PAIRS TRADING : A MACHINE LEARNING APPROACH

Waldyr Faustini

Stevens Institute of Technology, NJ
Hoboken, USA
wmiguelf@stevens.edu

Sandeep Ranjan

Stevens Institute of Technology, NJ
Hoboken, USA
sranjan4@stevens.edu

Final report presented by the end of the course FE670 - Algorithmic Trading Strategies offered in the Fall 2022, as part of the final project.

ABSTRACT

The main idea for this project was to investigate in more details about the Pairs Trading Strategy, a common quantitative trading strategy used in hedge funds.

Pairs trading is an approach that takes advantage of the mis-pricing between two (or more) co-moving assets, by taking a long position in one (many) and shorting the other(s), betting that the relationship will hold and that prices will converge back to an equilibrium level.

In this case we used two different methods to predict the spreads between two stocks and then compared the two methods.

The first method was a Time-Series model and very well known, **ARIMA Model - Auto-Regressive Integrated Moving Average**.

The second method was a Machine Learning approach to study the spreads amongst some stocks in the S&P50. The algorithm we choose here was the **Random Forest**, accordingly to what we studied was the best approach for this kind of situation.

However, is not a simple task to really identify if one asset is truly overvalued/undervalued, especially because depends of a bunch of parameters to determine that. Based in the spreads results the final goal is to predict the future spreads for the selected pair of stocks.

1. OBJECTIVES

There are 3 main steps for building a Pair trading strategy:

1) Pairs Selection: this is a fundamental step, we tried to find stocks which historically move together and have some sort of long-run correlation. In order to find that we ran the **Cointegration test** to find some possible pairs of trading. After having a pre-selection of possible pairs we ran the **Johansen test** and **ADF test** to confirm we are looking for the best pairs available.

2) Spread Forecast: we had to model the spread between the stocks prices based on the historical data. And then considering the model we used that to predict the future spreads between the pairs. In order to judge and analyze our models we used 2 different models:

- **ARIMA**: an ARIMA model is after all very similar to a multivariable regression.

- **Random Forest**: is a supervised machine learning algorithm

3) Trading Rules: based in the trading signals generated by our model we defined some trading rules to get in and get out of some pair trades.

2. DATASET AND PAIRS SELECTION

In this project, we considered all the stocks in the **S&P 500 from 2018 to 2021**, one of the considerations to select S&P 500 stocks was the good liquidity of the stocks in the S&P 500 Index. We decided to get the data from 2018-2021 in order to have data pre-Covid, during the worst part of the pandemic situation (2020) and then 2021 when the market was a bit more stable.

In this project we are not considering intraday prices for each stock, we are just working with the **Adj Close Price** to define the spreads of the pairs. Since we are dealing with pairs of stocks, we are more interested in the spreads of the selected pairs.

We first ran the cointegration test for all the S&P500 stocks and then we had 2 stocks selected for our pair trading strategy: **ETR vs SRE**, both stocks are in the Energy sector. This pair of trading represents the lower p-value considering all the stocks in the S&P500.

Figure 1: *This chart is showing that SRE and ETR have some sort of long-run correlation. It's a spread between the price curves*

After that in order to have another pairs to test our strategy we decided to select stocks sharing the same characteristics and risk exposures. We also selected the Financial sector and then we ran the cointegration test just for stocks among the Financial Sector. In this case we end up with another pair of stock:

- Financial Sector: RE vs GL

So for this project we identified 2 different pairs: **ETR** vs **SRE** and **RE** vs **GL**.

3. METHODOLOGY AND FEATURES

3.1. Features

After the pair selection we had to decide how to define our features and trading rules.

Since the main goal for this model was to predict future spreads, we basically trained our model using the spreads between the pairs of trading.

