NEOMA BUSINESS SCHOOL

Applied Data Science

SP500 Financial Index tracking

Msc Finance & Big Data
2021-2022

Saad HANDAR
Mohamed TOUZI
Mohamed DAKKOURI
Basma EL KHAMLICHI
Josselin MASSE
Satnam SINGH

Introduction :

A tracker fund is an index fund that tracks a broad market index or a portion of it. Tracking funds, also known as index funds, are designed to provide investors with low-cost exposure to the entire index. These funds attempt to track the holdings and performance of specific indices structured as ETFs or alternative investments to achieve the fund's tracking objectives.

The term "tracking fund" originates from the tracking function that drives the management of index funds. Tracking funds attempt to replicate the performance of market indices. Market innovation has significantly expanded the number of tracker funds available in the investable market.

Investing in index funds is a form of passive investing. Index funds were originally introduced to provide investors with a low-cost investment vehicle that would provide exposure to the many securities that make up a market index. The main benefit of this strategy is the lower expense ratio of index funds.

Popular indexes for U.S. market exposure include the S&P 500, Dow Jones Industrial Average, and the Nasdaq Composite. Investors often choose traditional tracker funds because a majority of investment fund managers fail to beat broad market indexes on a consistent basis.

In this context, we will study the case of the S&P 500 and we will try to build a model to track this index based on the stock assets from 2013 to 2017. We will select K=50.

We will respect the 3 instructions of sparsing the portfolio assets selection. First, select randomly K assets from index assets. Then, with the most correlated stocks with SP500 and finally with clustering using KMedoids.

For the weight estimation, we will have to find the weights by minimizing square loss first and then by minimizing downside risk.

To sum up, our work consists of building a machine learning model to predict SP500 return from sparse portfolio regression.

Methodology :

I. Problem scoping :

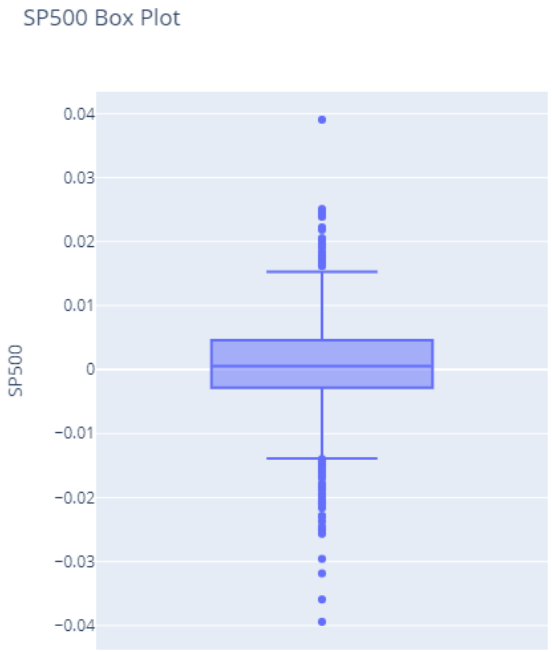
- Construct SP500 tracker with only 50 stocks in order to minimize costs.
- Minimize the downside risk.
- Static weights VS weights rebalancing every 5 days.

II. EDA : Exploratory data analysis :

We took the time to understand the data before starting any machine learning model and to analyze the main statistical characteristics of our data set.

And as we are working with return here, we already know that the returns time series should be stationary and the distribution is normal but with fat tails, mean 0 and a given variance (we don't talk about heteroscedasticity to keep it simple). So, we should fill the missing value with 0.

We tackle many other aspects of the data, for example, here is the boxplot of the sp500 returns:



III. Data Cleaning :

1- For every problem, there is a specific way to address missing data, either with the mean, the median, 0 or fill with the previous or next value. But as we've seen in the Exploratory data analysis, as we are working with returns rather than prices, we filled missing data with 0. (If we were working with prices, we couldn't have done the same, because it is not relevant from a financial point of view to fill missing prices with 0).
2- We proceeded to clean the CSV, keeping the names of stocks columns and renaming the Data column and the first column with sp500.
Now we have the final data frame with dates, the return of the SP500, and the other stocks from January 2013 to December 2017.

IV. Feature selection :

NB : for feature engineering, we didn't do one hot encoding because we are not dealing with categorical data. We are dealing directly with numerical data so no need for one hot encoding.
For feature selection, we are aware of explicative factors (a.k.a. features), usually designed by financial experts, like the momentum, or from mathematical analysis, are model parameters that can be fitted to a training data set. But here we worked only with returns and we didn't add other features to the model.

V. Modeling :

For this exercise we used tensor flow MeanSquaredError for the first part and in the second part, we used the MeanSquaredError with Relu activation function. As a bonus, we used Sklearn Ridge model.

VI. Evaluation :

To evaluate our model, we used the MSE metric.

VII. Monitoring :

We plotted the predicted returns and the actual returns and also, we created an index to track the performance of our index and the SP500 see the graphics below.

For selecting the stock, we have 3 methods to do.

Minimizing square loss

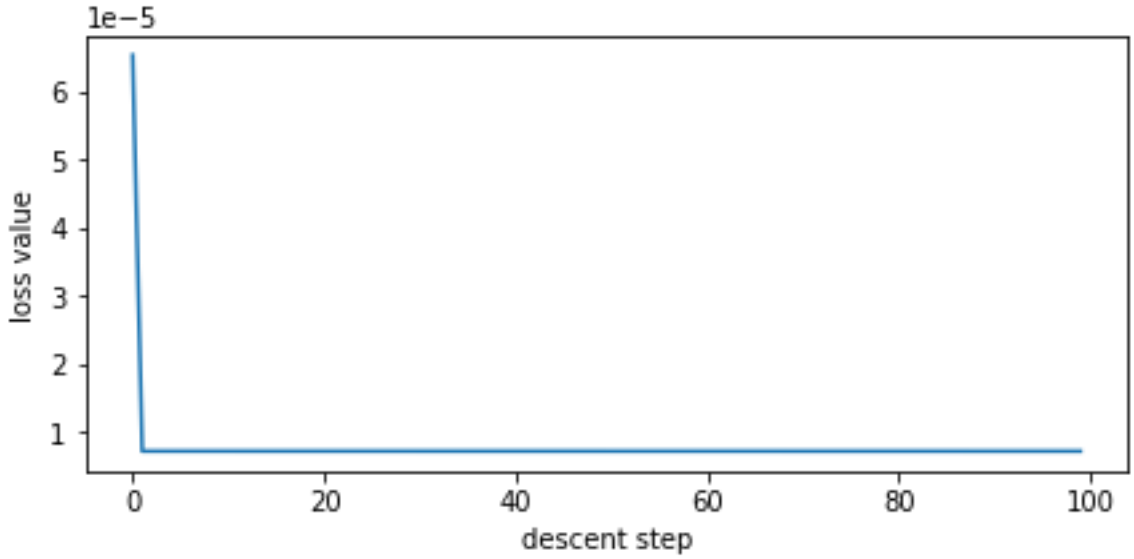
Tensferflow :

First, we will proceed by **selecting 50 random stocks** with using the function (Random.Choice).

Now for the study, we separate the data to train and test and we used a train ratio 0.8, then we split to train and test datasets.

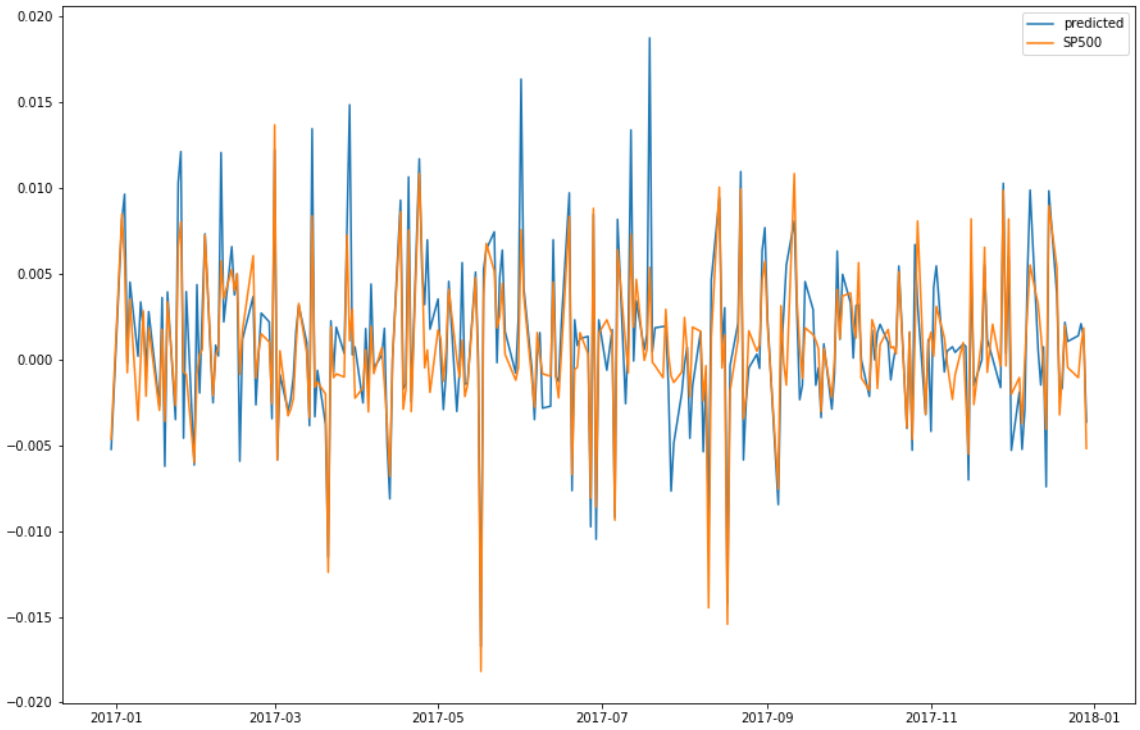
We had two functions that helped us, first, to constrain weights to be positive and to sum to 1, then, to compute the square regression loss.

