Stock market analysis and prediction

Table of content

1. Introduction

The goals of this project are to:

- Get stock data online for a given time period
- Display in an interactive graph various technical indicators for a given stock or a stock portfolio
- Create a machine learning algorithm to predict if a given stock is a "buy" or a "sell" on a given day
- Optimize various technical indicators to find the ones that provide the best growth over a period
- Try to make money!

2. Data preparation and functions

Libraries import

In this section we import all the necessary libraries to run our project.

P.S. If code is hidden, the reader won't see anything in this section

Function to show the direction of a given stock using various technical indicators

With this function, the objective is to go get stock info online for given timeline, derive the several technical indicators for the stock and determine if the stock is currently a buy or a sell.

The inputs are:

- the stock ticker,
- the period start and end time,
- the Exponential Moving Average (EMA) short and long periods (https://www.investopedia.com/terms/e/ema.asp (https://www.investopedia.com/terms/e/ema.asp)),
- the Commodity Channel Index (CCI) period (usually 21 days) and typical price divider (usually 3 days) (https://www.investopedia.com/terms/c/commoditychannelindex.asp (https://www.investopedia.com/terms/c/commoditychannelindex.asp)),
- the Slow Stochastic short (usually 3 days) and long periods (usually 17 days) (https://www.investopedia.com/terms/s/stochasticoscillator.asp (https://www.investopedia.com/terms/s/stochasticoscillator.asp)),
- the Moving Average Convergence Divergence (MACD) short period (usually 7 days), long period (usually 27 days) and signal (usually 7 days) (https://www.investopedia.com/terms/m/macd.asp (https://www.investopedia.com/terms/m/macd.asp))

The outputs are:

- the Exponential Moving Average (EMA) recommendation and number of days since trend beginning,
- the Directionnal Movement Index (DMI) recommendation (https://www.investopedia.com/terms/d/dmi.asp (https://www.investopedia.com/terms/d/dmi.asp)),
- the Moving Average Convergence Divergence (MACD) recommendation and number of days since trend beginning,
- the Relative Strength Index (RSI) recommendation and number of days since trend beginning https://www.investopedia.com/terms/r/rsi.asp (https://www.investopedia.com/terms/r/rsi.asp),
- the Commodity Channel Index (CCI) recommendation and number of days since trend beginning,
- Slow Stochastic recommendation and number of days since trend beginning,
- the data in a Panda table.

Function to tap stock data online

For this function, the objective is to go get stock info online for a given timeline.

The inputs are:

- the stock ticker,
- the start time in unix format,
- the end time in unix format,

The outputs are:

- a Pandas dataframe containing for each day: Open, High, Low, Close, Adjusted Close and Volume

Function to chart multiple technical indicators for a given stock over a given timeline

With this function, the objective is to go get stock info online for given timeline, derive the several technical indicators for the stock and plot them in a compelling chart format.

The inputs are:

- the stock ticker,
- the start time in unix format,
- the end time in unix format,

The output is a chart showing the following technical indicator over a given period:

- the Exponential Moving Average (EMA) short and long period lines (https://www.investopedia.com/terms/e/ema.asp (https://www.investopedia.com/terms/e/ema.asp)) superposed on the daily OHCL chart,
- the Directional Moving Index (DMI) +DI14, -DI14 and ADX14 lines (https://www.investopedia.com/terms/d/dmi.asp (https://www.investopedia.com/terms/d/dmi.asp)),
- the Relative Strength Index (RSI) (https://www.investopedia.com/terms/r/rsi.asp (https://www.investopedia.com/terms/r/rsi.asp)) normalized to be superposed with the stock highs and lows,
- the Relative Strength Index (RSI) line (https://www.investopedia.com/terms/r/rsi.asp (https://www.investopedia.com/terms/r/rsi.asp)),
- the Moving Average Convergence Divergence (MACD) line and signal line (https://www.investopedia.com/terms/m/macd.asp (https://www.investopedia.com/terms/m/macd.asp)),
- the Commodity Channel Index line (https://www.investopedia.com/terms/c/commoditychannelindex.asp (https://www.investopedia.com/terms/c/commoditychannelindex.asp)),
- the Slow Stochastic 3 days line and the 14 days lines (https://www.investopedia.com/terms/s/stochasticoscillator.asp (https://www.investopedia.com/terms/s/stochasticoscillator.asp))
- a textual output of EMA and DMI trends.

Note: the technical indicator are shown with their "normal" periods.

Function that computes and shows a confusion matrix

The objective of this function is plot a confusion matrix

3. Stock Charts

In this section, we plot the charts of a desired stock and a portfolio based on a desired timeframe.

Timeframe input

The user inputs the timeframe from which he wants to see visual charts.

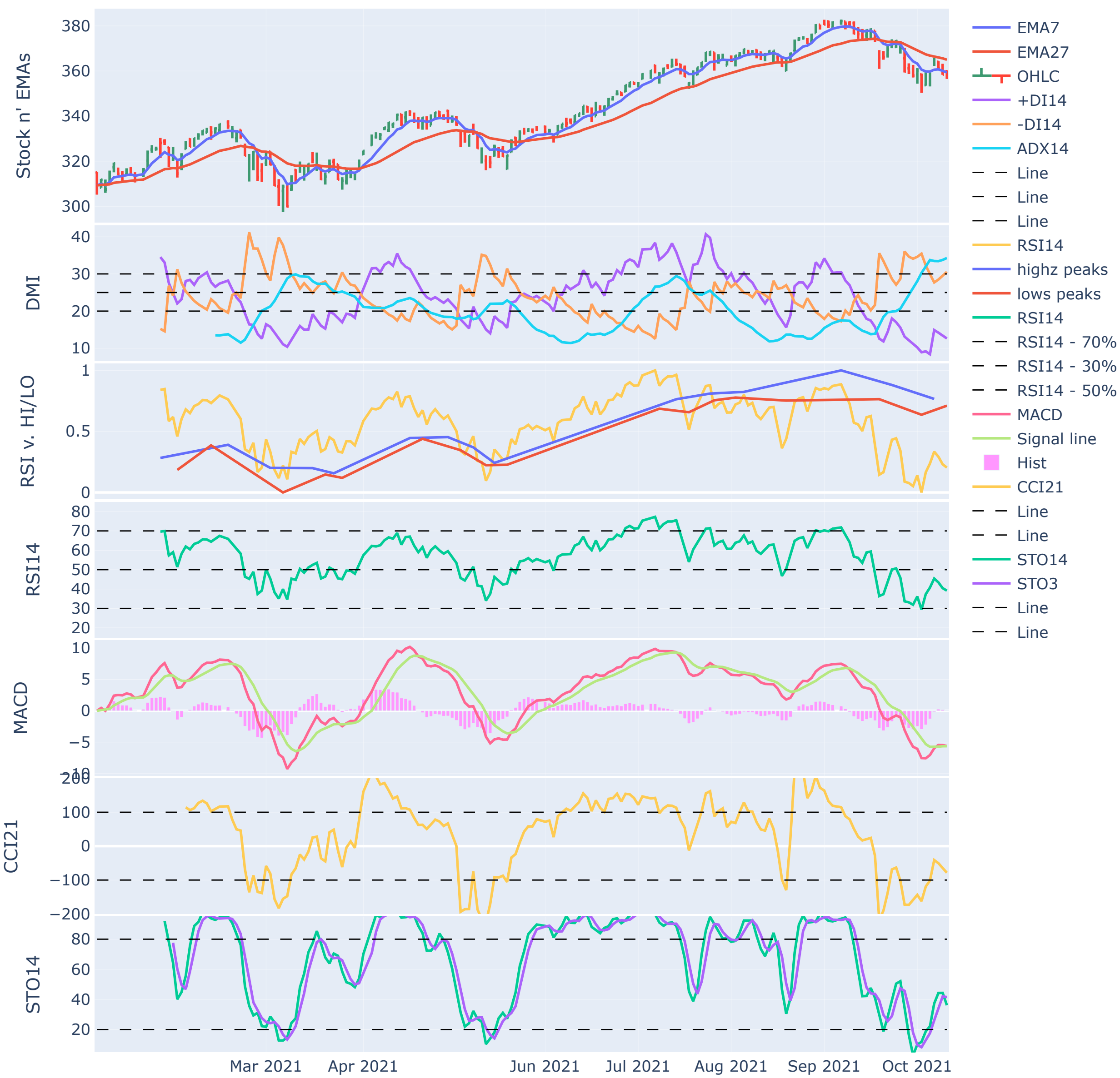
```
please enter start date and end date. Use this format: 2020-09-24
Enter a start date in YYYY-MM-DD format: 2021-01-01
Enter a end date in YYYY-MM-DD format: 2021-10-13
Enter a short period term for the EMA (in days, normal 7): 7
Enter a long period term for the EM (in days, normal 27): 27
Enter a CCI period (in days, normal 21): 21
Enter a CCI time divider (in days, normal 3): 3
Enter a Slow Stochastic long period (in days, normal 17): 17
Enter a Slow Stochastic short period (in days, normal 3): 3
Enter a MACD short period (in days, normal 7): 7
Enter a MACD short period (in days, normal 27): 27
Enter a MACD signal period (in days, normal 7): 7

Start date: 2021-01-01   End date: 2021-10-13
Start date in unix: 1609477200   End date in unix: 1634097600
EMA Short period: 7 days   EMA Long period: 27 days

CCI period: 21
CCI typ divider: 3

Slow Sto long period: 17
Slow Sto short period: 3

MACD short period: 7
MACD long period: 27
MACD signal period: 7
```