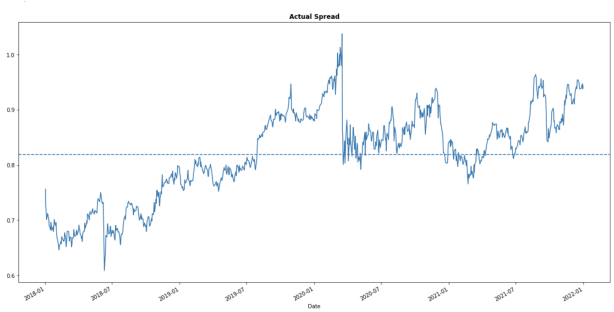

Figure 2: *Spread between SRE and ETR. In this chart we are considering all the data collected, without any splits*

We decided to split the data in 2 groups: **75% to train the model** and **25% to test our model**. So we used this divison for the **ARIMA method** and also for the **Random Forest model**.

3.2. ARIMA Model

An ARIMA model is after all very similar to a multivariable regression. ARIMA uses maximum likelihood estimation (MLE).

ARIMA (p,d,q):

$y'_t = AR(p) + MA(q)$

$y'_t = \phi_0 + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \epsilon_t + \theta_1 \epsilon_{t-1} + ... + \theta_q \epsilon_{t-q}$

ARIMA was implemented in python using **statsmodels** library

3.3. Random Forest

Random forest is a **Supervised Learning Algorithm**, this method could be used for classification or regression. Since we are working with prediction, we are using the **regression**

4. TRADING STRATEGY

We used the spread information to create a simple trading strategy to determine when to buy or sell the stocks.

In particular, we look at the value of the normalized spread, using the **MinMaxScaler** function, so all the spreads are **[0,1]**.

We looked for opportunities when the spread significantly deviates from the mean of the spread, in this case we basically calculated the **60dma - 60 day moving average** as the **mean**.

To determine how far of the mean are the prices we calculated the **Z-Score**, the Z-Score it's scaled between -1 and +1. Basically every time deviates from -1/+1 will generate a Sell/Buy signal for the Pair Trading Strategy

The strategy is basically to take a short position in the overvalued asset and a long position in the undervalued asset based on the Z-Score.

As soon as the spread converges back to its mean, we unwind both the positions and estimate the profit.

5. RESULTS AND DISCUSSIONS

Due the fact we don't have much space to report all the results, we decided to show the results related to **ETR vs SRE** pair trading strategy.

Our first approach was trying to train the models using **technical indicators**, but seems like the model was overfitting the data. We will discuss more in the next section.

Below we showed the difference of the methods. The **Random Forest method worked better without the technical indicators** as we can see in the charts and also in the **Test Score**.

6. SPREAD COMPARISON ETR VS SRE : USING TECHNICAL INDICATORS AS FEATURES

In this chart we are basically comparing the true value of the spreads of between ETR vs SRE, the spread predicted by the ARIMA model and the spread by the Random Forest Algorithm.

In this case case we tried to train the Random Forest Algorithm using Technical Indicators.

We calculated the following technical indicators : **RSI, MFI, AD, VP, BB, ATR, ADX, EMA and MACD.** and used that as the features for the prediction model.

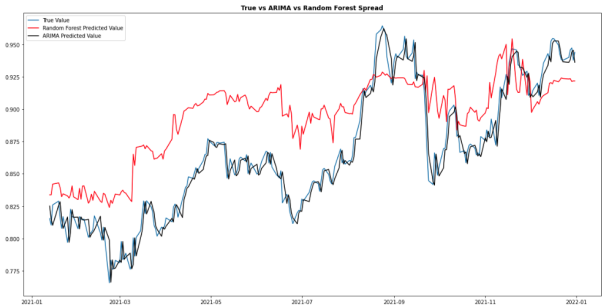

Figure 3: *Spread Comparison - with technical indicators*

As we can note from the chart above, the red line (Random Forest Method) is very disconnected from the true value of the spreads. So considering we used data pre-covid to train the model, seems like the excessive number of features and the fact were technical indicators overfitted the model and then when tried to predict the real values wasn't that great.

```
Mean Absolute Error: 0.0362
Mean Squared Error: 0.0017
Root Mean Squared Error: 0.0407
(R^2) Score: 0.3133
Train Score : 99.80% and Test Score : 31.33% using Random Tree Regressor.
Accuracy: 93.6 %.
```

Figure 4: *MSE - with technical indicators*

From the results above is a bit more clear how the model was overfitting before, because we had the Test Score 99.80% in the training set and only **31.33% in the test set**.