So, we did our regression function, check if the weight sums to 1 and plot losses during gradient descent, if everything is ok, loss has to decrease.

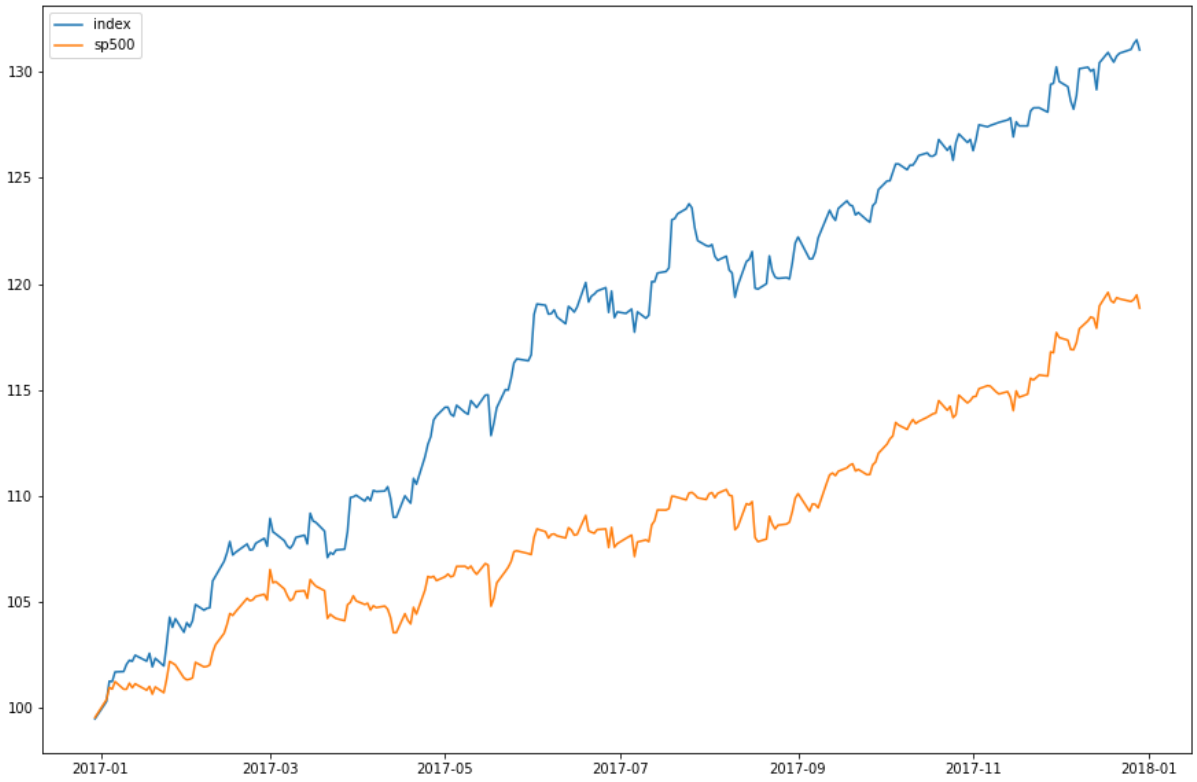


The test loss (Mean Squared Error) MSE is : **5.16920135851251e-06**

We have then plotted the predicted returns of the index due to the function that we have used for.

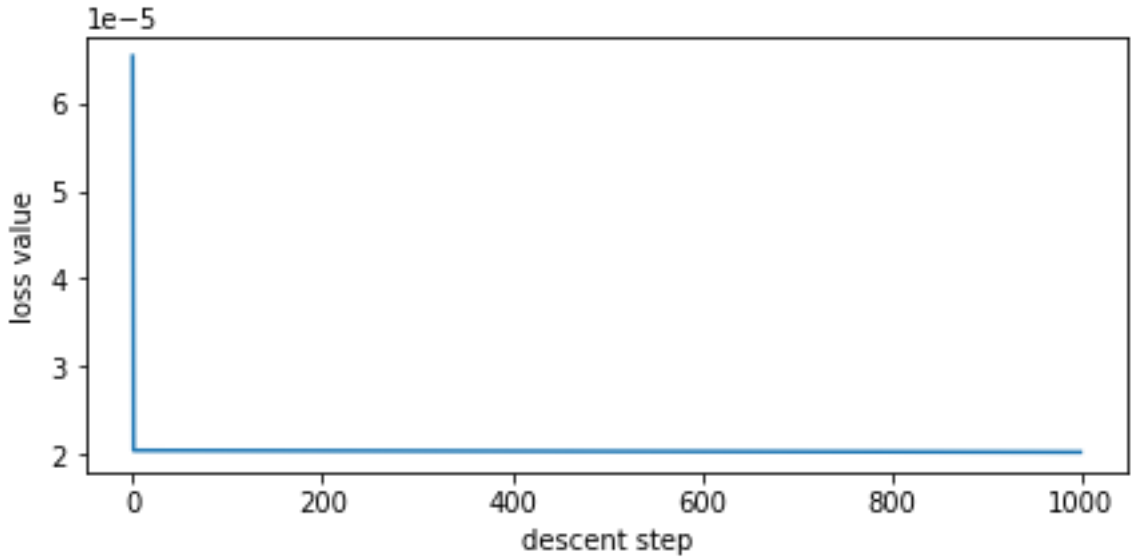


The evolution of the stock price :



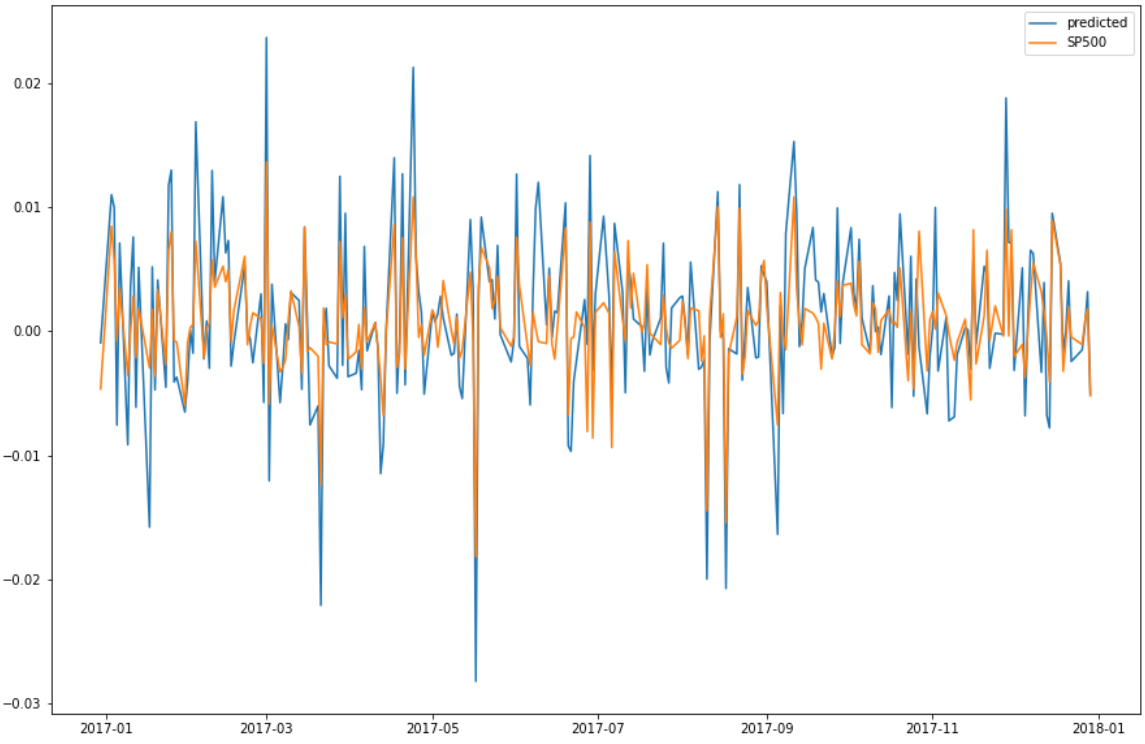
Here we see, since the start, that our prediction significantly outperforms our SP500 benchmark. We are taking more of the upside the result might change depending on the random data that we selected.