Individual stock input

The user inputs the stock ticker ot the company he wants to see the visual chart

```
Enter stock ticker of choice: qqq
```

qqq

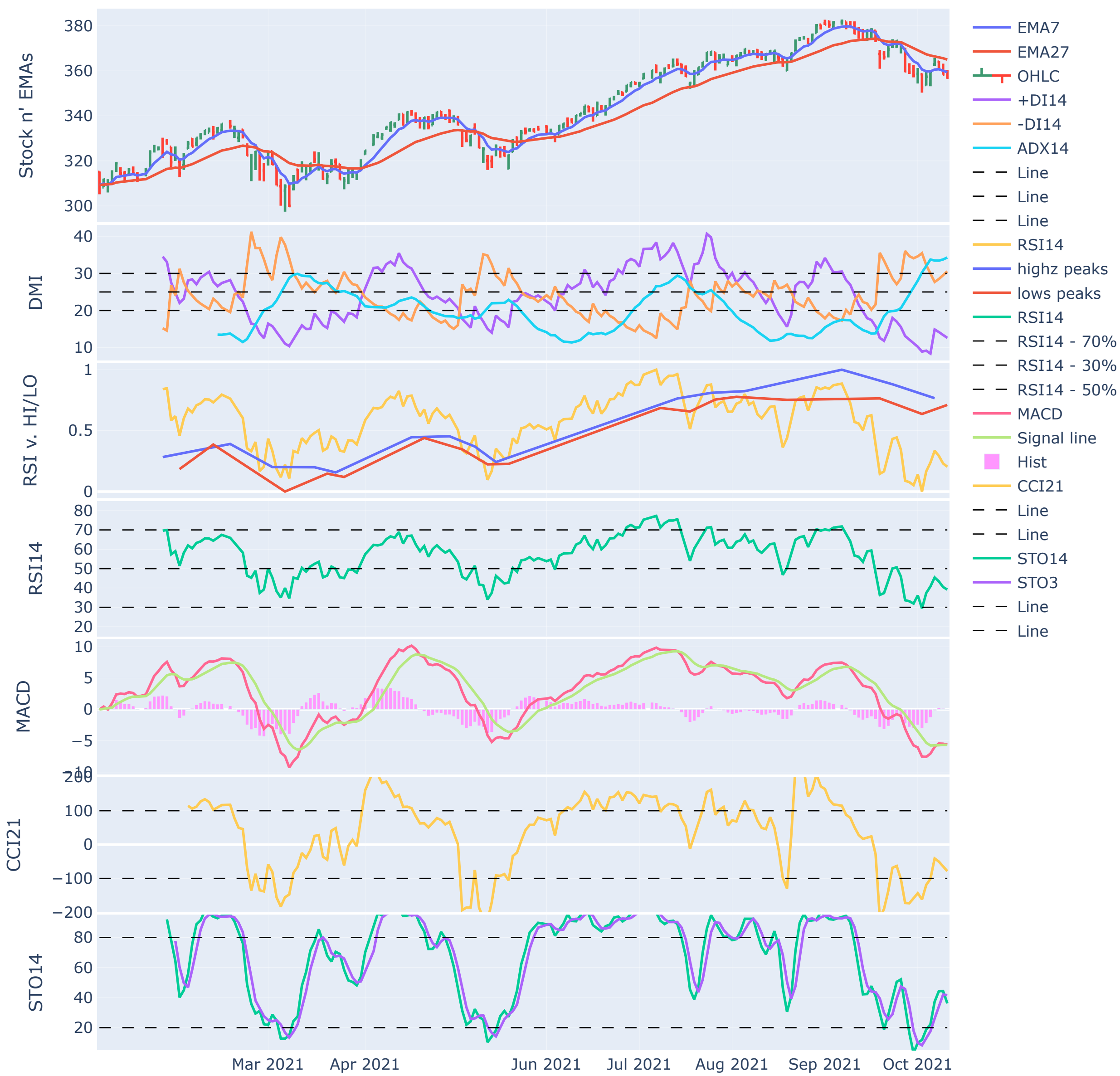

Stock n' EMAs
DMI
RSI v. HI/LO
RSI14
MACD
CCI21
STO14
Mar 2021
Apr 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
EMA7
EMA27
OHLC
+DI14
-DI14
ADX14
Line
Line
Line
RSI14
highz peaks
lows peaks
RSI14
RSI14 - 70%
RSI14 - 30%
RSI14 - 50%
MACD
Signal line
Hist
CCI21
Line
Line
STO14
STO3
Line
Line

Visual charts of a given stock portfolio

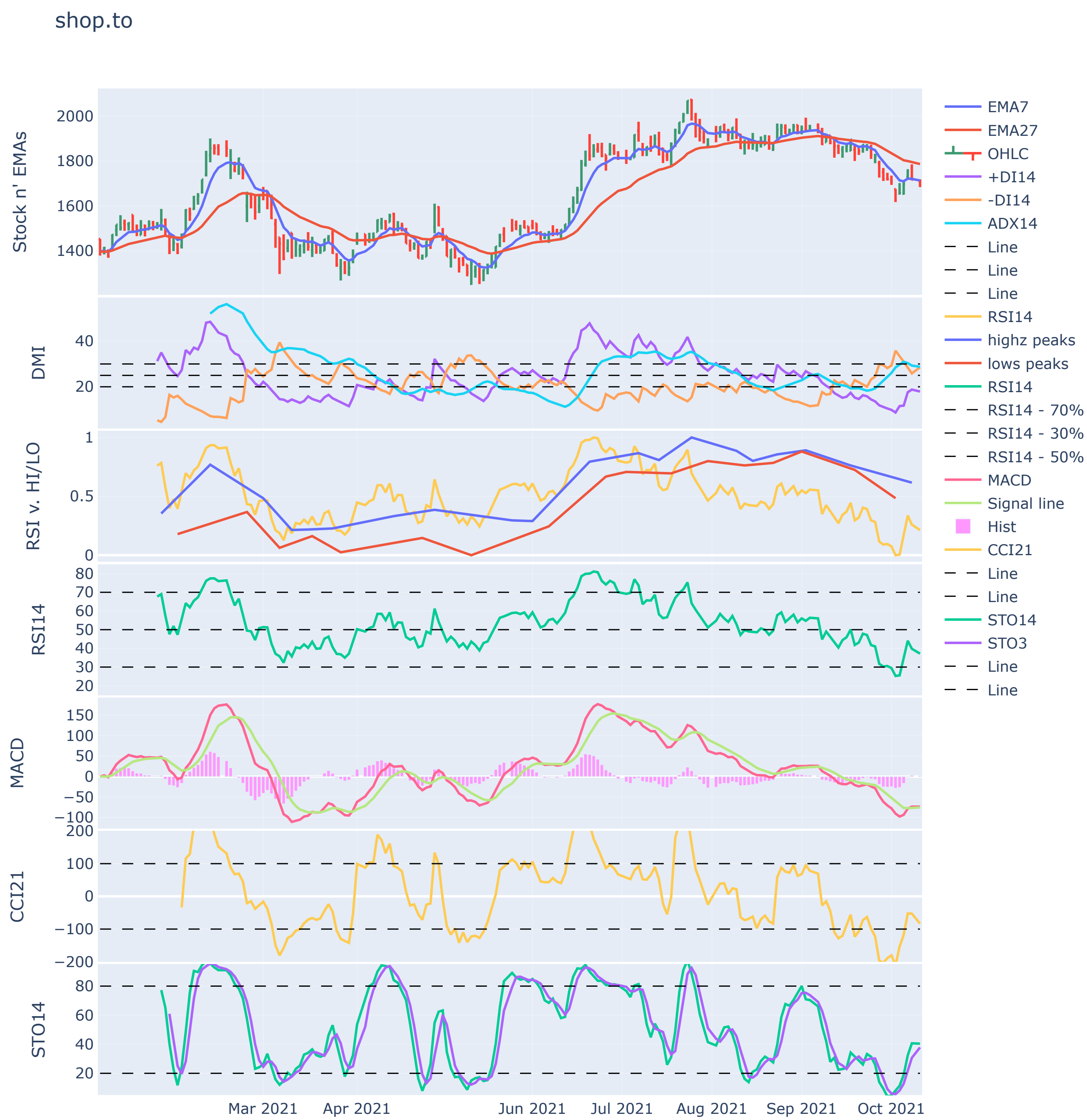
In this section, the charts of a list of stock are created.