7. SPREAD COMPARISON ETR VS SRE : ONLY USING SPREADS AS FEATURES

We knew the Random Forest Algorithm was very well known for the capacity of prediction with a decent accuracy.

However we didn't have much success when we tried to use the technical indicators as the features.

In this case we start to think how to tweak the model in order to have better results, so we decided to ignore the technical indicators as features and used the actual spread to train the model.

Figure 5: *Spread Comparison - without technical indicators*

As we can note from the chart above, the red line (Random Forest Method) is not a perfect prediction (as we should expect) for the true value, but the results are much better when compared with the prediction using the technical indicators as features.

So considering the results, we could assume final curve was pretty accurate considering our expectations.

```
Mean Absolute Error: 0.0101
Mean Squared Error: 0.0002
Root Mean Squared Error: 0.0135
(R^2) Score: 0.9248
Train Score : 99.78% and Test Score : 92.48% using Random Tree Regressor.
Accuracy: 93.77 %.
```

Figure 6: *MSE - without technical indicators*

If you take a careful look in the table above we can see how the results are much better considering the Random Forest Model being trained by the spreads.

Now we have a Test Score of **92.48%**, much more accurate than the **31.33%** obtained using the technical indicators as features.

8. Z-SCORE AND TRADING SIGNALS

We basically calculated the 60dma - 60 day moving average as the mean and then calculated the Z-score based on this.

So the **main calculation for this report was the Z-Score.**

Starting from the Z-score we can understand how much the spreads differs from the standard deviation. The mean in this case was every time deviates from -1/+1 will generate a Sell/Buy signal for the Pair Trading Strategy.

Accordingly to the chart below, every time we see the curve crossing the red line or the green line will generate a trading signal.

If the curve (blue line) is crossing the red line (Z-Score = +1) means we the spread is overvalued and we should sell the spread while the curve is above the red line.

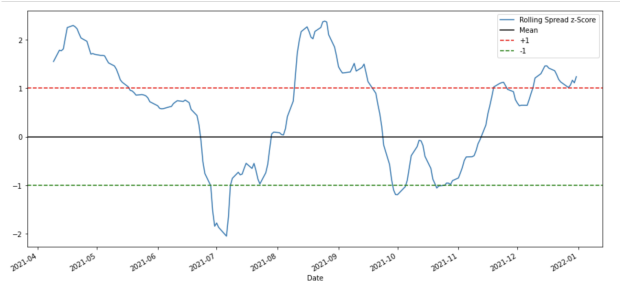

Figure 7: *Z-Score chart for the spreads between SRE and ETR*

In the other hand, if the curve (blue line) is crossing the green line (Z-Score = -1) means we the spread is undervalued and we should buy the spread while the curve is below the green line.

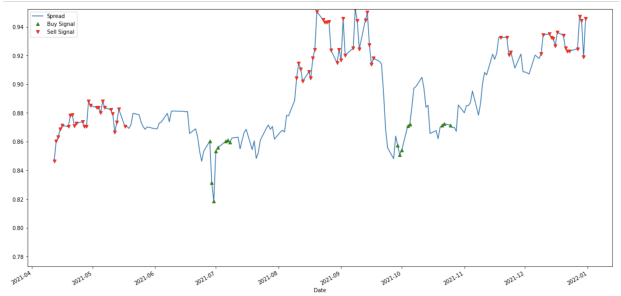

Figure 8: *Spreads - Trading Signals*

Above we can see the signals being generated in the spreads chart. For each red dot we should sell the spread and for each green dot we should buy the spread. If we look the Z-Score chart and this chart above with the trading signals we could see the correlation between the two charts.

