has not seen. These percentages are however the 5-day average.

In *Table 8: label 1 dispersions* on the next page we can see the dispersion of the %-correct predictions of 1's each weekday. For the 1's, it varied from 41% to 78% on the OOS. The difference between percentage correct labels between the FULL period and OOS is, except for Thursdays, not that way off. *Table 9: label 0 dispersions* shows the predictions of the 0's, which varied from 51% to 62% on the OOS and has also not severe dispersion between the FULL time period and OOS.

Although the OOS period extends over a 2.5 year period, looking at *Table 7: 5-day average predictions* again below we can see that the average number of 1's on the OOS period is 18. This is a very small number when considering percentages: one week more or less is 1/18 = ± 5.5%.

Profit and Loss is presented in Appendix 2

5-DAY AVERAGE PREDICTIONS

	OOS	FULL
Total nr of Weeks:	133	629
Total nr of actual 1s:	18	82
Total nr of predicted 1s:	59	260
Actual 1s that we predicted to be 1s:	11	51
% correctly predicted 1s:	**61%**	**62%**
Total nr of actual 0s:	115	546
Total nr of predicted 0s:	75	369
Actual 0s that we predicted to be 0s:	68	337
% correctly predicted 0s:	**59%**	**62%**
Total Accuracy:	59%	62%

Table 7: 5-day average predictions

Conclusion

We have a strategy showing strong risk-adjusted metrics, in-sample as well as out-of-sample. The average difference between correctly predicted labels between in-sample and out-of-sample indicates that we have a strategy that is not fitted to noise. However, although we so far have decent accuracy out-of-sample on the 1's, due to the small number of 1's, we are not super confident in the strategy's ability to detect the larger drawdowns.

Further more, this project was all about testing out and applying machine learning on finance. A lot of testing was done in terms of trying out code-snippets, trying different scoring functions, different machine learning models and so on.

In addition, threshold values of 0% and -1% on the weekly returns that divide the target classifications were tested on Mondays and Tuesdays to confirm our theory that using these thresholds, together with our rather noisy features, will make the model more likely to create overfitted strategies than if we use a threshold of -2%. Since the period 2016-2018 had not a reflective historical distribution of target labels when threshold value was -2%, comparison were made on out-of-sample territory between 2018-06-01 to 2020-11-30. These test confirmed the theory in that using thresholds of 0% and -1% resulted in the model making strategies that performed extremely good on the trained data but performed poorly on the unseen data.

Due to all of these testing's, the trying-outs and the threshold value investigation, we have increased the strategy-risk going forward**.**

What would I have changed if I was to do it all over?
The first thing I would have changed if I was to do it all over again and used the same target label classifications would be to increase the out-of-sample period. There is simply to few number of weeks when SPY moved lower than -2% on the current 2.5 years that make up the out-of-sample that was used. I know now that the period 2016-2018 had a low number of weeks when SPY moved worse than -2%, therefore I would've started the training period later and save the financial crisis period in 2007-2009 as a second out-of-sample period. Given that the period have the same target label distribution as the average of the full period, 85% of 0's and 15% of 1's, on a 100 weeks period (2 years) that would increase the number of 1's by 15. With a total number of 15+18=33 weeks we would at least have reached the Central Limit Theory number sample size of 30, which would still be small in my opinion but better than the current average number of 18.

In terms of features, we used single stocks based on their weights on SPY today. This weighting might not be accurate in the future. Therefore, a better approach would be to construct dynamic features for these which aims to select the highest by market cap stocks over time. In addition, the majority of drivers used is also based on publicly available price timeseries, which makes them noisy. Many alternatives are on the table that are more robust in the alternative data category, but also in the TA-computations category.

%-correctly predicted 1s SPY < -2%	MON	TUE	WED	THUR	FRI	AVG
FULL	55.29%	58.00%	64.00%	56.00%	78.00%	62.26%
OOS	55.00%	58.00%	78.00%	41.00%	73.00%	61.00%
DIFF (OOS-FULL)	**-0.29%**	**0.00%**	**14.00%**	**-15.00%**	**-5.00%**	**-1.26%**

Table 8: label 1 dispersions

%-correctly predicted 0s SPY > -2%	MON	TUE	WED	THUR	FRI	AVG
FULL	67.00%	62.00%	64.00%	66.00%	50.00%	61.80%
OOS	59.00%	62.00%	61.00%	60.00%	51.00%	58.60%
DIFF (OOS-FULL)	**-8.00%**	**0.00%**	**-3.00%**	**-6.00%**	**1.00%**	**-3.20%**

Table 9: label 0 dispersions

Appendix 1 – Decision Tree (detailed)

In this fictive example, “BB” represents the Bollinger Band feature on SPY itself. It has three values:

- -1 represent the occurrence when the price is above 2 standard deviation from the 20-day moving average and hence when SPY is expected to revert to the SMA20 and move downwards next week.