spy
Stock n' EMAs
DMI
RSI v. HI/LO
RSI14
MACD
CCI21
STO14
Mar 2021
Apr 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
EMA7
EMA27
OHLC
+DI14
-DI14
ADX14
Line
Line
Line
RSI14
highz peaks
lows peaks
RSI14
RSI14 - 70%
RSI14 - 30%
RSI14 - 50%
MACD
Signal line
Hist
CCI21
Line
Line
STO14
STO3
Line
Line

qqq



Stock n' EMAs
380
360
340
320
300
DMI
40
30
20
10
RSI v. HI/LO
1
0.5
0
RSI14
80
70
60
50
40
30
20
MACD
10
5
0
−5
−10
CCI21
200
100
0
−100
−200
STO14
80
60
40
20
Mar 2021
Apr 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
EMA7
EMA27
OHLC
+DI14
-DI14
ADX14
Line
Line
Line
RSI14
highz peaks
lows peaks
RSI14
RSI14 - 70%
RSI14 - 30%
RSI14 - 50%
MACD
Signal line
Hist
CCI21
Line
Line
STO14
STO3
Line
Line



shop.to
Stock n' EMAs
DMI
RSI v. HI/LO
RSI14
MACD
CCI21
STO14
Mar 2021
Apr 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
EMA7
EMA27
OHLC
+DI14
-DI14
ADX14
Line
Line
Line
RSI14
highz peaks
lows peaks
RSI14
RSI14 - 70%
RSI14 - 30%
RSI14 - 50%
MACD
Signal line
Hist
CCI21
Line
Line
STO14
STO3
Line
Line

tsla

Stock n' EMAs
DMI
RSI v. HI/LO
RSI14
MACD
CCI21
STO14
Mar 2021
Apr 2021
Jun 2021
Jul 2021
Aug 2021
Sep 2021
Oct 2021
EMA7
EMA27
OHLC
+DI14
-DI14
ADX14
Line
Line
Line
RSI14
highz peaks
lows peaks
RSI14
RSI14 - 70%
RSI14 - 30%
RSI14 - 50%
MACD
Signal line
Hist
CCI21
Line
Line
STO14
STO3
Line
Line

4. Trends

In this section, we look at the various technical indicator recommendations for a given portfolio.

A user needs to input its portfolio of choice in Jupyter's homepage.

Out[18]:

	Symbol	EMA_recommendation	EMA_difference_trend	DMI_recommendation	MACD_recommendation	MACD_days_signal	RSI14_recommendation	RSI14_days_signal	CCI_recommendation	CCI_days_sig
3	tsla	buy	increasing	buy	buy	0.0	buy	151.0	buy	
0	spy	sell	increasing	sell	buy	3.0	sell	179.0	buy	
1	qqq	sell	increasing	sell	buy	2.0	buy	5.0	buy	
2	shop.to	sell	increasing	sell	buy	1.0	buy	3.0	buy	

5. Development of machine learning model to predict price direction

The objective of this section is to create a machine learning model to try to predict the direction of a stock several days in advance.

Concretly, we want to study how a stock react in the future based on present technical indicator values. We will therefore select a past timeframe and study how the stock reacts to its technical indicator (train a model). We will then select another past timeframe in the past and try to predict how the stock will react to the technical indicators in the future (test the model).

We will then be able to verify how a given stock reacts to a given set of technical indicators, determinie which type of machine learning model is the most efficient and better performs to predict the direction of the stock price.

Here's the breakdown of what we need to to:

1. Input a timeframe
2. Obtain stock data, prepare technical indicators and create variable of interest
3. Pre-process and split data into train and test sets
4. Identify the best machine learning, supervised learning classification algorithms
5. Development of a Random Forest classifier
6. Backtest using technical indicators

Timeframe input

The timeframe needs to cover a few years worth of data.

```
please enter start date and end date. Use this format: 2020-09-24
Enter a start date in YYYY-MM-DD format: 2018-01-01
Enter a end date in YYYY-MM-DD format: 2021-10-13

User inputs
Start date: 2018-01-01   End date: 2021-10-13
Start date in unix: 1609477200   End date in unix: 1634097600
```

Obtain stock data, prepare technical indicators and create variable of interest

We input the stock we want to predict future direction

```
Enter stock ticker of choice: qqq
```

We then obtain data online for the given timeframe and stock and derive several technical indicators using a function created in the beginning of the analysis.

Out[15]:

	Date	Open	High	Low	Close	Adj Close	Volume	EMA_short	EMA_long	EMA_difference_trend	EMA_recommendation	MACD_Short	MACD_Long	MACD	Signal	MA
0	2018-01-02	156.559998	158.529999	156.169998	158.490005	154.448959	32573300	158.490005	158.490005	NaN	sell	158.490005	158.490005	0.000000	0.000000	
1	2018-01-03	158.639999	160.169998	158.610001	160.029999	155.949661	29383600	158.875003	158.600005	increasing	buy	158.875003	158.600005	0.274999	0.068750	
2	2018-01-04	160.580002	160.789993	160.080002	160.309998	156.222534	24776100	159.233752	158.722147	increasing	buy	159.233752	158.722147	0.511605	0.179464	
3	2018-01-05	161.070007	162.029999	160.770004	161.919998	157.791489	26992300	159.905314	158.950565	increasing	buy	159.905314	158.950565	0.954749	0.373285	
4	2018-01-08	161.919998	162.630005	161.860001	162.550003	158.405441	23159100	160.566486	159.207668	increasing	buy	160.566486	159.207668	1.358818	0.619668	

In order to try to predict the direction of a stock days in advance, we have the indepedent variables which are the technical indicators values and we need to have a dependent variable, the variable of interest: the direction of a stock several days in advance.

A raise of the stock price corresponds to a value of 1 and a drop in the stock price corresponds to a value of 0. The algorithm will be trained and tested with this variable and ultimatly this is what we are trying to predict, this variable is the basis of this whole predictive analytics study.

The variable of interest is shown as "Gain_or_loss_binary" in the following table.

Out[16]:

	Date	Open	High	Low	Close	Adj_Close	Volume	Gain_or_loss_binary	Gain_or_loss_perc	EMA_difference_trend	EMA_recommendation	MACD_Short	MACD_Long	MACD	MACD_MACD	M
944	2021-10-01	358.60	361.25	354.38	360.18	360.18	56375500	0	-0.1	1	0	363.263	369.322	-6.059	-2.488	
945	2021-10-04	358.52	358.86	350.32	352.62	352.62	76766000	1	2.9	1	0	360.602	368.129	-7.527	-2.967	
946	2021-10-05	353.71	359.69	353.48	357.38	357.38	47232900	1	1.1	1	0	359.797	367.361	-7.565	-2.253	
947	2021-10-06	354.08	359.95	353.15	359.67	359.67	56806500	0	-0.4	0	0	359.765	366.812	-7.047	-1.302	
948	2021-10-07	362.80	365.69	362.25	362.97	362.97	39411200	0	-1.6	0	0	360.566	366.537	-5.971	-0.170	

Pre-process and split data into train and test sets

We will split the whole timeframe into two large data sets: one to train a machine learning model, the other to test it. We will split the data set using a 80%-20% sequential proportion.

The train set will be constituted of the first 80% of rows while the test set will be constituted of the last 20%.

```
test set is made of the first  725  rows, which constitute  80.0  percent of the dataset
train set is made of the last  181  rows which constitute  20.0  percent of the dataset
the sum of the train and test sets is  906  and must equal the length of the whole data set which is 906
```

Out of the dataset, we select the technical indicators that we want and then scale them to improve the quality of the results. Scaling

We need to define the X and Y of our datasets:

x_train will be constituted of the technical indicators in the first 80% of rows.
y_train will be constituted of the variable of interest: the stock direction 4 days in advance in the first 80% of rows.

x_test will be constituted of the technical indicators in the last 20% of rows.
y_test will be constituted of the variable of interest: the stock direction 4 days in advance in the last 20% of rows .

```
X_train dimensions are (725, 14)  same length as the train fraction: 725
X_test dimensions are (181, 14)  same length as the test fraction: 181
y_train dimensions are (725,)  same length as the train fraction: 725
y_test dimensions are (181,)  same length as the test fraction: 181
```

Identify the best machine learning, supervised learning classification algorithms

We will test several classifier to check their ability to get trained using the train data set.

We compare classifiers for the sake of this analysis because what we want to do is for a given day and associated technical indicators values, determine if the stock will raise of drop several days later in time.

Basically, we want to classify each day in one of two bins: Yes or No and act upon.

The classifiers are:

1. Logistic regression: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html)
2. KNeighbors classifier: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html)
3. Support Vector classification: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html (https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html)
4. Decision Tree classifier: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html)
5. Random Forest classifier: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)
6. Gaussian Naive Bayes classifier: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html (https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html)

```
               classifier  train_score  training_time
0     Logistic Regression     0.605517       0.078125
1       Nearest Neighbors     0.786207       0.031250
2  Support Vector Machine     0.605517       0.062500
3           Decision Tree     1.000000       0.031250
4           Random Forest     1.000000       0.328125
5            Naives Bayes     0.626207       0.000000
```

As it can be seen the Decision Tree and Random Forest classifiers obtain score of 1.00 (meaning it can relate 100% of Ys to Xs) though Random Forest is the fastest one. We select the Random Forest to go forward as by making several random trees, it can usually produce more accurate resuts.