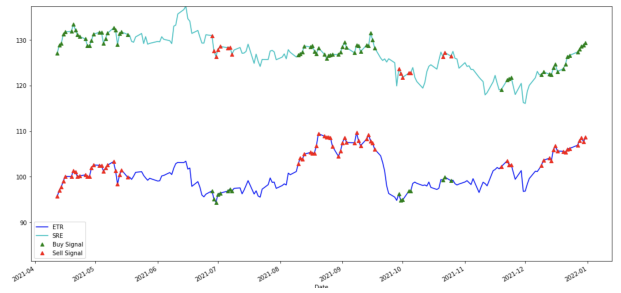

Figure 9: *Trading Signals for each stock in the pair trading, in this case the ETR and SRE*

Above we can see the signals generated for each stock of the pair trading. The focus is the spread value, in this case sometimes you are buying SRE and selling ETR and in other cases doing the opposite.

In the chart above is pretty clear how each stock is impacting the spread value in this pair trading.

9. TRADING RESULTS

Basically in our implementation, the trading strategy should happen in multiple days. Firstly we assume we have no PnL and no position at beginning.

The first question in the trading strategy it's when should we open a position in the spreads.

For each day the closing price (in this project we just worked with closing prices) it's above the +1 in the Z-Score chart we will sell 1 share of the overvalued stock and buy 1 share of the undervalued share. It's exactly the same strategy if the pricing is below -1 in the Z-Score chart.

So basically we will open a position accordingly to the signals generated by the Z-Score chart. If the curve in the Z-score chart it's > +1 we will open a short position in the spread.If the curve in the Z-score chart it's < -1 we will open a long position in the spread. We will keep trading while the curve is out of the range of -1 or +1.

But now the second part of the trade is when should we close the positions.

If we have a short position in the spread, we will close this position when the Z-Score is +0.5 or lower, meaning we are seeing a reversion in the spreads and we should monetize. But we if we have a long position in the spread, we will close this position when the Z-Score is -0.5 or higher.

We tested our strategy for the year of 2021, basically we ran the strategy for almost 1 year, was exactly 245 business days.

```
In [37]: def trade(S1, S2, spread, window1, window2):
             # If window length is 0, algorithm doesn't make sense, so exit
             if (window1 == 0) or (window2 == 0):
                 return 0

             ma1 = spread.rolling(window=window1, center=False).mean()
             ma2 = spread.rolling(window=window2, center=False).mean()
             std = spread.rolling(window=window2, center=False).std()
             zscore = (ma1 - ma2)/std
             # Simulate trading, assume we start with $0 and no positions
             money = 0
             countS1 = 0
             countS2 = 0
             for i in range(len(spread)):
                 # Sell short if the z-score is > 1
                 if zscore[i] > 1:
                     money += S1[i] - S2[i] * spread[i]
                     countS1 -= 1
                     countS2 += spread[i]
                 # Buy long if the z-score is < -1
                 elif zscore[i] < -1:
                     money -= S1[i] - S2[i] * spread[i]
                     countS1 += 1
                     countS2 -= spread[i]
                 # Clear positions if the z-score between -.5 and .5
                 elif abs(zscore[i]) < 0.5:
                     money += countS1 * S1[i] - S2[i] * countS2
                     countS1 = 0
                     countS2 = 0
             return money

In [38]: #data = test_data['Actual_Spread']
         profit = trade(S1, S2, data, 30, 5)
         profit

Out[38]: 15356.376554099214
```

Figure 10: *Part of our implementation of the trading strategy to make sure our trading signals were good to do some trades.*

So our trading strategy running for 245 days we had a profit of 15,356.00 USD. We considered was a pretty good trading result as the positions were very small, every time we traded was just 1 share per day.

10. CONCLUSION AND FUTURE WORK

- As we already discussed but it's good to make the point again. Our Random Forest model perform much better using the Spreads as the features to train the model than using the technical indicators.

- We had pretty good results using the Random Forest approach, but the Arima method was more accurate.

- Our data selection contained pre-covid, covid and now after pandemic situation, we might get even better results if we dont get a period with so many disruptions in the market.

- If we can collect more data, we probably can improve even more our results. We just trained our model with 3 years of data.

- In future work might be good to generate less trading, improving our accuracy for the trading signals. So we could have less points of trading but our trading strategy could be even better just trading right before the reversion for example rather than trading for multiple days until we have the signs of reversion.