International Journal of Innovations in Engineering and Technology (IJIET)
http://dx.doi.org/10.21172/ijiet.134.11

Statistical Arbitrage Trading Using Machine Learning Algorithms

Kaushik Ramnath G[1]
[1]*Department of Computer Science, Student, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu 603203, India*

Abstract- Pair trading strategy or statistical arbitrage strategy is a quantitative trading strategy that exploits the stock market that is out of equilibrium. Pair trading strategy is a market neutral strategy which means that we can make profit irrespective of the market trend. By choosing a pair of stocks that move together and assuming that their price is mean reverting, a trader can profit from the deviations from the mean by taking a long-short position in the chosen pair. This research analyses the performance of both Supervised and Unsupervised Machine Learning algorithms in Pair Trading and uses Python programming language to automate this trading strategy in the Indian market. Moreover, this research executes pair trading through a method called co-integration. In Supervised Learning approach, we use a Linear Regression model and in Unsupervised Learning approach, we use Principal Component Analysis for extracting risk factors of a stock and Density-Based Clustering for grouping the stock pairs together. Finally, this trading strategy was back-tested and programmed in Python for automatically triggering buy and short signals in the stock market.
Index Terms -Supervised Learning, Unsupervised Learning, Long, Short , Mean reversion

I. INTRODUCTION

1.1 Background

The pair trading strategy was first introduced in early 80's by Morgan Stanley. After this discovery it was adopted by a lot of trading firms in Wall Street. Unlike all the other trading strategy this does not bet on market trend/movement. First, a pair of stocks is selected from the same sector (eg: banking sector, automobile sector etc.), that are known to move together historically. Spread or Residual is calculated from the pair of stocks which in turn is used for tracking the deviations from the mean. An investor buys the undervalued stock and shorts the overvalued stock. As soon as the residual or spread converges back to its mean, the trader should exit buy and sell positions, resulting in a profit.\

1.2 Problem trying to solve

By identifying pair of stocks that has the mean reversion power, one can buy/short on those stocks. Picking the right pair of stock and triggering long and short signals automatically is an uphill task. This research solves these two problems using Supervised and Unsupervised Machine Learning Algorithms

1.3 Benefits of the proposed solution

This research provides the best optimal solution for profitable trading using Supervised and Unsupervised machine learning algorithms in Python. This research examines statistical arbitrage through co-integration pairs trading which is used to find the pair of stocks which are highly correlated. Co-integration yields better solutions when compared to other techniques like correlation, distance, stochastic and stochastic differential residual. The main benefit in this research is that this complex trading strategy has been automated using Python. Comparison of both Supervised and Unsupervised algorithms in pair trading is provided which gives an overall view of the advantages and disadvantages of both algorithms. This research also gives an in-depth knowledge of how both Supervised and Unsupervised algorithm plays a huge role in finding a stock pair and triggering buy and short signals. Another attractive feature of this research on pairs trading is the ability to profit whether the market is going up, down or sideways.

1.4 How is the solution better/different from existing solutions?

There is no research which analyses both supervised and unsupervised machine learning algorithm in pair trading. The combination of both these techniques gives an in-depth view of the usage of machine learning in pair trading. Moreover, this research examines statistical arbitrage through co-integration pairs trading whereas others mostly use correlation, distance, time series or stochastic differential residual. In this research, Python code is implemented to automate the pair trade easily and efficiently. In Unsupervised machine learning algorithm, mostly K-nearest neighbours algorithm is used for clustering but we use Density-Based Spatial Clustering of Applications with Noise(DBSCAN) algorithm to cluster the stocks with similar risk profiles because it does not require a predefined number of clusters in advance. In Linear regression, the residuals are used for the ADF test because residuals display

certain properties which can help identify pair trading pattern and also in Principal Component Analysis model we use Adjacent Closing Price instead of Actual Closing Price.