Second, we have selected the portfolio with **the 50 most correlated companies.**

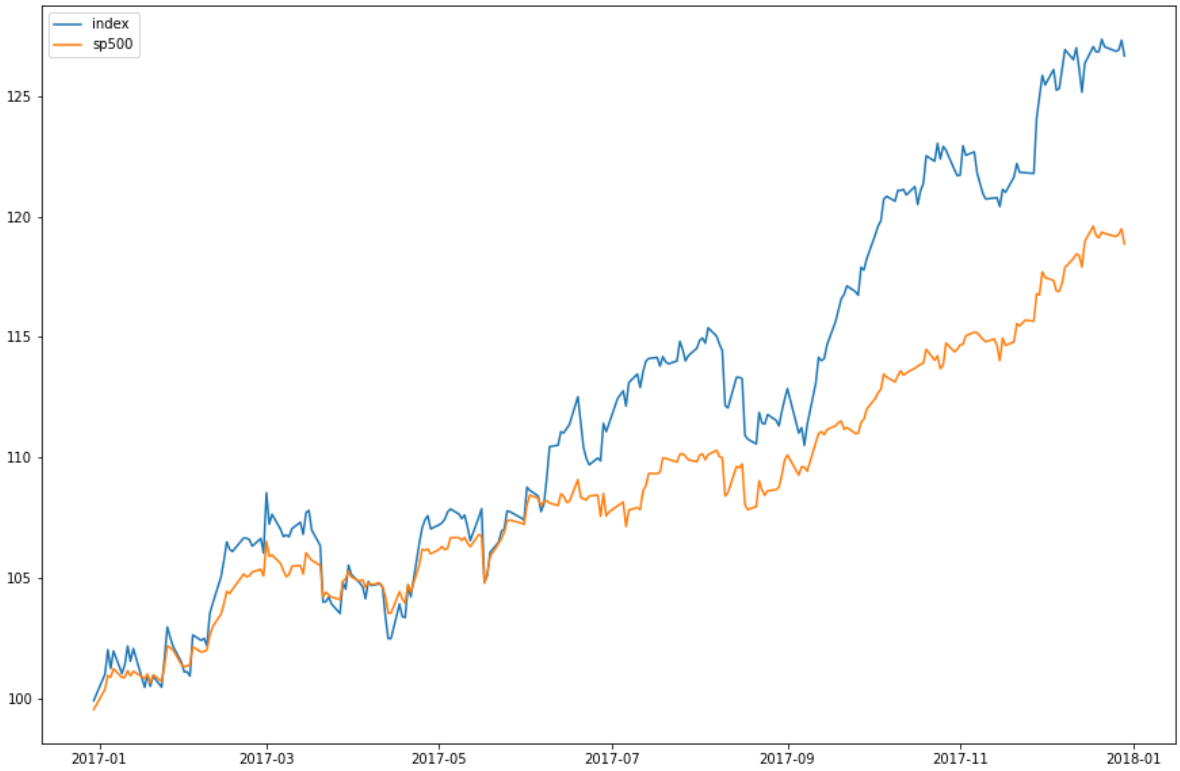


The test loss (Mean Squared Error) MSE is : **1.507839078840334e-05**

We have plotted the predicted returns of the index due to the function that we have used.

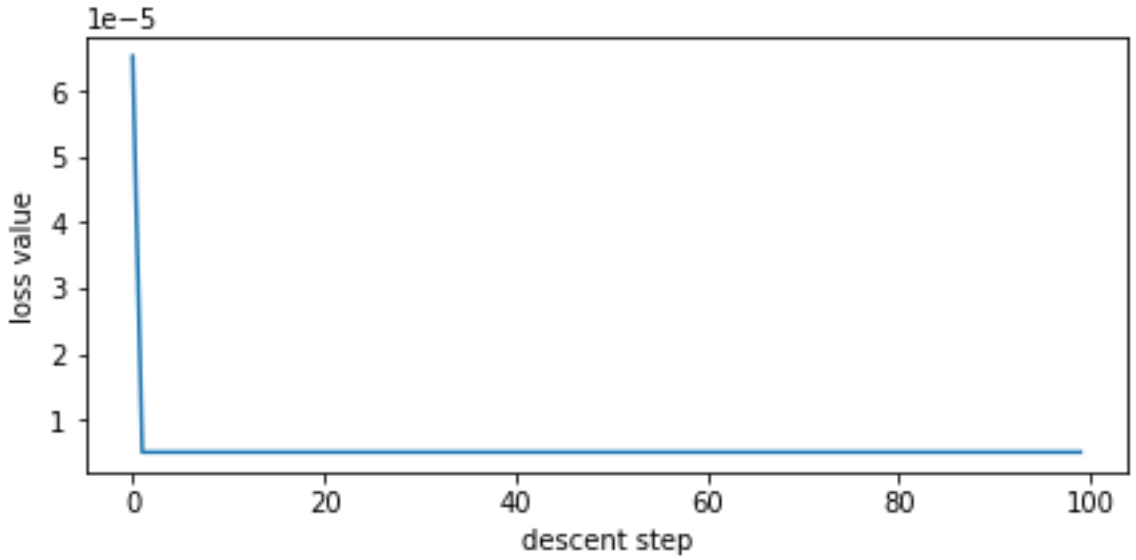


The evolution of the stock price :



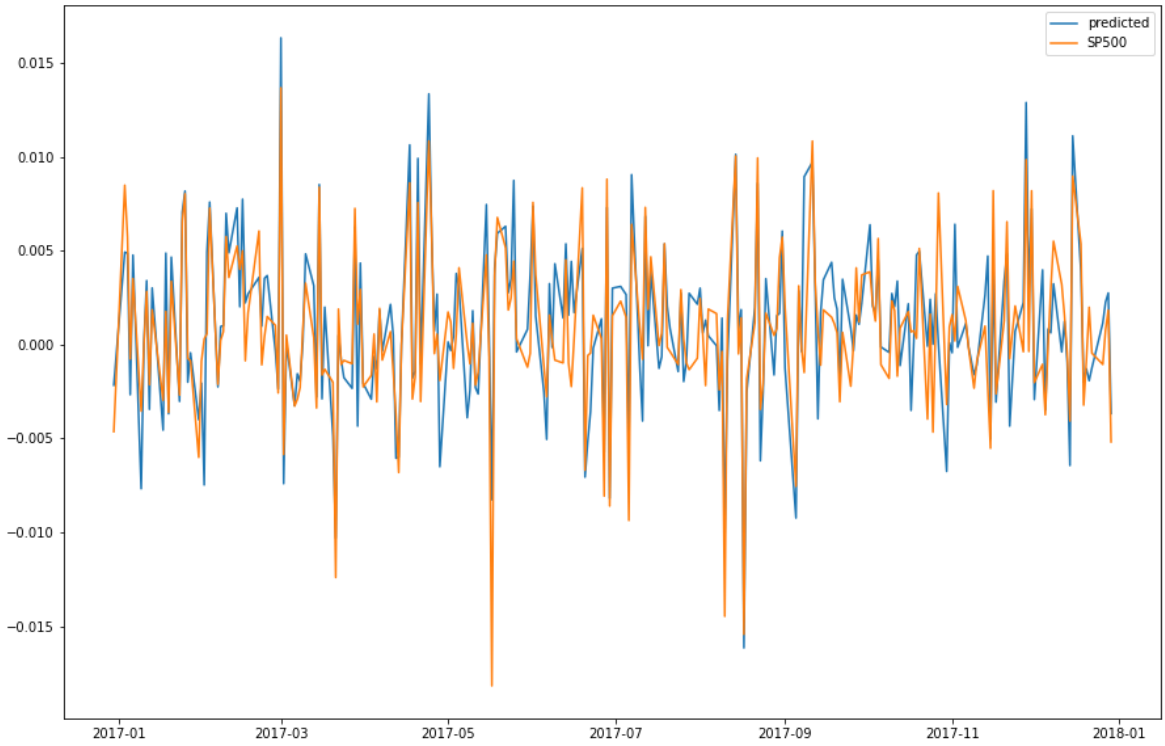
Here we see, since June 2017, that our prediction significantly outperforms our SP500 benchmark. We have taken much more of the upside.

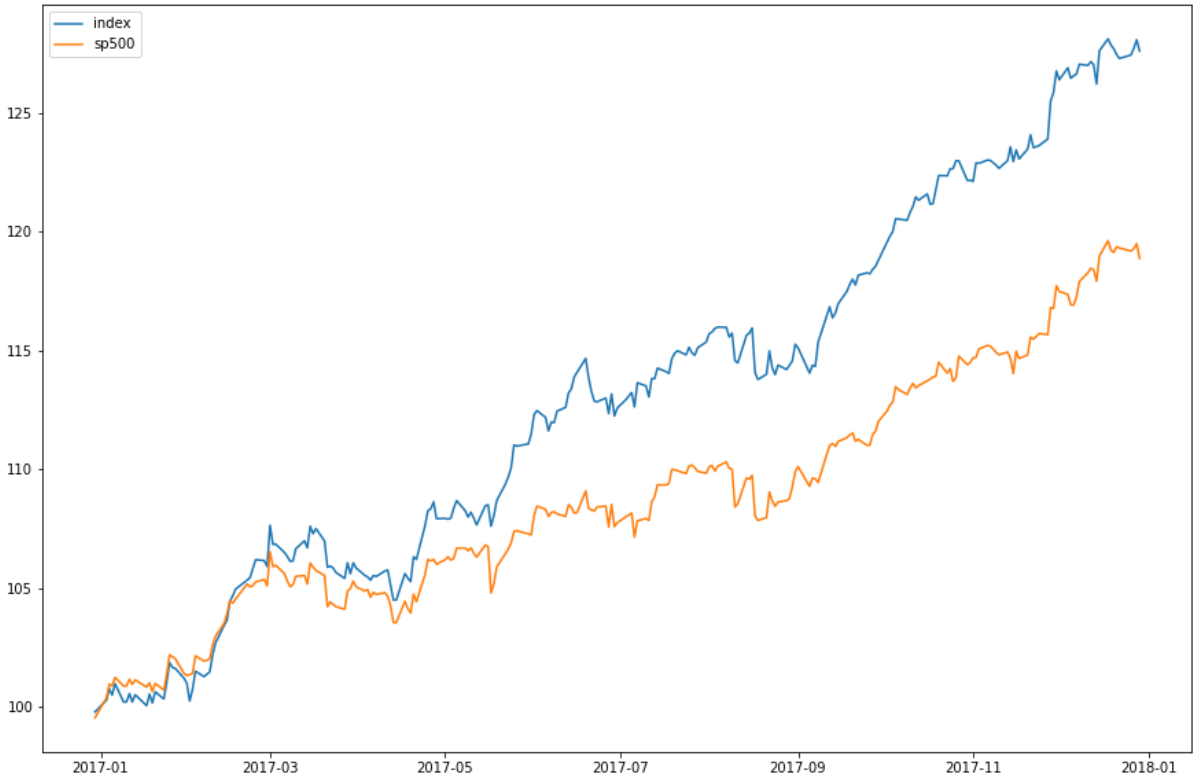
Finally, we used to do the prediction with **50 stocks using the Clustering**



The test loss (Mean Squared Error) MSE is : **6.016874976921827e-06**

We have then plotted the predicted returns of the index due to the function that we have used for.