Development of a Random Forest classifier

The idea of this section is to build upon the above and train a Random Forest model in order to predict the direction of a stock.

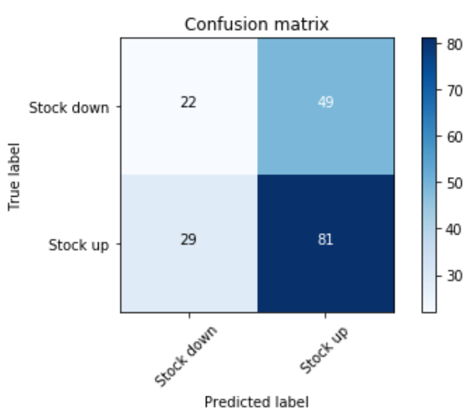
The classifier will be trained with the train data set for then be tested with the test data set. For both train and test sets, we know the direction of the stock a given amount of days later in time.

The model will be trained with the training data set to "understand" how technical indicators values are associated with the stock price rise or fall (this is what was done above to select the best classifier).

The model will then be tested using the test data set. Presented with the technical indicators, the model will predict the direction of a stock a certain amount of days in time. Since we know how the stock behaved for this data set, we will be able to measure the global accuracy of the algorithm and understand any biased it might have.

When statisfied with the results, we will have the model predict current day technical indicators! (tbd in the future)

```
The model accuracy score is:  0.57


The model classification report is:
              precision    recall  f1-score   support

           0       0.43      0.31      0.36        71
           1       0.62      0.74      0.68       110

    accuracy                           0.57       181
   macro avg       0.53      0.52      0.52       181
weighted avg       0.55      0.57      0.55       181

Confusion matrix, without normalization
```



The accuracy score gives us only the global accuracy of the classifier but does not give the accuracy of the model to predict false or positive values (stock goes down or up).

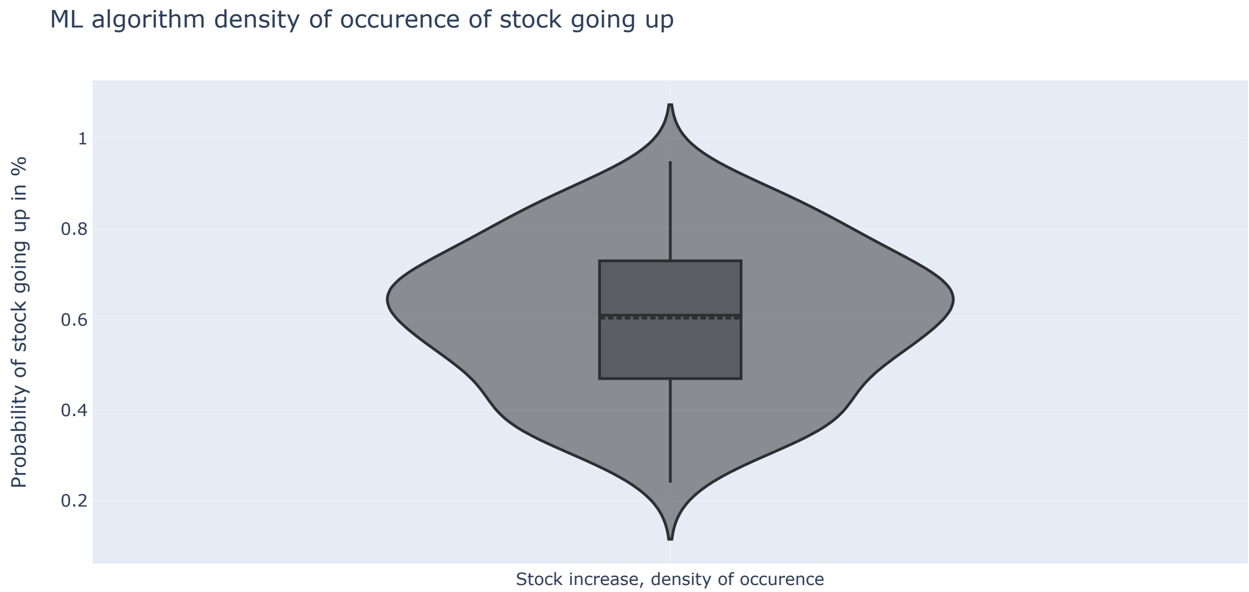
As it can be seen in the matrix, the algorithm does a much better job at predicting a stock price that goes up than it does at predicting a stock that goes down.

Next, we will visually compare the test sets and the predictions done by the model.

```
Let's first have a visual look at the test set and the prediction of the algorithm

Test set: if value is 1 stock went down, if value is 0 stock went down:
[1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 1 1 1 1 1 1 1 0
 0 0 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 0 1 0 0
 0 0 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 1 1 1 1 0 0 0 0 1 1 1 1 1 0 0 1 0 1 1 1 1 1 0 0 0 1 1 0 0 0 1 1 1 1
 0 1 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 1 1 1 0 0 0 1 0 0 0 1 1 0 0]


Model predictions: if value is 1 stock will go up, if value if 0 stock will go down:
[1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 1 1 0 1 0 0 1 0 1
 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 1 0 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1
 1 1 0 1 1 0 0 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1 1 0 1 1 0
 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1
 0 1 1 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1]
```

As we know, the machine learning predict the movement of a stock, up or down. For each prediction, its calculates a probability between 0% and 100%.

Using a violin plot, we will show the density of occurence of each probability.



As seen in the violin plot above, there is a bump in the density of occurence between 60% and 80%. This means that the majority of time, the algorithm will predict that the stock will go up with 60-80% chance.

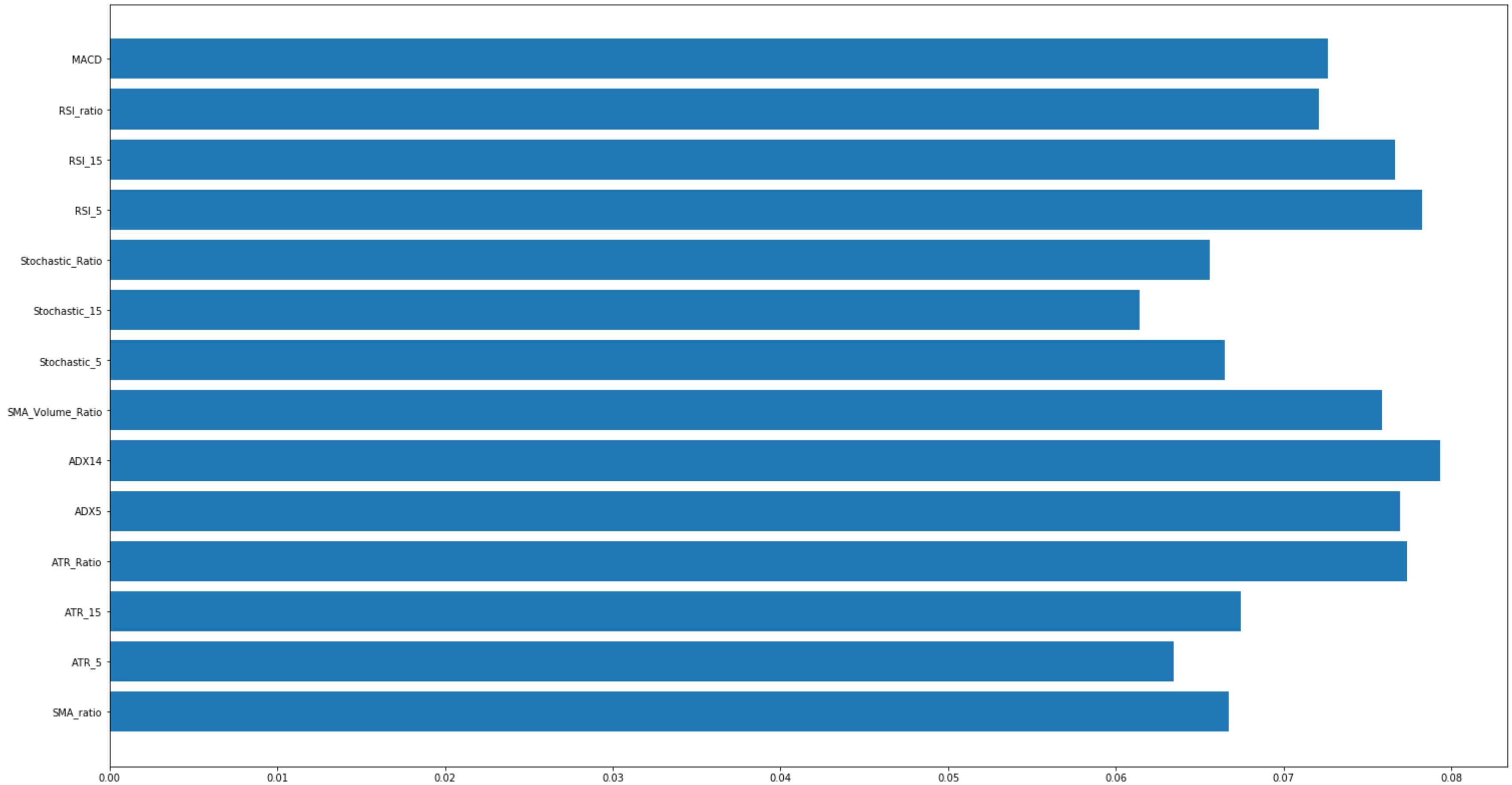
Specific technical indicator importance

In this section we evaluate the importance of each technical indicator in predicting the stock direction.