II. PROPOSED ALGORITHM

2.1. Supervised Learning Algorithm

2.1.1 Linear Regression

Linear regression is used to find a straight relationship between two continuous variables. One variable is a predictor or independent variable and another variable is the response or dependent variable. It looks for a statistical relationship between two variables and not deterministic relationship (if one variable can be accurately expressed by the other). Actually, a line is fitted when a Linear Regression is used between two variables. The line is called as best fit line when the error is minimum for all the data points. This best fit line is used for the prediction purpose. Mathematically this best fit line is nothing but a linear equation (y=mx+c). Linear Regression gives the value of both intercept and slope of the equation. With the help of this, one can predict the value of y (dependent variable). For example in stock market, we can apply linear regression to two stock and with the help of slope and intercept from the linear regression output, one can predict the value of the dependent stock. Nowadays Python and MS-Excel are majorly used for Linear Regression.

2.1.2 Residuals

The difference between the observed stock value of the dependent variable (y) and the predicted stock value ($\hat{y}$) is called the residual (e). Each data point has one residual.

Residual = Observed value – Predicted value

Table1: Values of 2 variables

X	Y
11	4
13	6
9	5
10	28
21	37
18	22
19	29

In table1, let's assume X and Y as 2 stock prices. If linear regression is applied to these values, we get:
Slope of the equation = 2.010225
Intercept (or constant) = -10.2904
The linear equation for the table:
y = 2.010225 *x - 10.2904
This equation helps us to predict the value of dependent variable (y) for a given unknown independent variable (x)-

Table1.1: Predicting the value of Y

X	Y
11	4
13	6
9	5
10	28
21	37
18	22
19	29
17	???

In table1.1 a new data point for x is added (17), now using the slope and intercept, one can predict the value of y.
y = 2.010225 *17 - 10.2904
=34.173825-10.2904

= 23.88
So, if x is 17, the predicted value of y is 23.88.Is this prediction accurate? The answer is it's not accurate. This is where the concept of residuals takes place.
For example, let's consider the value of x is 18 (refer to the last but one data point), then predictor value should be:
y = 2.010225 *18 - 10.2904
=36.18405-10.2904
=25.89365
However, the actual value of y is 22 in the table.
Now we can find two values of y:
Predicted value of y using linear equation
Actual value of y from the table
The difference between the two values of y is called the residuals. For example, the residual for y (difference between actual and predicted y), when x = 18 is
Residual= 25.89365 – 22
= 3.89365
We actually use residuals because they are stationary and normally distributed which is very essential for pair trading.

2.1.3 Error Ratio
To find which stock is dependent and which stock is independent depend on Error Ratio. Error Ratio is nothing but a ratio of Standard Error of the Intercept to the Standard Error. Standard Error of intercept and Standard Error values can be found in the output sheet of MS-Excel when Linear Regression is applied to two variables. Standard Error of intercept is the variance of the intercept and Standard Error is the standard deviation of the residuals. Applying regression to Stock 1 with Stock 2 and Stock 2 with Stock 1 yields two ratio values. Whichever gives the lowest error ratio tells us which stock is dependent variable and which stock is independent variable [3].
Error Ratio = Standard Error of Intercept / Standard Error

2.1.4 ADF test (Augmented Dickey-Fuller Test)
If two stock/time series are co-integrated, then it means that the two stocks move together and one can expect the two stocks to revert back to its mean. This is actually the concept of pair trading. We need stocks that possess the ability to revert back to its mean. So, for pair trading two stocks has to be co-integrated in order to bet on mean-reversion.
To check if two stocks are co-integrated, we use ADF test which throws a probability output. First we need to run a linear regression on two stocks, then take up the residuals from the linear regression output, and check if the residual is stationary or not. If it is stationary, then two stocks are said to be co-integrated. As discussed before if they are co-integrated, the two stocks move together and the percentage of the stocks to revert back to the mean is high [1].
We can assume a time series is stationary if it satisfies the following three conditions:

1. constant mean
2. constant standard deviation
3. no autocorrelation within the series

The ADF test is one of the best to test the stationarity of a time series. Here, we use residual as our time series. As I mentioned before the ADF test calculates the three conditions and throw a probability output for the time series (residuals). If the output of the ADF test of a time series is 0.35, then this means the series has a 35% chance of not being stationary or in other words, there is 75% chance of being stationary. Statistically this probability values is known as P-value. The threshold P-value for our time series is 0.05 (5%) or lower. This means that there is 95% chance of the time series being stationary.