Here we see, that from January 2017 until February the SP500 outperformed our index but since May 2017, our index significantly outperforms the SP500 benchmark. We have taken much more of the upside.

⇒ Using the **TensorFlow**, **for minimizing the square loss**, the lowest MSE if for **selecting the 50 random stocks.** However, it's not also the case since we selected the stocks randomly. We have did an extra work using **Scikit Learn** and we have these results for :

50 random stocks : Test loss is 4.46093145001214e-06

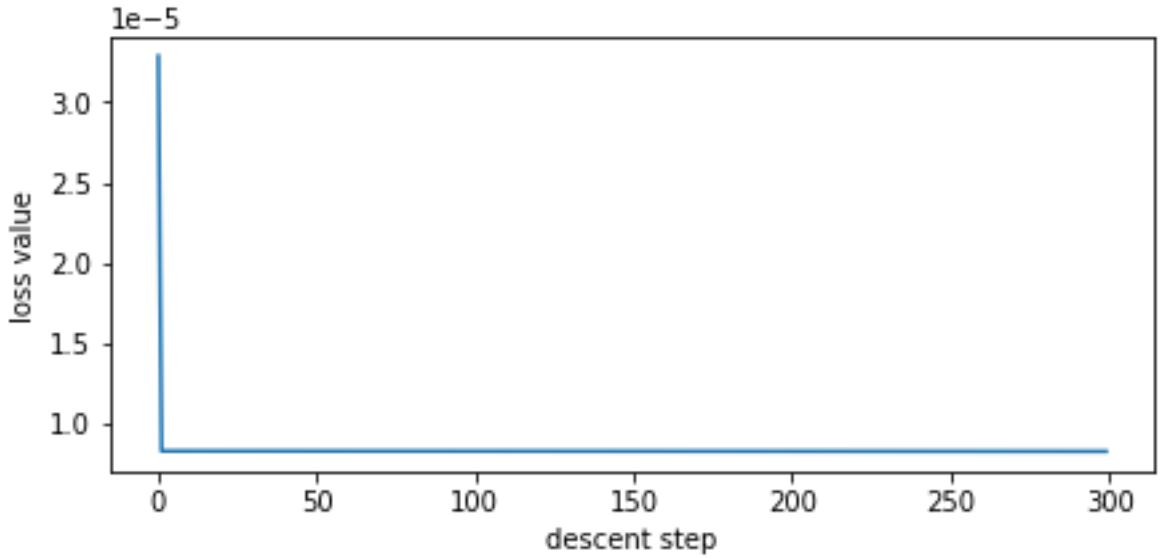
50 most correlated stocks : Test loss is 8.004121809790377e-06

Clustering : Test loss is 5.58138390260865e-06

We got better results with these methods but we find that the best MSE is for 50 random stocks. Note that, we can use this method only if we can just when you are allowed to shorten stocks (have negative weights).

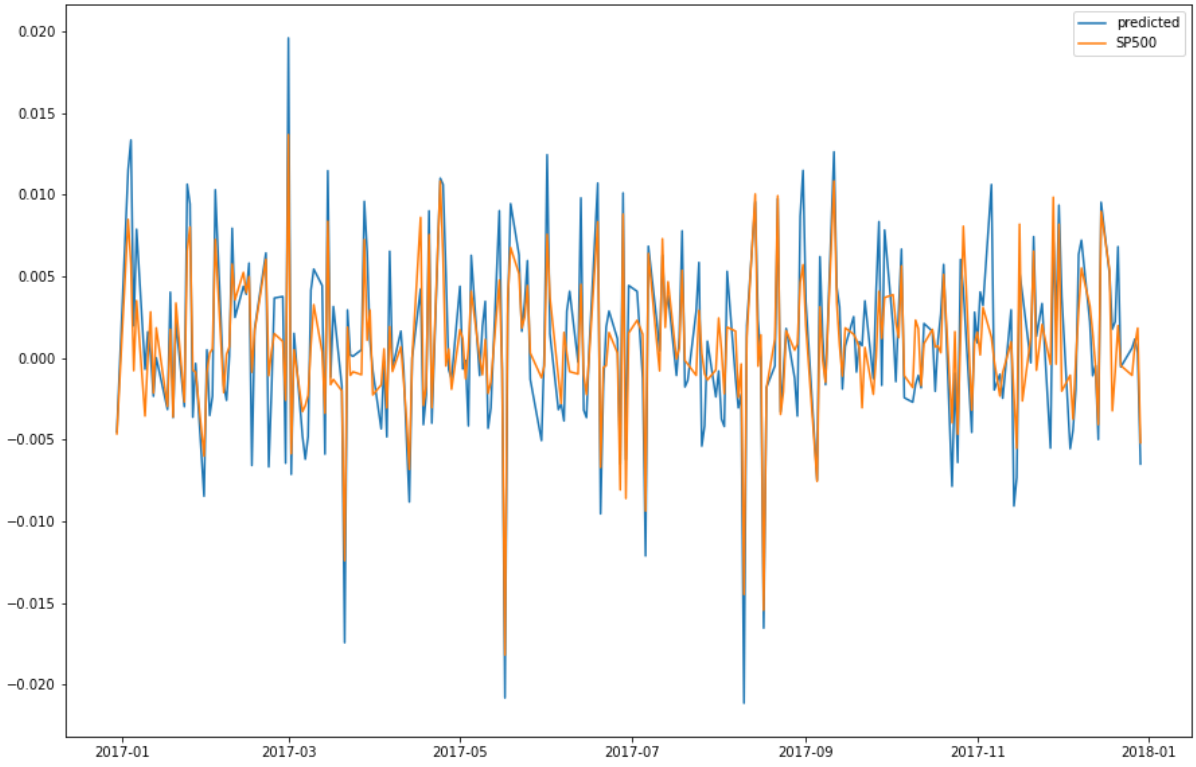
Minimizing downside risk

First, we will proceed by **selecting 50 random** stocks using the function (Random.Choice).

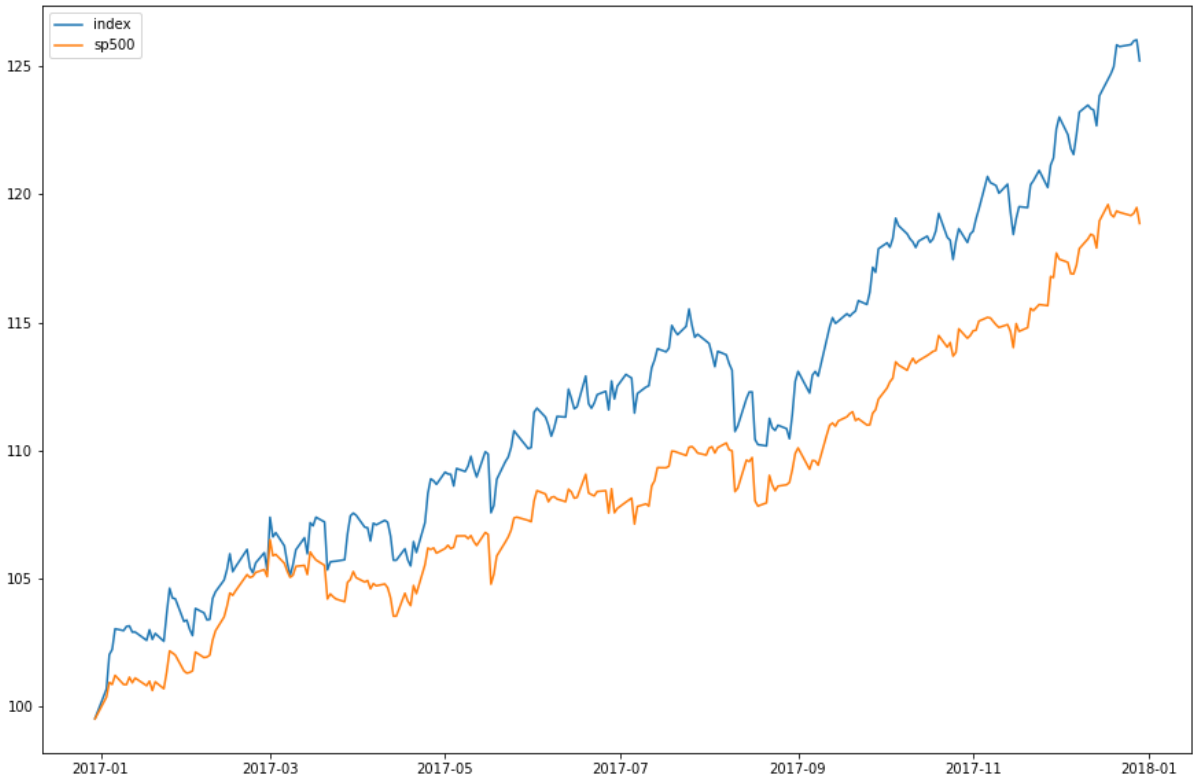


The test loss (Mean Squared Errors) MSE is **4.46093145001214e-06.**

We have plotted then the predicted returns of the index due to the function that we have used for.