The blue bars are the feature importances of the forest, along with their inter-trees variability represented by the error bars (https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html (https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html)).

This section can be used to do some feature engineering by optimizing the selection of technical indicators.

```
Out[27]: <BarContainer object of 14 artists>
```



As seen above, each technical indicators has a different influence on helping the machine learning model predict the direction of a stock price.

Our experience has shown that the timeframe, the stock and the mix of technical indicators chosen all have an impact on the graph above.

Evaluation and backtest

In this section, we look at how the model performs when we compare its prediction to a "buy and hold" strategy.

A "buy and hold" strategy consists of buying a certain amount of stock (10000$ in this example) with the hope to see the value of the invetment grow in time without interactions.

On the other side, the algorithm tries to predict the rise or the fall of the stock in a short time frame. When the model predicts a rise in the stock value, we ask the model to buy the stock if no positions are currently held or we ask the model to hold on to the stock if positions are currently held. When the model predicts a drop in the stock value, we ask the model to sell the stock if positions are currently held or we ask the model to stay out of the market if positions are not currently held.

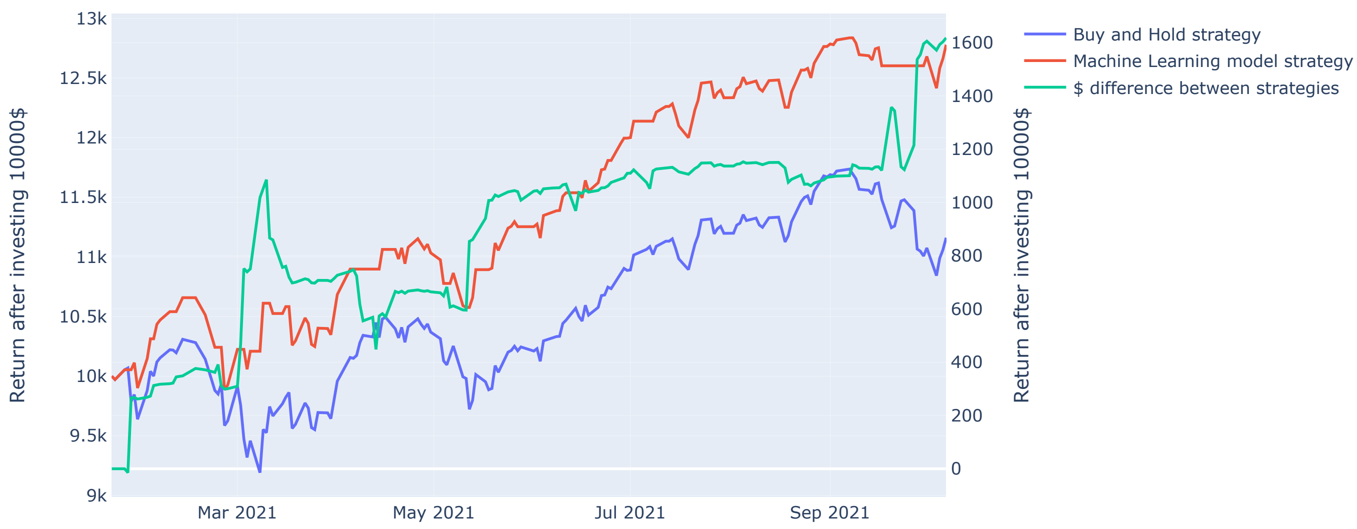
Same as for the "buy and hold" strategy, the machine learning strategy starts with an investment of 10000$.

Finally, we are going into dive into the Modern Portfolio Theory (MPT: [https://www.investopedia.com/terms/m/modernportfoliotheory.asp (https://www.investopedia.com/terms/m/modernportfoliotheory.asp)](https://www.investopedia.com/terms/m/modernportfoliotheory.asp)) and look into the return and risks of both strategies. We will calculate the "Risk" as being the standard deviation of the daily return, the "Return" being the ratio on between the final value and the initial investment and the "Performance Ratio" being the ratio of the "Return" and the "Risk".

Out[28]:

	index	Date	Open	Adj_Close	Gain_or_loss_binary	Prediction	buy_n_hold	ML_model	buy_n_hold_Yield	ML_Yield
0	768	2021-01-21	325.15	325.210	1	1	10000.000000	10000.000000	0.000000	0.000000
1	769	2021-01-22	325.29	324.273	0	1	9971.187848	9971.187848	0.997119	0.997119
2	770	2021-01-25	328.91	326.954	0	0	10053.626887	10053.626887	1.008268	1.008268
3	771	2021-01-26	328.85	327.432	0	0	10068.325082	10053.626887	1.001462	1.000000
4	772	2021-01-27	326.26	318.305	1	1	9787.675656	10053.626887	0.972126	1.000000
...	...	...	...	...	...	...	...	...	...	...
176	944	2021-10-01	358.60	360.180	0	1	11075.305187	12681.213986	1.006202	1.006202
177	945	2021-10-04	358.52	352.620	1	1	10842.840011	12415.041579	0.979010	0.979010
178	946	2021-10-05	353.71	357.380	1	1	10989.206974	12582.631613	1.013499	1.013499
179	947	2021-10-06	354.08	359.670	0	1	11059.623013	12663.257911	1.006408	1.006408
180	948	2021-10-07	362.80	362.970	0	1	11161.095907	12779.444279	1.009175	1.009175

181 rows × 10 columns




```
Buy_n_hold performance metrics:
Risk: 7.5 %
Growth: 111.6 %
Buy_n_hold model performance ratio: 111.6 / 7.5 = 14.9


Machine learning algorithm performance metrics:
Risk: 7.5 %
Growth: 127.8 %
ML model performance ratio: 127.8 / 7.5 = 17.0
```

As we can see above, for the given timeframe, the Random Forest machine learning model yields an higher growth and a lower risk than the "buy and hold" strategy thus leading to a higher performance ratio!

What needs to be noted though is that while the results above seem very promising in regards of the machine learning algorithm versus the buy and hold strategy, the results can vary significantly in relation to when the algorithm gets in the market. It would be very interesting in the future to study the market entry point in relation with the machine learning model performance.

6. Backtest using technical indicators

In this section, we will compare our Random Forest machine learning model to more traditionnal technical indicator investing techniques.

For each technical indicator investing technique, we will try to find the respective parameters that yield the highest performance ratio (as described in the previous section).

Buy and Hold strategy

```
Buy and hold performance:
```

Out[127]:

	Buy_hold_standard_deviation	Buy_hold_growth	Buy_hold_Performance
0	7.523605	111.61	14.834644

Exponential Moving Average strategy

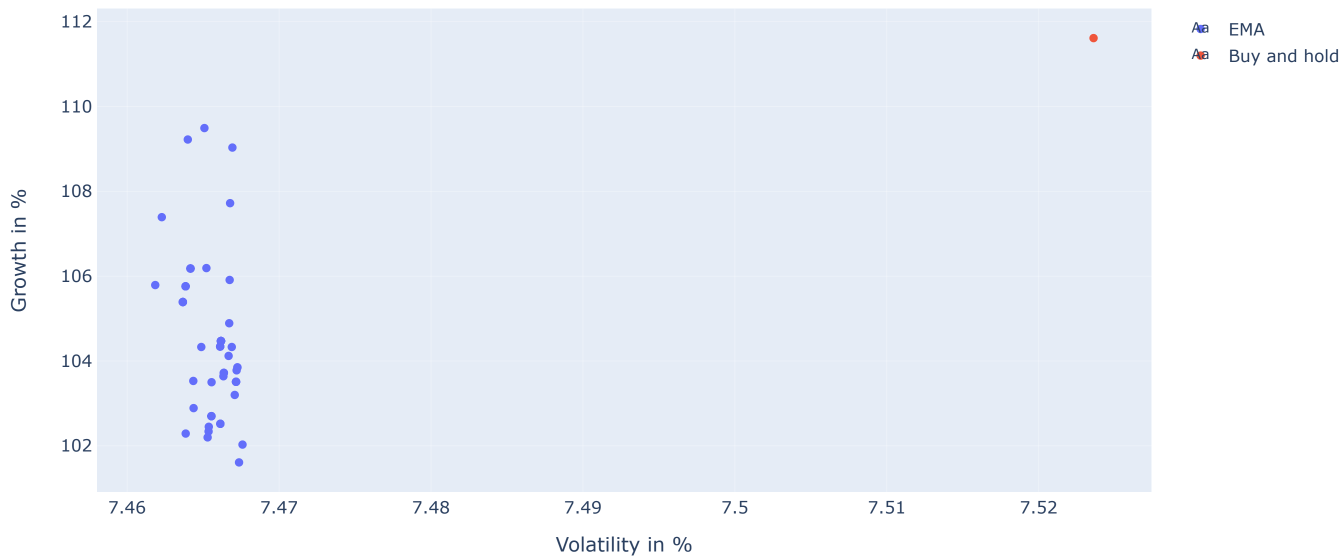
In this strategy, an investor holds a position when the short EMA value is above the long EMA and clears its position when the EMA value is smaller than the long EMA.