2.1.5 Beta value
Beta value is nothing but the slope value of the linear equation. Beta value tells us about the number stocks required to hedge 1 stock of y. For example let's use the same table above:

HDFC PRICE	ICICI PRICE
11	4
13	6
9	5
10	28

21	37
18	22
19	29

Table1.2: Values for understanding Beta

From table 1.2:
Slope of the equation = 2.010225 (This is the beta value)
Intercept (or constant) = -10.2904
ICICI PRICE= 2.010225 *HDFC PRICE - 10.2904
This equation tells us 2.010225 shares of HDFC equals 1 share of ICICI.
Negative beta value (slope) should be avoided as it is not suitable for trading.

2.2. Un-Supervised Learning Algorithm

Unlike Supervised Learning, we don't have a target variable in Unsupervised Learning. In unsupervised learning we don't have a target/dependent variable to feed the learning model. For situations like this where we don't have dependent variable we use Un-supervised algorithms like Clustering and Principal Component Analysis (PCA).
In this algorithm we use PCA for extracting the risk factors of stocks price/return. We then introduce a clustering algorithm known as DBSCAN (Density-Based Spatial Clustering of Applications with Noise) for grouping the stocks for pair trading. Finally T-SNE algorithm is used to visualize the multiple dimensions of the data.

2.2.1 PCA (Principal Component Analysis):

PCA is a method of compressing a lot of data into something that captures the essence of the original data. PCA takes a dataset with a lot of dimensions and flattens it to 2 or 3 dimensions so we can look at it. PCA creates new variables known as principal components. The 1st principal component will try to explain the direction of the most variation. The 2nd principal component will try to explain the remaining variability. Actually these components are calculating the systematic risks.
In our data, we apply PCA on stock returns. PCA plot converts the correlations among all of the cells into a 2-D graph. We use PCA to find the common risk factors of stock returns which is helpful to group pairs accordingly using clustering algorithms.
According to Kim (2005) the components are divided into three parts:

1. The 1st component tells about the market risk
2. The 2nd component tells about the synchronized fluctuations
3. The other components tells about the random fluctuations

The number of clusters is determined by a technique known as elbowing in which the components are iterated one by one and finally determines a point where the principal component cost function is small and insignificant.

2.2.2 Density-Based Spatial Clustering of Applications with Noise (DBSCAN):

After finding the principal components, we are now ready to group these components accordingly. The grouping/clustering is done through a technique called DBSCAN. DBSCAN finds the dense regions and clusters them. DBSCAN groups these dense regions using Euclidean distance which groups the points that are close to each other. Unlike k-means clustering, DBSCAN doesn't require predefined count of clusters in advance.
Initially it DBSCAN starts with a random data point. If the data points are close to each other it is grouped as one cluster. The closeness is calculated using the Euclidean Distance Formula. If the data points don't get clustered it is considered as noise. This process is iterated until all cluster are formed

2.2.3 t-Distributed Stochastic Neighbour Embedding (t-SNE)

After clustering the data, we need to visualize this high dimensional data. Since there are numerous dimensions and observations, we humans cannot visualize these high dimensional data. To show the high dimensional data in low dimensional visualization we use a nonlinear dimensionality technique known as t-SNE (t-Distributed Stochastic Neighbour Embedding).
Initially the t-SNE algorithm creates a probability distribution for both similar and dissimilar data points. The similar data points have high probability of being picked. Then again comes another probability distribution for over the points in the low dimensional map. This is used to map the dissimilar data points far apart in the map to show the low dimensional points.
This t-SNE algorithm is used in many industries for visualizing high dimensional data. It is used in Cancer Research, Computer Security Research, Neural Networks etc.

2.2.4 Stationarity and Co-integration:

A time series is said to be stationary if the values do not change over time. As discussed in Section (2.4) , to be stationary a time series must satisfy 3 conditions-(i) constant mean (ii) constant standard deviation (iii) no autocorrelation. The most imperative feature of stationary series is the constant mean because when the spread/residual/ratio deviates from the mean, we can capitalize this trading opportunity.