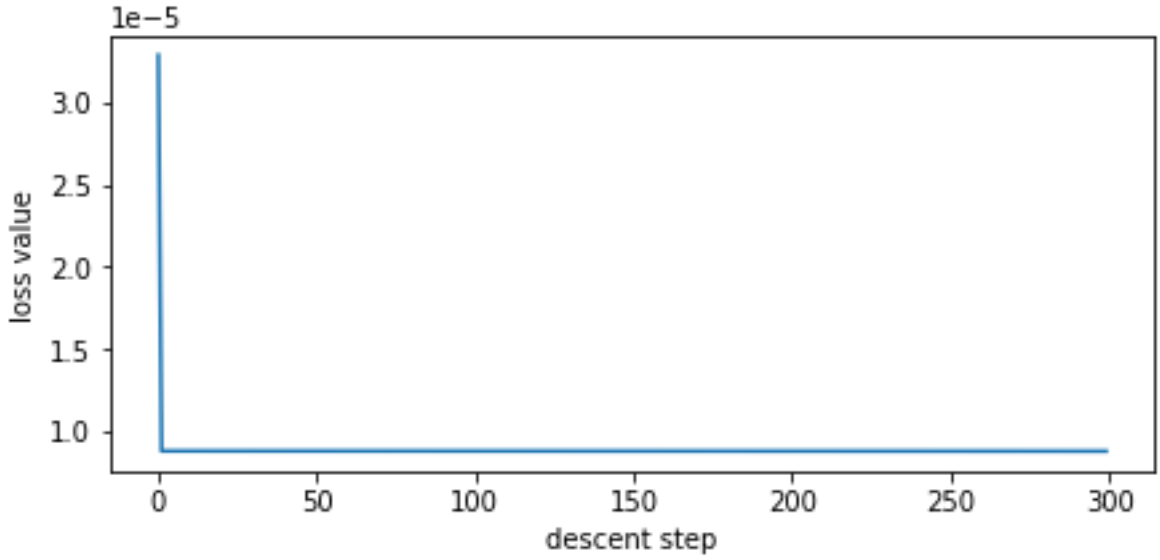


The evolution of the stock price :



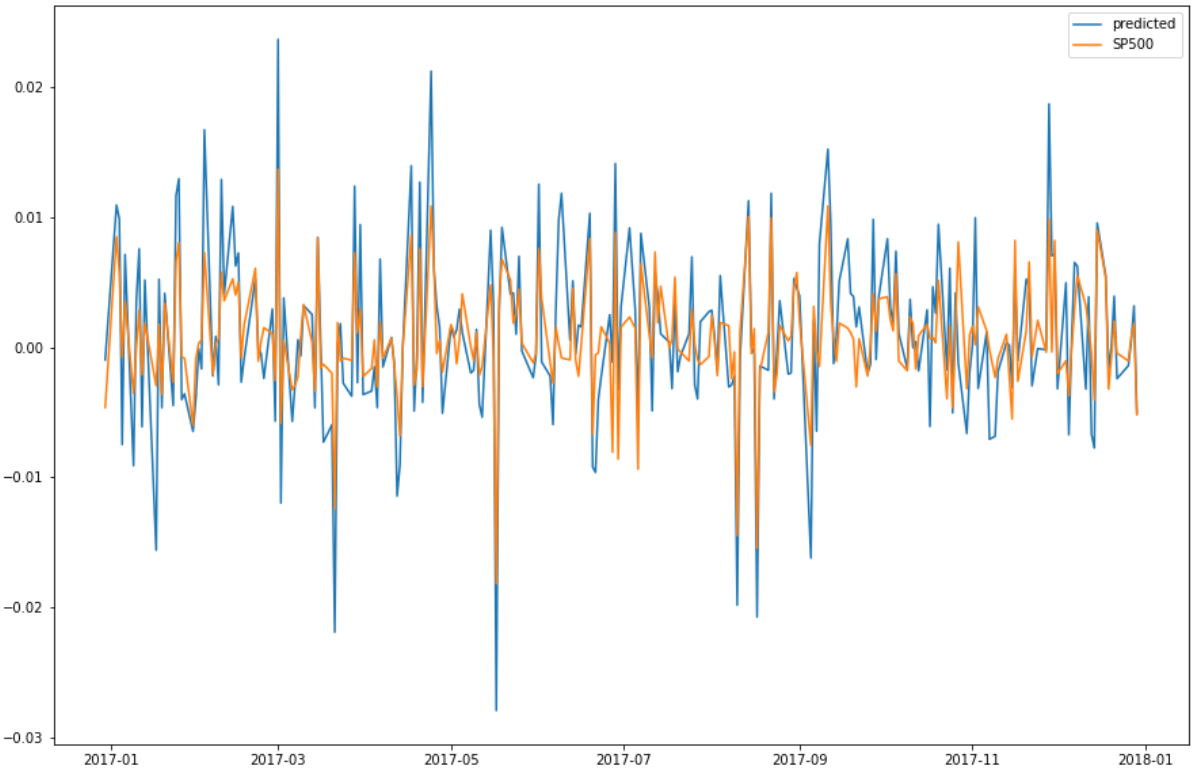
Here we see, since the start, that our index significantly outperforms the SP500 benchmark. That's the objective of the minimization of the downside risk. We are taking more of the upside and trying to minimize the downside.

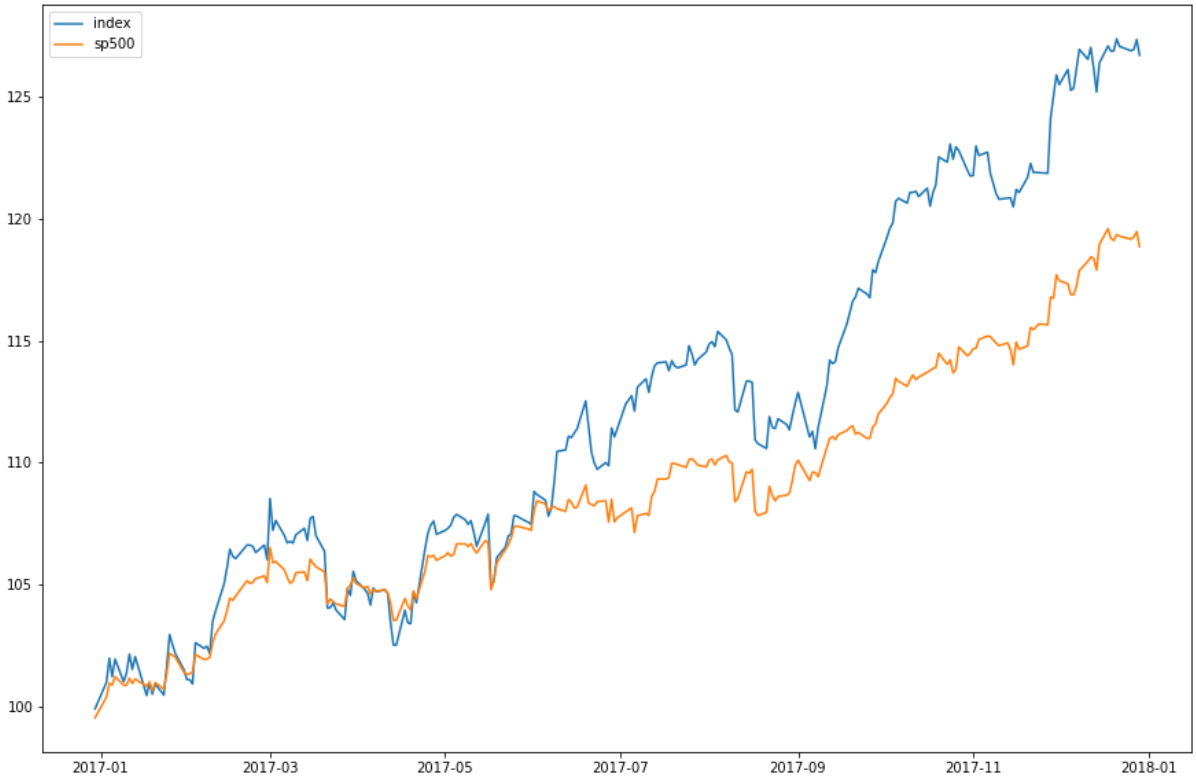
Second, we have selected the portfolio with **the 50 most correlated companies**



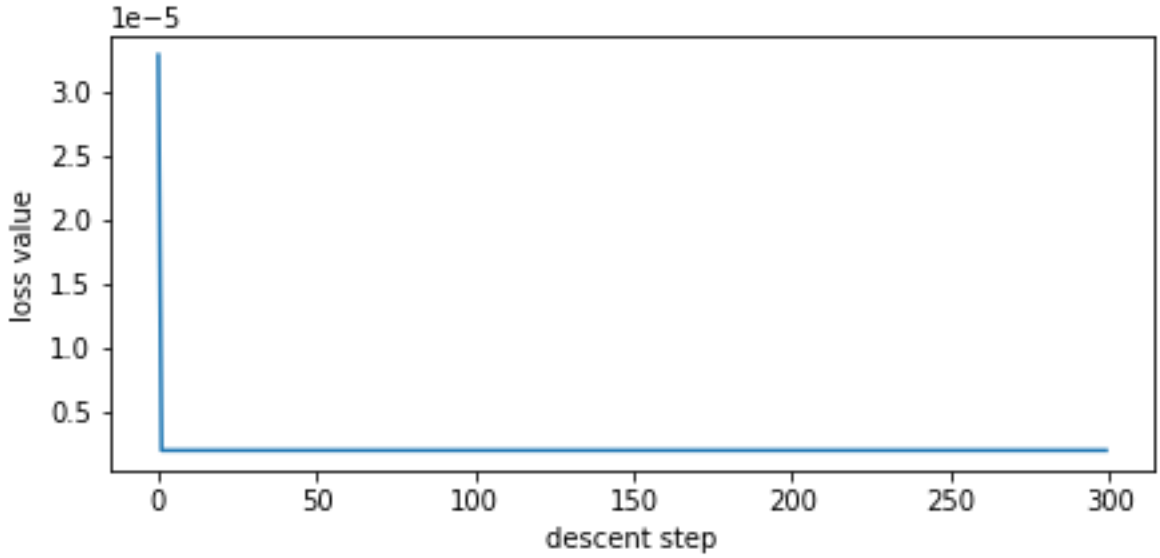
The test loss MSE is : **8.004121809790377e-06**

We have then plotted the predicted returns of the index due to the function that we have used for.



