MarketRegimeDetector (MRDetect)

Detecting Stock Market regime (Bear / Bull Market)

using HMM and RNN.

A picture containing silhouette

Description automatically generated

By Emmanuel Berdugo, Asi Sheratzky, Yonatan Mirsky and Roy Toledano

Project goals:

The main objective of the project is to design a Trading Strategy that can beat the Market! It seems quite ambitious, but we believe it is achievable. We know that on the very long run (decades), investing in the Stock Market will yield higher return that any other investment vehicle. But on some shorter periods, Stock Markets can also perform really bad!

We want to build a tool that will help us to time the market.

Data Extraction and exploration:

This phase was quite simple since we took our data from Yahoo Finance package.

We decided to analyze the SPY ETF price, since it reflects the price of the S&P 500 index, which is widely considered as benchmark for determining the state of the overall economy, and it is tradable as is on the stock market, so it could enable us to have a Trading Strategy ready to be used.

Chart, line chart

Description automatically generated

We added some features to the data, mainly the Moving Average for several periods (10, 30, 60 and 120 days) and the historical (realized) volatility for the same periods. The formula for the realized volatility is as follows:

Text

Description automatically generated

We intend to use several “observations” to learn our “Market Regime States” with our HMM, and the volatility is some important observed feature. We know that bear markets are correlated with high volatility, so we would like to help our model to predict the relevant market regime by providing it with this information.

We decided to discard the first 10 years of the data, because we suspect the market behavior is a changing business, and it might be variable with time. Moreover, we kept 2 years of the most recent data for the test set, to check the validity of our model. So we have roughly 16 years of training set.

Model Baseline Building:

We chose eventually to use the Return and the 10-days Historical Volatility, since it gave the best results.

As a metric, we chose the Z-score of the daily returns. The calculation is as follows:

Z-score = mean\_of\_returns / (std\_of\_returns / sqrt(nb\_of\_days))

This is a quite frequent profitability measure in trading since we want to maximize the return of the strategy but to minimize the Standard deviation. A strategy very profitable but with a high standard deviation might be profitable only by chance… A Z-score of 3 or higher is a proof the strategy is really valuable.

Graphical user interface

Description automatically generated

We trained our model to detect between 3 hidden states, and we will now try to have a look at that he did…

Chart, line chart

Description automatically generated

It seems the predictions on the train set are quite good, in this case we see that the state “1” seems to be the loser\_state. The “0” state seems to be the winner\_state, while the “2” state seems winning also, but with a higher volatility. Let’s check.

Graphical user interface, text, application, Word

Description automatically generated

The results are as we expected.

Let’s have a look at our trading strategy now. It will be very simple: when we are in the loser\_state, we will be out of the market (selling if we were long – doing nothing if we had no position) and we will be long in the other cases (buying if we had no position – doing nothing if we were already long). Then we compare the graph of the value of our position (rebased at the same index).

Chart

Description automatically generated

We see here a clear advantage for the HMM Regime detection based strategy! We got out of the market during the bear market phase of the Subprime crisis, and got back to the long position at time. Overall we achieved roughly 70% higher return on the whole period, and, even more important, got a Z-score almost twice as good! This means our strategy seems valuable!

But we need now to check the results on the test set…

Graphical user interface

Description automatically generated with medium confidence

Chart, line chart, scatter chart

Description automatically generated

It looks good! Let’s have a look at our trading strategy…

Chart

Description automatically generated

Well… We need to say this is far above our expectations! It’s hard to believe these predictions were done step by step by the model! The model managed to avoid all the bear periods! Overall it achieves a performance 54% higher (77.63% instead of 50.45%), with a very high Z-score (3.838 instead of 1.328)! No doubt that even these results are particularly valuable for trading! But we would like to achieve better…

Next Steps:

We want to be sure of the validity of our model, so we will apply it to other indices to check if it works

We want to try other methods, based on Deep-Learning (RNN, LSTM, GRU, Transformers?) to try to achieve better results