Co-integration implies that two time series share similar trends and since they are stationary they don't diverge too far from each other. So the overall concept is if we find two stocks that are stationary and co-integrated, then any short-term deviations from the mean, can be an opportunity to place trades which means we are betting on mean reversion theory. Again we use ADF test for checking stationarity of a time series. In this test there are two hypothesis (i) the null hypothesis is that there is no co-integration and (ii) alternate hypothesis is that there is a co-integrated relationship. The output of this test is a probability number (between 0 and 1). If the probability value is less than 0.05 (5%), then we say that the time series has 95% chance of being stationary and only 5% chance of not being stationary. So if the P-values is less than 0.05, then the time series is said to be stationary and therefore the two stocks/variables are co-integrated.

2.2.5 Trade Execution:

Here comes the final step for executing the trade using all these unsupervised concepts. Again we use the concept of mean reversion among the stocks which are previously clustered using PCA, DBSCAN and t- SNE algorithms. After the clustering the pairs, stationarity is checked for the time series. Now, comes the climax, where we combine all these concepts discussed above into a trading algorithm.

Unlike supervised learning algorithm, we use spread time series rather than taking residual of the series. So we focus on the spread of the stocks. To calculate the movement of the spread we use z-score which tells us the distance from the spread mean in terms of standard deviation. The z-score is nothing but the standard deviation of the spread. To calculate the deviation regularly we use rolling z-score which calculates the z-score regularly.

Like discussed in supervised learning, here also we use the same strategy for initiating the trades. A trade is executed when:

1. The spread goes less than -2 Z-score(z-score < -2) we long on the pair (buy y, short x) and exit the position if it hits -1 Z-score (z-score > -1)
2. The spread goes greater than 2 Z-score(z-score > 2) we short on the pair (buy y, short x) and exit the position if it hits +1 Z-score (z-score < +1)

So basically we initiate a trade when the z-score exceeds +2/-2 and exit the trade when it drops below +1 and above -1.Since fundamental data of Indian stocks are difficult to collect, this research shows only the method to implement it and not the back-testing algorithm. Implementation of this strategy with proper data can yield more profits.

2.2.6 Trade Identification

Finally the war is here. After proving that the residual/time series is stationary we are now going to use this residual to trigger buy and short signals.

1. A trade is executed when:
2. The residuals hit -2 standard deviation (-2 SD) we long on the pair (buy y, short x) and exit the position if it hits -1 standard deviation (-1 SD)

The residuals hit +2 standard deviation (+2 SD) we short on the pair (short y, buy x)
and exit the position if it hits =1 standard deviation (+1 SD)

The valid question would be here is to ask why 2nd standard deviation is taken. The answer to this question is that in a normally distributed data 95% of the data is within 2 standard deviations. From this we can say that the residual series has 95% chance to revert back to its mean.

III. EXPERIMENT AND RESULT

Python is used for executing all the outputs shown below. We use Indian banking stocks for backtesting the pair trading strategy.Nine bank stocks are chosen and the values of all the stocks are downloaded from yahoo finance website. Profit and Loss of these nine banking stocks are given at the end

```
#Visualizing the Adjacent Plots
res.hist(figsize=(15,10))
plt.show()
```

Fig. 1. Histogram plot of 9 banking stocks

```
#Fitting the linear regression model for each pair
of stocks and taking resiudal from the linear
regression output
def run_ADF_regression(t1,t2,i):
    x = res[t1]
    y = res[t2]
    reg = linear_model.LinearRegression()
    reg.fit(x.values.reshape((len(x), 1)),y)
    print(t1+'and'+t2)
    print(reg.intercept_)
    coef.append(reg.coef_)

    pred= x*reg.coef_ + reg.intercept_
    #Residual=Actual value - Predicted value
    residuals[i] = y-pred
    print(residuals[i])
    return reg
dep=['AXISBANK','BANKBARODA','FEDERALBNK','HDFCBANK'
,'ICICIBANK','KOTAKBANK','PNB','SBIN','YESBANK']
indep=['AXISBANK','BANKBARODA','FEDERALBNK','HDFCBAN
K','ICICIBANK','KOTAKBANK','PNB','SBIN','YESBANK']
inc=0
coef=[]
residuals=pd.DataFrame()
for i in dep:
    for j in indep:
        if i!=j:

            x=run_ADF_regression(i,j,inc)
            inc=inc+1
```