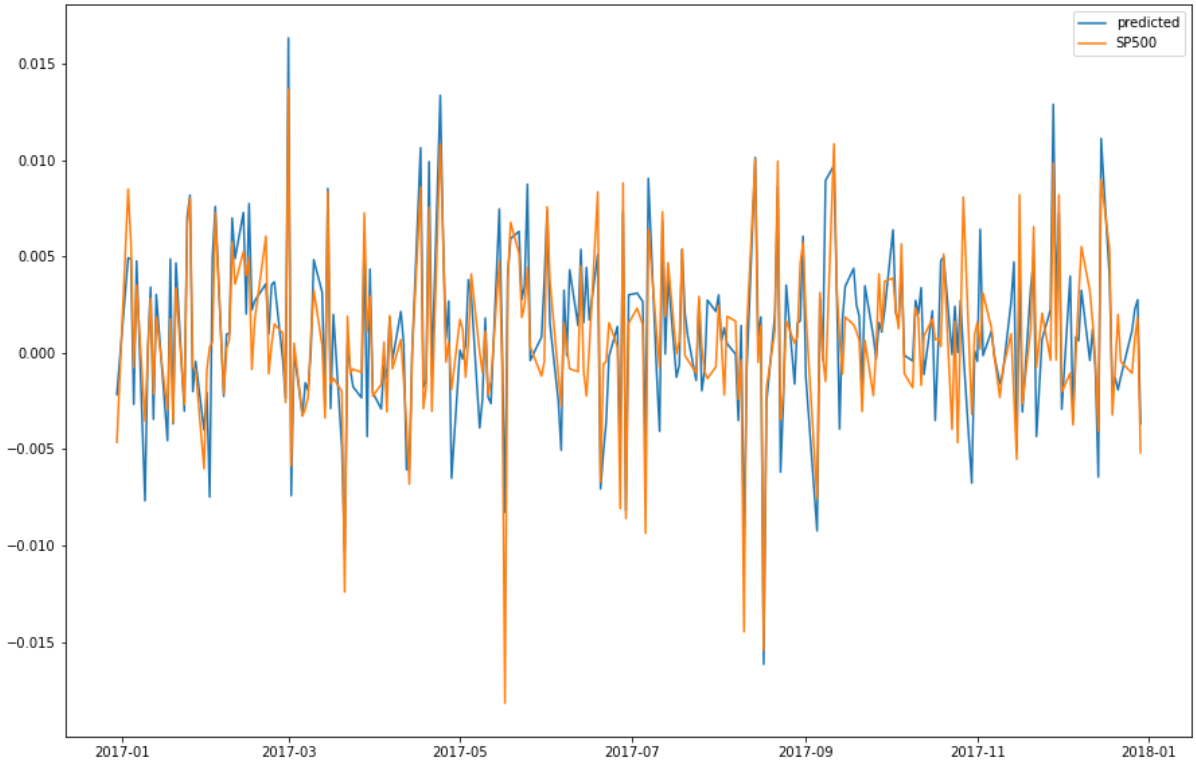


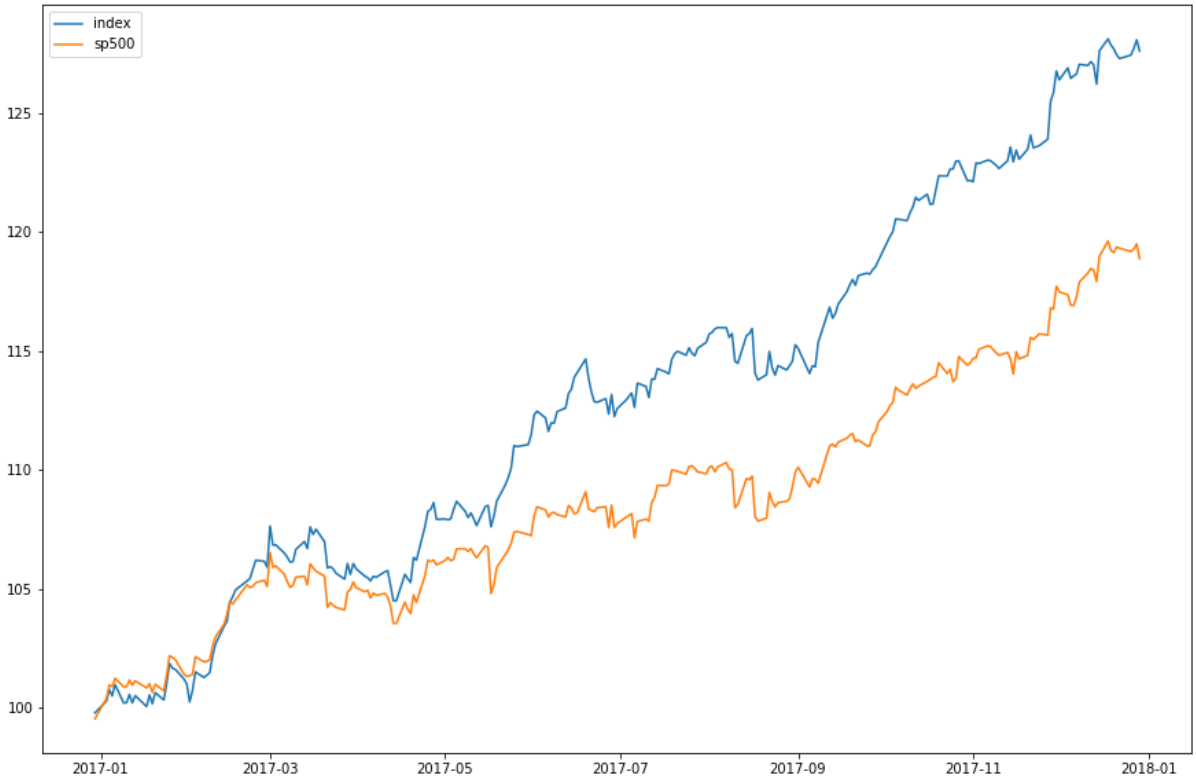
Here we see, since the start, that our prediction significantly outperforms our SP500 benchmark. We are taking more of the upside. In April 2017, we have taken a much more important downside as the sp500. Then in June 2017 before outperforming till the end. Finally, we used to do the prediction with **50 stocks using the Clustering**



The test loss MSE is : **5.58138390260865e-06**

We have plotted then the predicted returns of the index due to the function that we have used for.





Here we see, since the start, our index follows the sp500 until April 2017 when our prediction significantly outperforms our SP500 benchmark. We are taking more of the upside and trying to minimize the downside.

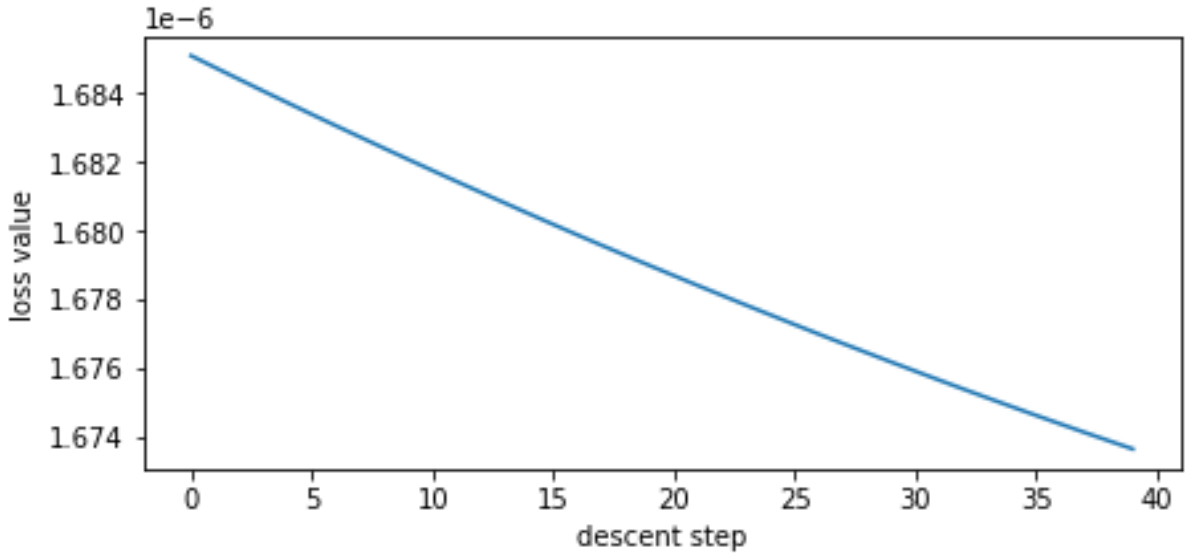
⇒ Using the **TensorFlow**, **for minimizing the downside risk**, the lowest MSE if for **selecting the 50 random stocks.** However, it's not also the case since we selected the stocks randomly.