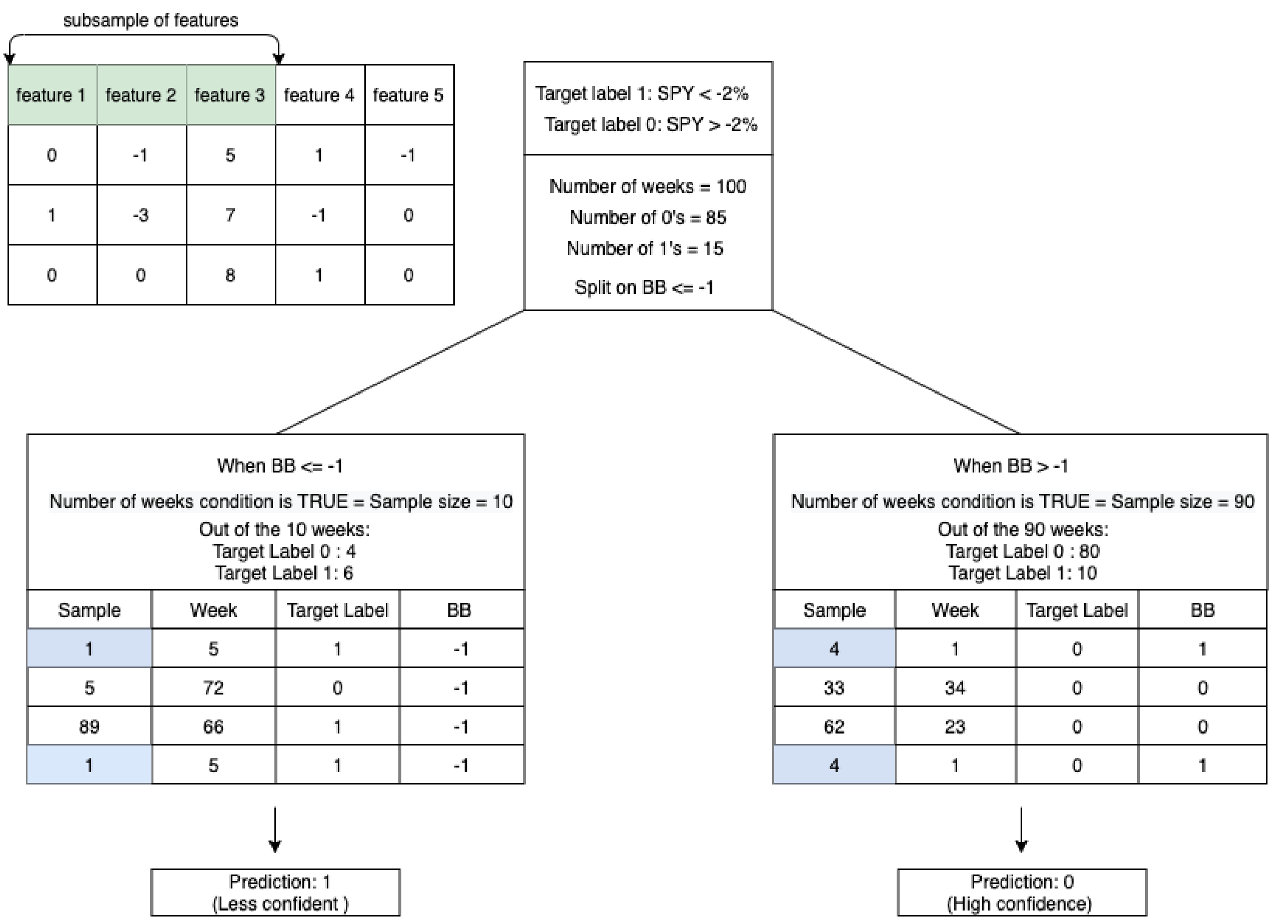
- +1 represent the occurrence when the price is below 2 standard deviations from the 20-day moving average and hence when SPY is expected to revert to the SMA20 and move upwards next week.

- 0 represents all occurrences when -1 or +1 is false. This is the majority category.

Recall that:

- Target label 0: All weeks SPY had returns > -2%.
- Target label 1: All weeks SPY had returns < -2%.

We split on BB <= -1 and move to the left where we have a subsample of all weeks when BB = -1 with corresponding target label output. We move to the right for the weeks BB was not -1, which generated a subsample for the weeks when BB was either 0 or 1 with corresponding target label outputs for those weeks. In comparison to *Figure 1: Decision Tree*, we are not 100% sure if the prediction is 1 or 0 when BB = -1. Out of the 10 samples when BB = -1 there are four samples with target label 0 and six samples with target label 1. The probability is 60% that the target is a “1” and the probability is 40% that it is a “0”, the target label prediction will therefore be a “1” for BB = -1.