We test for various period lengths and try to find the combination that yields the best performance.

We sort the results with the highest performance ratio first.

Out[129]:

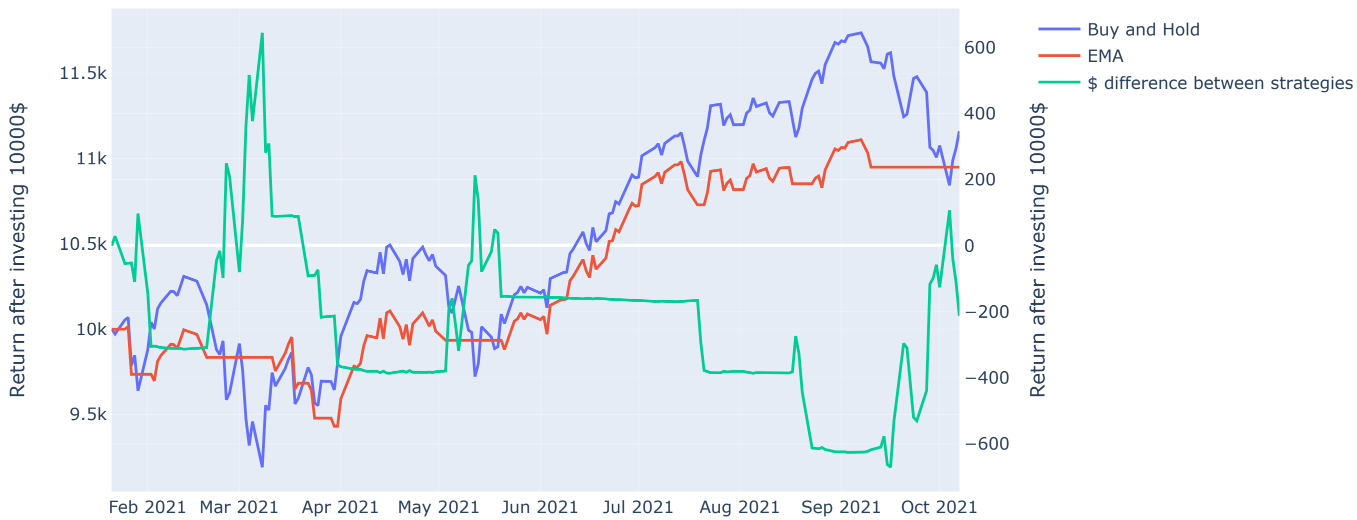
	EMA_short	EMA_long	Money_EMA	Standard_deviation	Growth	Performance_ratio
3	3.0	9.0	10949.299818	7.465086	109.49	14.666944
2	3.0	8.0	10921.504850	7.463993	109.22	14.632919
7	4.0	9.0	10903.016339	7.466928	109.03	14.601721
6	4.0	8.0	10771.664414	7.466777	107.72	14.426573
1	3.0	7.0	10739.262036	7.462281	107.39	14.391042
55	8.0	21.0	10618.494995	7.464170	106.18	14.225293
54	8.0	20.0	10618.494995	7.464170	106.18	14.225293
53	8.0	19.0	10618.494995	7.464170	106.18	14.225293
44	7.0	22.0	10618.494995	7.464170	106.18	14.225293
43	7.0	21.0	10618.494995	7.464170	106.18	14.225293



From the above, we select the set with the highest performance ratio and show on a graph the return distribution compared to a buy and hold strategy.

Out[131]:

	Date	Open	High	Low	Close	Adj_Close	Yield_buy_hold	Yield_strategy	EMA_short	EMA_long	EMA_money	Buy_N_Hold
176	2021-10-01	358.600006	361.250000	354.380005	360.179993	360.179993	1.006202	1.0	360.185374	364.593199	10949.299818	11075.302146
177	2021-10-04	358.519989	358.859985	350.320007	352.619995	352.619995	0.979010	1.0	356.402684	362.198558	10949.299818	10842.837090
178	2021-10-05	353.709991	359.690002	353.480011	357.380005	357.380005	1.013499	1.0	356.891345	361.234848	10949.299818	10989.204323
179	2021-10-06	354.079987	359.950012	353.149994	359.670013	359.670013	1.006408	1.0	358.280679	360.921881	10949.299818	11059.620590
180	2021-10-07	362.799988	365.690002	362.250000	362.970001	362.970001	1.009175	1.0	360.625340	361.331505	10949.299818	11161.093089



We can see than despite choosing the ultimate periods for the EMA strategy, the performance ratio and the value of investment are smaller than for the "buy and hold" strategy. This could be explained by the fact that EMAs follow trends when they are already established.

One upside of the EMA strategy is that the volatility is lower than the buy and hold strategy and this can be seen on the graph above. Indeed, EMA's line can drop but never lower than a certain amount hence reducing the overal volatility.

Commodity Channel Index (CCI) and Slow Stochastics (STO)

In this strategy, an investor reacts to two technical indicators to buy or sell its positions. The idea is that these two indicators are too volatile to be obeyed individually hence they are joined in a pair to provide buy or sell signals.

The CCI indicator buy signals used it here is when the indicator crosses the -100 or +100 lines going up while the sell signal used in this analysis are when the indicator crossed the -100 or +100 lines going down.

The STO indicator buy signals used it here is when the indicator crosses the +20 or +80 lines going up while the sell signal used in this analysis are when the indicator crossed the +20 or +80 line lines going down.

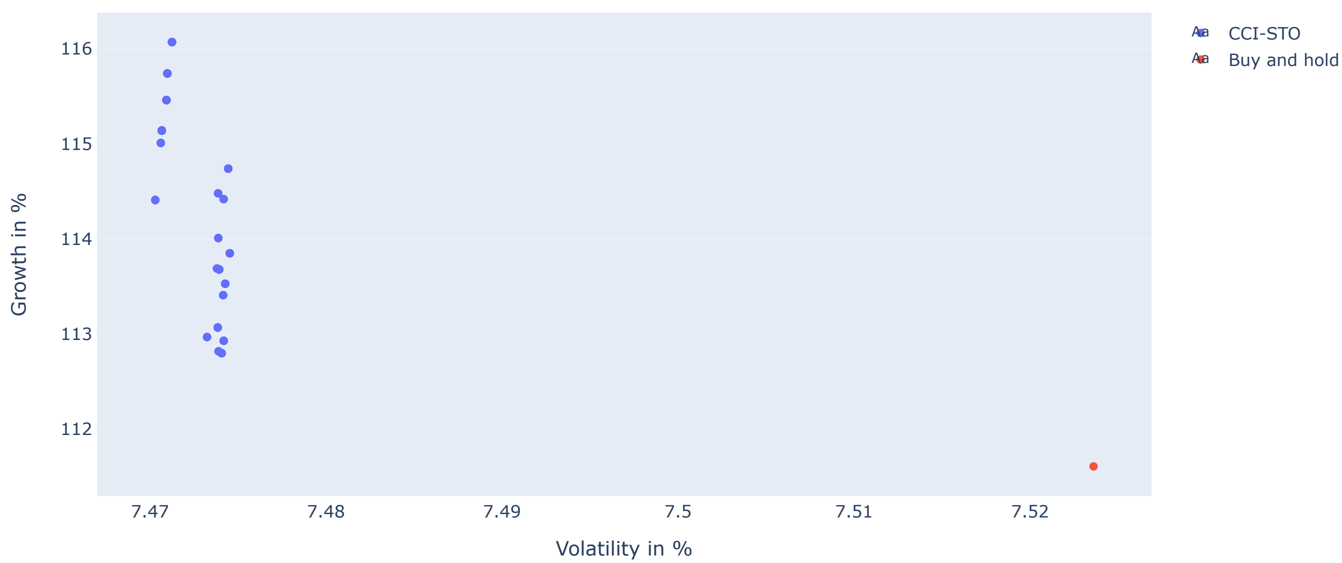
The algorithm reacts when both indicators produce the same type of signal (buy or sell).

We test for various period lengths and try to find the combination that yields the best performance.

We sort the results with the highest performance ratio first.