Predicting sparse portfolio weights

After doing the direct regression we will do our work by first predicting sparse portfolio weights, so updating our portfolio each 5 days.

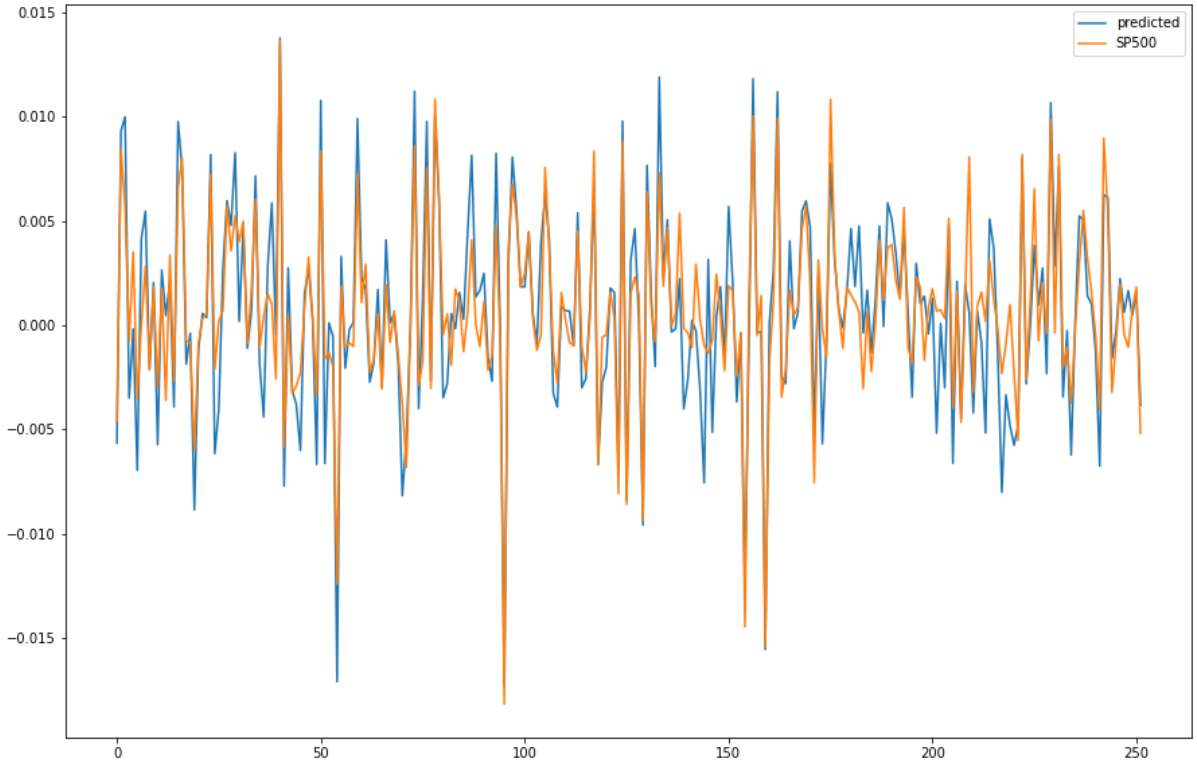
1. **50 random stocks**

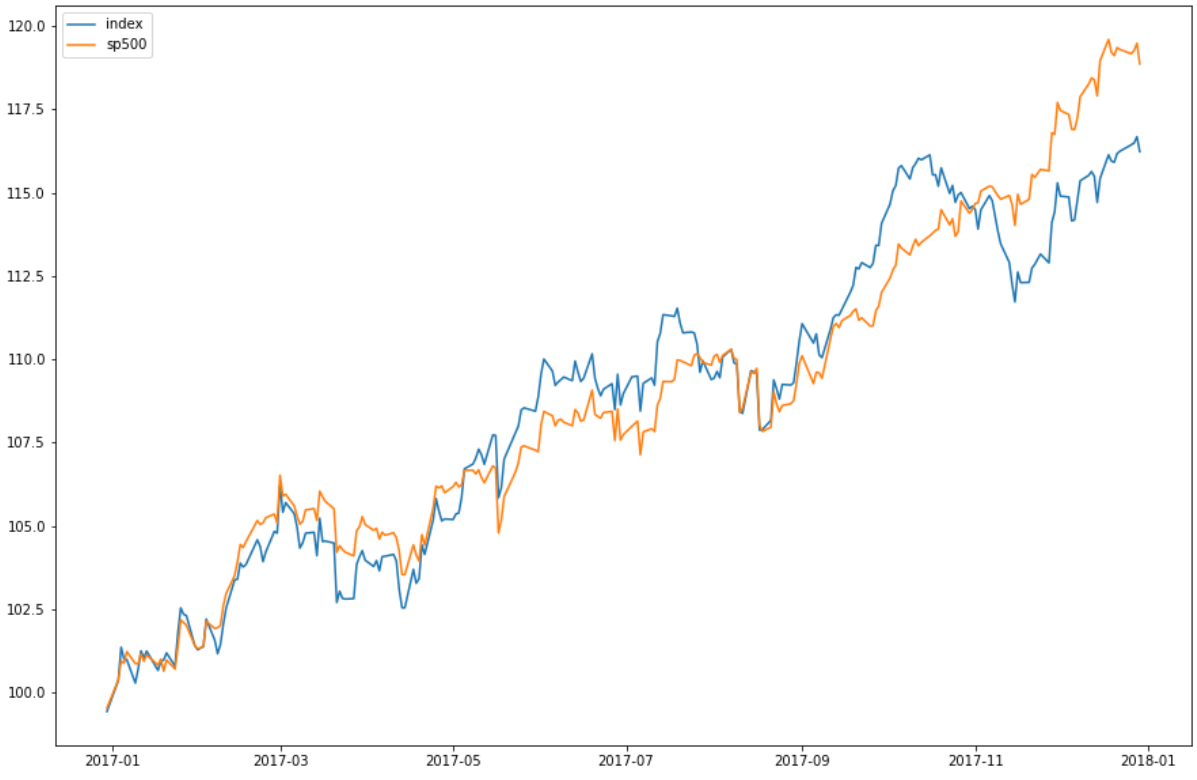
We had the final weights and checked that regression weights sum to 1.
We plot losses during gradient descent :



Here, we applied the loop for a range of 40 otherwise it will take 5 minutes The test loss (MSE) is : **4.778515e-06**

We have then plotted the predicted returns of the index due to the function that we have used for.