Appendix 2 – Profit & Loss

PROFIT AND LOSS
(Full Period)

	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	AVERAGE
Winning Weeks	62%	61%	65%	63%	65%	**63%**
Avg. Return Win Weeks	1.14%	1.03%	1.04%	1.05%	0.98%	**1.05%**
Max Return Win Weeks	12.09%	11.38%	8.33%	10.04%	17.92%	**11.95%**
Min Return Win Weeks	0.01%	0,00%	0.01%	0.01%	0.01%	**0.01%**
Losing Weeks	38%	38%	35%	37%	34%	**36%**
Avg. Return Lose Weeks	-1.13%	-1.04%	-0.99%	-0.98%	-0.83%	**-0.99%**
Max Return Lose Weeks	-0.001%	-0.004%	-0.005%	-0.004%	-0.004%	**-0.004%**
Min Return Lose Weeks	-10.99%	-9.8%	-7.51%	-6.95%	-8.05%	**-8.66%**
Weeks in SPY	64%	59%	61%	63%	47%	**59%**
Avg. Return in SPY	0.4%	0.34%	0.46%	0.43%	0.66%	**0.46%**
Max Return in SPY	12.09%	11.38%	8.33%	10.04%	17.92%	**11.95%**
Min Return in SPY	-10.99%	-9.8%	-7.51%	-6.95%	-7.29%	**-8.51%**
Winning Weeks in SPY	62%	60%	66%	65%	69%	**64%**
Avg. Return Win Weeks in SPY	1.52%	1.49%	1.41%	1.39%	1.52%	**1.47%**
Max Return Win Weeks in SPY	12.09%	11.38%	8.33%	10.04%	17.92%	**11.95%**
Min Return Win Weeks in SPY	0.005%	0.005%	0.008%	0.016%	0.011%	**0.009%**
Losing Weeks in SPY	38%	40%	34%	35%	31%	**36%**
Avg. Return Lose Weeks in SPY	-1.4%	-1.41%	-1.39%	-1.33%	-1.24%	**-1.36%**
Max Return Lose Weeks in SPY	-0.012%	-0.004%	-0.008%	-0.004%	-0.004%	**-0.007%**
Min Return Lose Weeks in SPY	-10.99%	-9.8%	-7.51%	-6.95%	-7.29%	**-8.51%**
Weeks in BND	36%	41%	39%	37%	53%	**41%**
Avg. Return in BND	0.08%	0.08%	0.12%	0.06%	0.09%	**0.09%**
Max Return in BND	5.4%	2.64%	2.27%	1.75%	5.94%	**3.6%**
Min Return in BND	-7.94%	-3.61%	-4.32%	-4.83%	-8.05%	**-5.75%**
Winning Weeks in BND	64%	62%	63%	59%	62%	**62%**
Avg. Return Win Weeks in BND	0.49%	0.39%	0.43%	0.4%	0.46%	**0.44%**
Max Return Win Weeks in BND	5.4%	2.64%	2.27%	1.75%	5.94%	**3.6%**
Min Return Win Weeks in BND	0.012%	0.003%	0.011%	0.013%	0.012%	**0.01%**
Losing Weeks in BND	36%	36%	37%	40%	37%	**37%**
Avg. Return Lose Weeks in BND	-0.64%	-0.45%	-0.43%	-0.45%	-0.52%	**-0.5%**
Max Return Lose Weeks in BND	-0.001%	-0.012%	-0.005%	-0.012%	-0.012%	**-0.009%**
Min Return Lose Weeks in BND	-7.94%	-3.61%	-4.32%	-4.83%	-8.05%	**-5.75%**

LUND UNIVERSITY FINANCE SOCIETY EST 1991

Disclaimer

Disclaimer

These analyses, documents and any other information originating from LINC Research & Analysis (Henceforth "LINC R&A") are created for information purposes only, for general dissipation and are not intended to be advisory. The information in the analysis is based on sources, data and persons which LINC R&A believes to be reliable. LINC R&A can never guarantee the accuracy of the information. The forward-looking information found in this analysis are based on assumptions about the future, and are therefore uncertain by nature and using information found in the analysis should therefore be done with care. Furthermore, LINC R&A can never guarantee that the projections and forward-looking statements will be fulfilled to any extent. This means that any investment decisions based on information from LINC R&A, any employee or person related to LINC R&A are to be regarded to be made independently by the investor. These analyses, documents and any other information derived from LINC R&A is intended to be one of several tools involved in investment decisions regarding all forms of investments regardless of the type of investment involved. Investors are urged to supplement with additional relevant data and information, as well as consulting a financial adviser prior to any investment decision. LINC R&A disclaims all liability for any loss or damage of any kind that may be based on the use of analyzes, documents and any other information derived from LINC R&A.

Conflicts of interest and impartiality

To ensure LINC R&A's independence, LINC R&A has established compliance rules for analysts. In addition, all analysts have signed an agreement in which they are required to report any and all conflicts of interest. These terms have been designed to ensure that *COMMISSION DELEGATED REGULATION (EU) 2016/958 of 9 March 2016, supplementing Regulation (EU) No 596/2014 of the European Parliament and of the Council with regard to regulatory technical standards for the technical arrangements for objective presentation of investment recommendations or other information recommending or suggesting an investment strategy and for disclosure of particular interests or indications of conflicts of interest.*

Other