Out[137]:

	CCI_number_period	Typ_pric_divider	STO_long	STO_short	Money	Standard_deviation	Growth	Performance_ratio
2	17.0	2.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
26	18.0	3.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
20	18.0	2.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
32	18.0	4.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
8	17.0	3.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
14	17.0	4.0	14.0	4.0	11606.654788	7.471247	116.07	15.535559
86	21.0	4.0	14.0	4.0	11574.082010	7.470985	115.74	15.491933
110	23.0	2.0	14.0	4.0	11574.082010	7.470985	115.74	15.491933
122	23.0	4.0	14.0	4.0	11574.082010	7.470985	115.74	15.491933
116	23.0	3.0	14.0	4.0	11574.082010	7.470985	115.74	15.491933

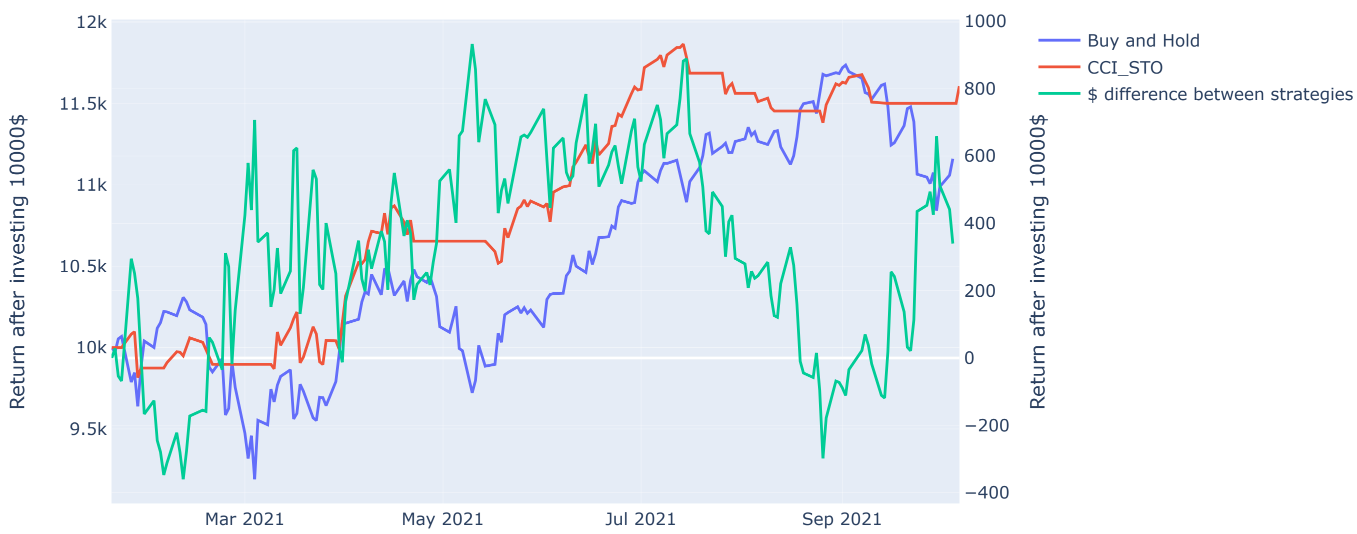


From the above, we select the set with the highest performance ratio and show on a graph the return distribution compared to a buy and hold strategy.

Out[139]:

	Date	Money
0	2021-01-19	10000.000000
1	2021-01-20	10000.000000
2	2021-01-21	10000.000000
3	2021-01-22	10000.000000
4	2021-01-25	10082.662059
...	...	...
178	2021-10-01	11501.131407
179	2021-10-04	11501.131407
180	2021-10-05	11501.131407
181	2021-10-06	11501.131407
182	2021-10-07	11606.654788

183 rows × 2 columns



For the CCI-STO strategy, we can see that the performance ratio and the value of investment are higher than for the "buy and hold" strategy.

Plus, similar to EMA strategy, one advantage is that the volatility is lower than the buy and hold strategy and this can be seen on the graph above. Indeed, the line for the CCI-STO strategy never drops as much as it sometimes does for the buy and hold strategy. This can be interesting for investors who can't support loosing more than a certain amount of their investment.

Moving Average Convergence Divergence (MACD)

In this strategy, an investor holds a position when the MACD line crosses the Signal line going up and clears its position when the MACD line crosses the Signal line going down.

We test for various period lengths and try to find the combination that yields the best performance.

Out[143]:

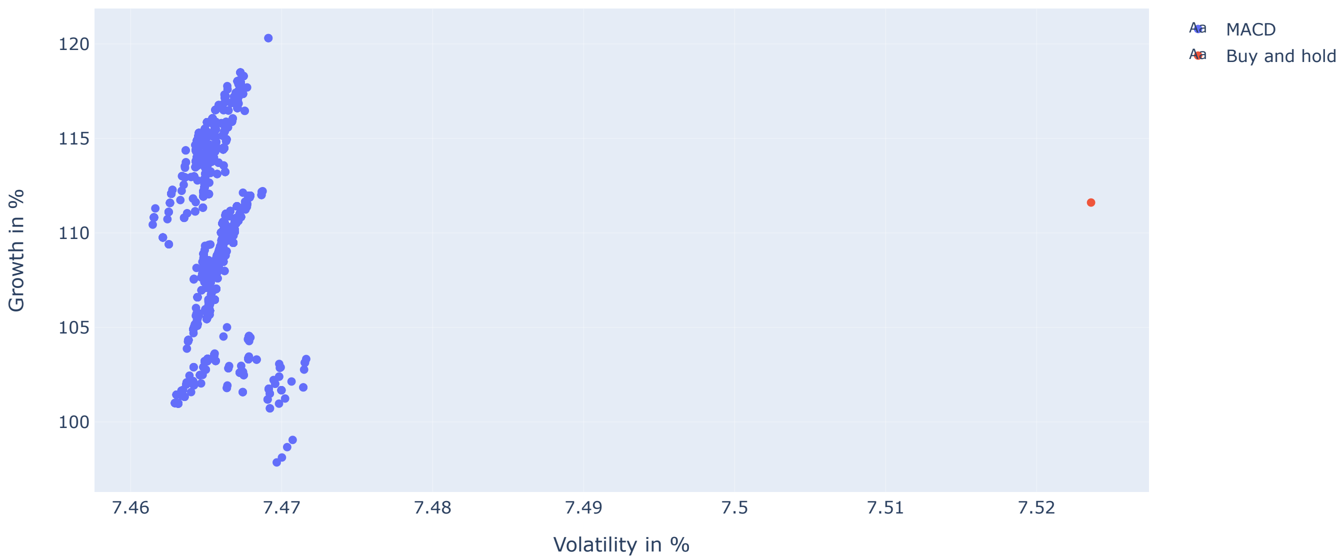
	Date	Open	Close	Adj_Close	Short	Long	MACD	Signal	Buy_N_Hold	MACD_money	Yield_strategy
0	2021-01-21	325.149994	326.359985	325.210083	326.359985	326.359985	0.000000	0.000000	10000.000000	10000.000000	0.000000
1	2021-01-22	325.290009	325.420013	324.273438	326.203323	326.335570	-0.132247	-0.017633	9971.198771	10000.000000	1.000000
2	2021-01-25	328.910004	328.109985	326.953949	326.521100	326.381659	0.139441	0.003310	10053.622753	10000.000000	1.000000
3	2021-01-26	328.850006	328.589996	327.432251	326.865916	326.439018	0.426898	0.059789	10068.330231	10014.629033	1.001463
4	2021-01-27	326.260010	319.429993	318.304504	325.626596	326.256966	-0.630370	-0.032233	9787.657906	9735.453723	0.972123
...	...	...	...	...	...	...	...	...	...	...	...
176	2021-10-01	358.600006	360.179993	360.179993	365.713242	363.391317	2.321925	8.791403	11075.302146	10289.651904	1.000000
177	2021-10-04	358.519989	352.619995	352.619995	363.531034	363.111543	0.419492	7.675149	10842.837090	10289.651904	1.000000
178	2021-10-05	353.709991	357.380005	357.380005	362.505863	362.962672	-0.456809	6.590888	10989.204323	10289.651904	1.000000
179	2021-10-06	354.079987	359.670013	359.670013	362.033221	362.877148	-0.843927	5.599579	11059.620590	10289.651904	1.000000
180	2021-10-07	362.799988	362.970001	362.970001	362.189351	362.879560	-0.690209	4.760941	11161.093089	10289.651904	1.000000

181 rows × 11 columns

We sort the results with the highest performance ratio first.