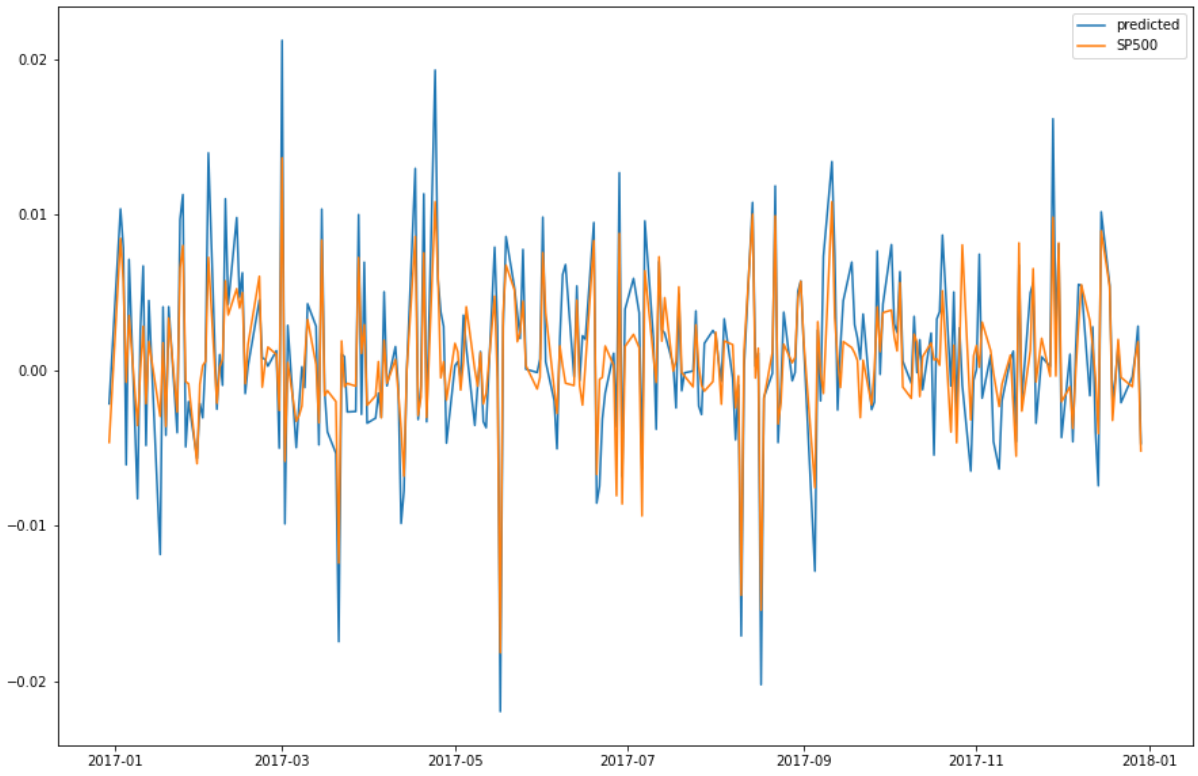




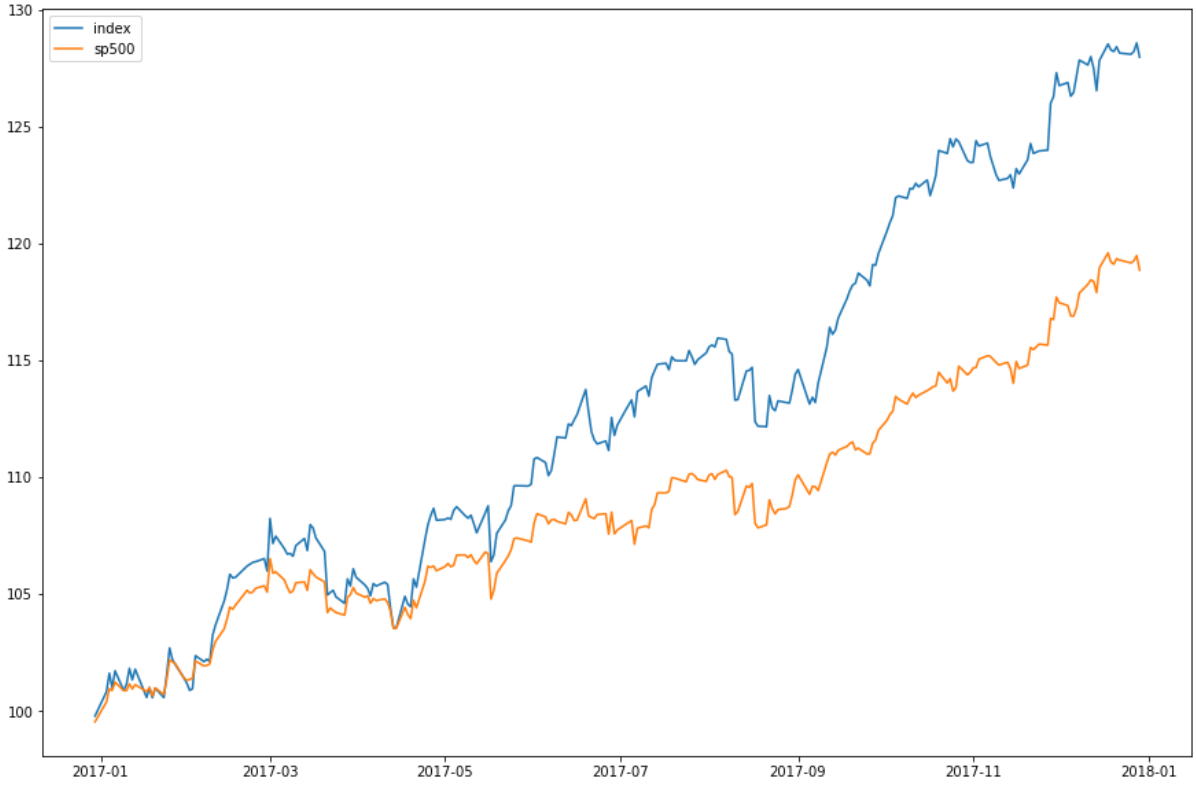
In this case, we have seen that for the start, the two index were aligned and then the sp500 sur performed our index this divergence is normal since we should call our model each 5 days to readjust the weights and to better tracking of the SP500.

2. 50 Most correlated stocks

The test loss (MSE) is : **8.004121809790377e-06**



Here, we see that our index surperformed since the start and had a better performance.

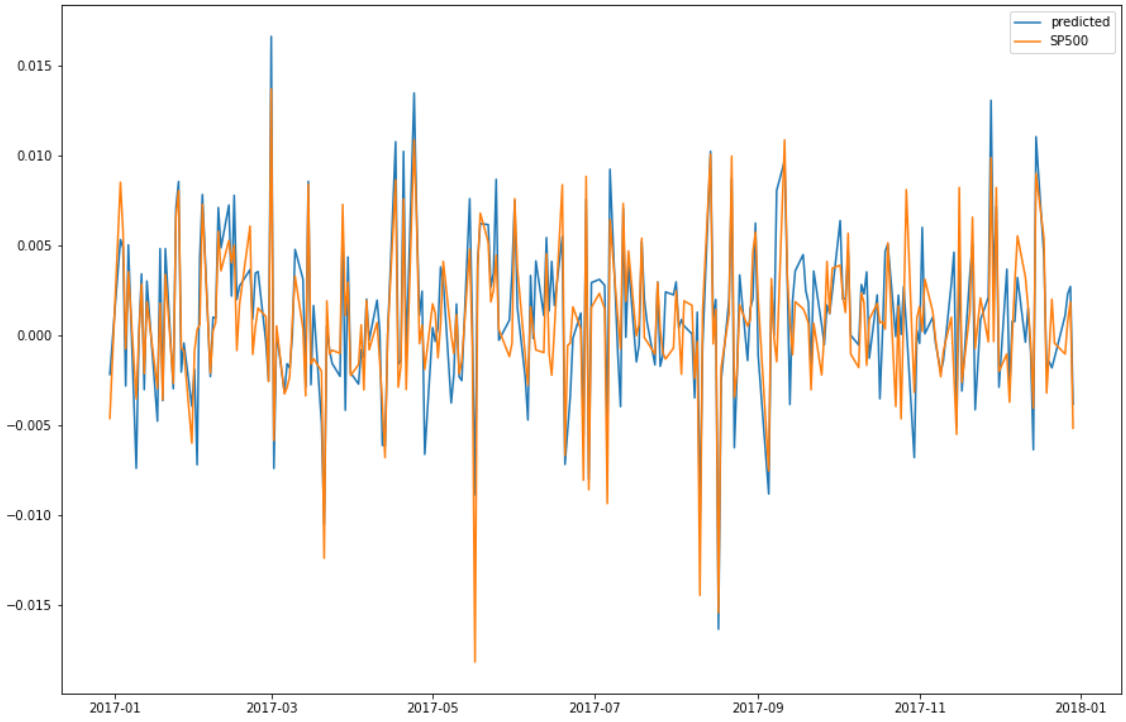


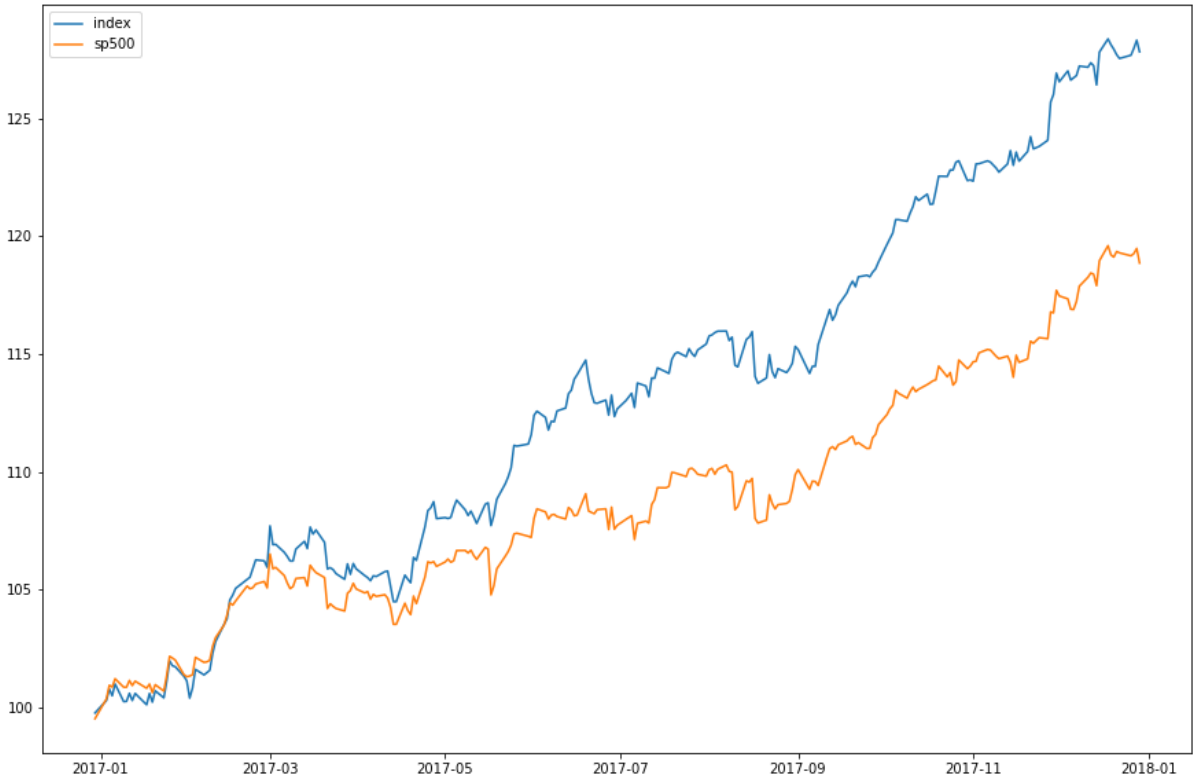
In this case, we have seen that for the start, the two index were aligned and then the sp500, then it diverges, this divergence is normal since we should call our model each 5 days to readjust the weights and to better tracking of the SP500.

3. 50 stocks using clustering

The test loss (MSE) is : **5.58138390260865e-06**

We have plotted then the predicted returns of the index due to the function that we have used for.





In this case, we have seen that for the start, the two indexes were aligned, and then the sp500, then it diverges, this divergence is normal since we should call our model every 5 days to readjust the weights and to better track the SP500.

Conclusion :

To conclude, the more we advance with time, the more the weights are not adapted to track the index, we move away and that pushes us to outperform. So, we have to update the model every 5 days to adjust the weights and have an ETF that replicates the performance of the SP500. The project was not easy to develop and at the end, however we managed to do what was asked and we find it very interesting.

Below is one of the best results that we get that track the SP500 perfectly we get it using the rebalanced weights and 50 randomly selected stocks.