Out[145]:

	X	Y	Z	Money	Standard_deviation	Growth	Performance_ratio
341	5.0	15.0	8.0	12031.065894	7.469118	120.31	16.107659
175	4.0	14.0	10.0	12031.065894	7.469118	120.31	16.107659
400	5.0	20.0	7.0	11848.858575	7.467263	118.49	15.867929
222	4.0	18.0	9.0	11848.858575	7.467263	118.49	15.867929
591	6.0	19.0	6.0	11848.858575	7.467263	118.49	15.867929
388	5.0	19.0	7.0	11848.858575	7.467263	118.49	15.867929
257	4.0	21.0	8.0	11848.858575	7.467263	118.49	15.867929
71	3.0	14.0	14.0	11848.858575	7.467263	118.49	15.867929
82	3.0	15.0	13.0	11848.858575	7.467263	118.49	15.867929
387	5.0	19.0	6.0	11830.308276	7.467475	118.30	15.842035

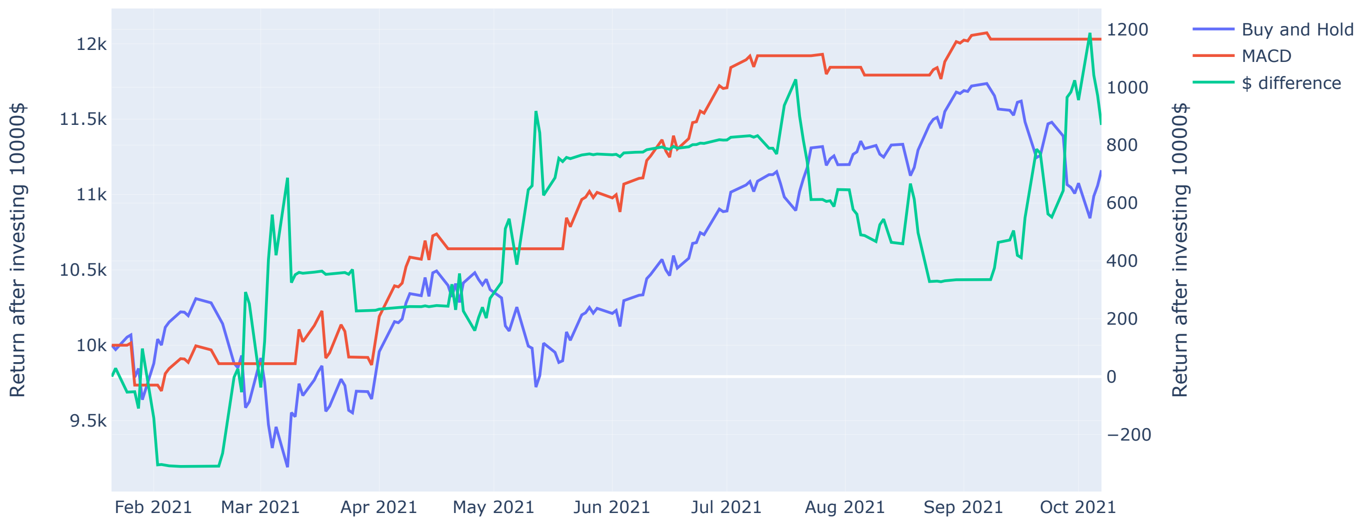


From the above, we select the set with the highest performance ratio and show on a graph the return distribution compared to a buy and hold strategy.

Out[147]:

	Date	Open	Close	Adj_Close	Short	Long	MACD	Signal	Buy_N_Hold	MACD_money	Yield_strategy
0	2021-01-21	325.149994	326.359985	325.210083	326.359985	326.359985	0.000000	0.000000	10000.000000	10000.000000	0.000000
1	2021-01-22	325.290009	325.420013	324.273438	325.983996	326.234655	-0.250659	-0.045574	9971.198771	10000.000000	1.000000
2	2021-01-25	328.910004	328.109985	326.953949	326.834392	326.484699	0.349692	0.026292	10053.622753	10000.000000	1.000000
3	2021-01-26	328.850006	328.589996	327.432251	327.536633	326.765406	0.771228	0.161735	10068.330231	10014.629033	1.001463
4	2021-01-27	326.260010	319.429993	318.304504	324.293977	325.787351	-1.493373	-0.139194	9787.657906	9735.453723	0.972123
...	...	...	...	...	...	...	...	...	...	...	...
176	2021-10-01	358.600006	360.179993	360.179993	360.930027	367.030127	-6.100101	-3.891757	11075.302146	12031.065894	1.000000
177	2021-10-04	358.519989	352.619995	352.619995	357.606014	365.108776	-7.502762	-4.548303	10842.837090	12031.065894	1.000000
178	2021-10-05	353.709991	357.380005	357.380005	357.515610	364.078274	-6.562663	-4.914551	10989.204323	12031.065894	1.000000
179	2021-10-06	354.079987	359.670013	359.670013	358.377371	363.490505	-5.113134	-4.950657	11059.620590	12031.065894	1.000000
180	2021-10-07	362.799988	362.970001	362.970001	360.214423	363.421105	-3.206682	-4.633570	11161.093089	12031.065894	1.000000

181 rows × 11 columns



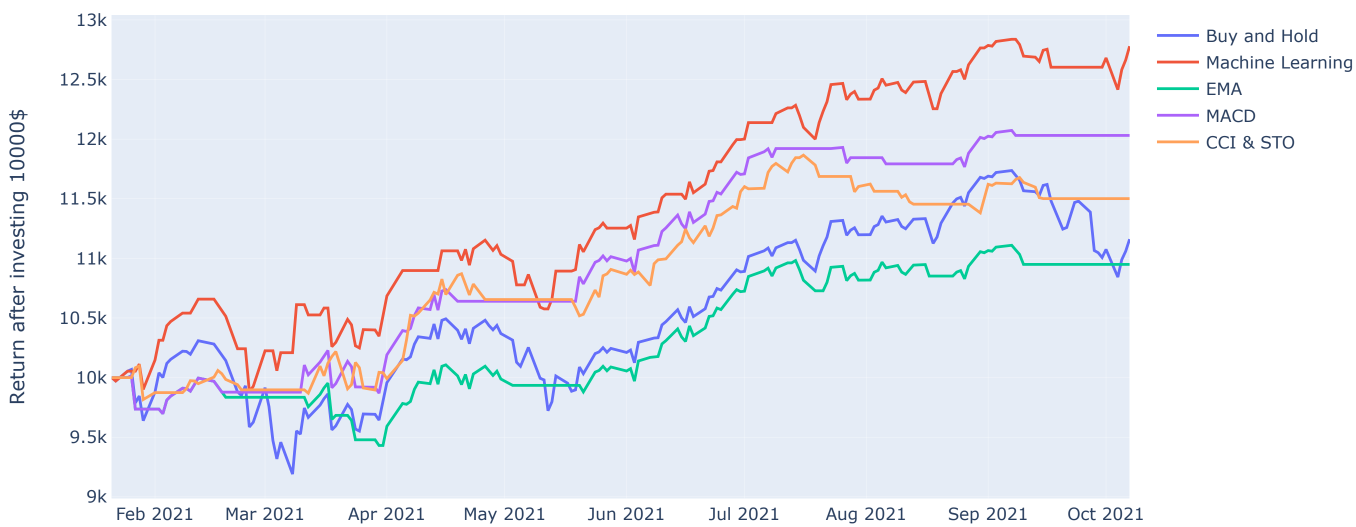
The MACD strategy is the best conventional technical indicator strategy analysed in this study. Indeed, the value of investment rarely drops more than a few hundred dollars and the end, it is higher than the buy and hold strategy.

Strategy side-by-side comparison

Out[151]:

	Open	Adj_Close	Date	EMA	MACD	CCI_STO	Machine_Learning	Buy_n_hold
0	325.15	325.210	2021-01-21	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
1	325.29	324.273	2021-01-22	10000.000000	10000.000000	10000.000000	9971.187848	9971.198771
2	328.91	326.954	2021-01-25	10000.000000	10000.000000	10000.000000	10053.626887	10053.622753
3	328.85	327.432	2021-01-26	10014.629033	10014.629033	10000.000000	10053.626887	10068.330231
4	326.26	318.305	2021-01-27	9735.453723	9735.453723	10082.662059	10053.626887	9787.657906
...	...	...	...	...	...	...	...	...
176	358.60	360.180	2021-10-01	10949.299818	12031.065894	11501.131407	12681.257640	11075.302146
177	358.52	352.620	2021-10-04	10949.299818	12031.065894	11501.131407	12415.084316	10842.837090
178	353.71	357.380	2021-10-05	10949.299818	12031.065894	11501.131407	12582.674928	10989.204323
179	354.08	359.670	2021-10-06	10949.299818	12031.065894	11501.131407	12663.301503	11059.620590
180	362.80	362.970	2021-10-07	10949.299818	12031.065894	11501.131407	12779.488272	11161.093089

181 rows × 8 columns



For the given time period and stock, two strategies are beating the "buy and hold" strategy: the Random Forest machine learning strategy and the MACD strategy.

It seems very promising but as stated earlier in this analysis, careful attention should be given to the time of entry into the market. This parameter is of great interest as different time of entry provides great differences in strategy outcomes (sometime positive, sometime negative). This must therefore be understood when looking at the results.

A path of further development to understand the above could be to study the relation of entry point vs. growth over the period and see what kind of relation can be obtained.

Ultimatly, given an entry point, we could try to derive a probability of growth with a certain statistical significance.