Fig.2. Function for finding residuals for all the closing stock values

In fig.2 the function takes 2 stock values (t1 and t2) and ‘i’ for incrementing the stock pairs one by one. Linear Regression is fitted to the stocks and Residual is calculated.

http://dx.doi.org/10.21172/ijiet.134.11

```
titles=[]
for i in dep:
        for j in indep:
            if i!=j:
                titles.append(i+'and'+j)
paircount=[]
for i in range(len(residuals.columns)):
    data=residuals.values
    #Checking for stationarity using ADF test
    result = adfuller(data[:,i])
    #Storing and visualizing the pairs which have pvalue less than 5%
    if result[1] < 0.05:
        print('ADF Statistic: %f' % result[0])
        print('p-value: %f' % result[1])
        print(i)
        paircount.append(i)
        plt.title(titles[i])
        plt.plot(residuals[i], c='green')
        plt.show()

print(paircount)
```

Fig. 3. Function for finding ADF values and perfect stock pairs

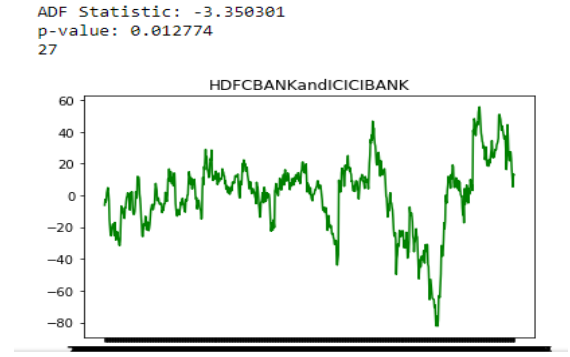

Fig. 4. Residual graph of HDFC Bank and ICICI Bank

Fig. 5. Residual graph of Axis Bank and Bank of Baroda

The code given in fig.3 is iterated on all the stock pairs. Actually there would be 72 pair of stocks for 9 banking stocks. From this only 11 pair of stocks satisfies the condition for ADF Test.
Like Fig.4 and Fig.5, there are 11 pair of stocks which has a p-value < 0.05.The 11 pair of stocks are:
(1)Axis bank and Bank of Baroda
(2) Federal Bank and SBIN

(3) Bank of Baroda and Axis bank
(4) HDFC Bank and Bank of Baroda
(5) HDFC Bank and ICICI Bank
(6) HDFC Bank and KOTAK
(7) ICICI Bank and Bank of Baroda
(8) ICICI Bank and HDFC Bank
(9) \KOTAK and Bank of Baroda
(10) KOTAK and HDFC Bank
(11) PNB and Bank of Baroda

These 11 pairs are put into MS-Excel for checking Error Ratio which can be calculated from the output of Linear Regression.
Out of 11 pairs, three (3) pairs of stocks gets eliminated because of high error ratio
They are (1) Bank of Baroda and Axis bank (2) ICICI Bank and HDFC Bank (3) KOTAK and HDFC Bank
Error Ratio rejects 3 pairs of stock. They are: 1. Bank of Baroda and Axis bank 2. ICICI and HDFC 3.Kotak and HDFC
A trade is executed when:
The residuals hit -2 standard deviation (-2 SD) we long on the pair (buy y, short x) and exit the position if it hits -1 standard deviation (-1 SD)
The residuals hit +2 standard deviation (+2 SD) we short on the pair (short y, buy x)
and exit the position if it hits =1 standard deviation (+1 SD)
So finally the profit and loss are calculated for the 8 pair of stocks.

Profits:
1. HDFC BANK AND ICICI BANK = Rs. 17313.5459
2. FEDERAL BANK AND SBIN = Rs.8302.90
3. KOTAK AND BANKOFBARODA = Rs.3297.148
4. AXIS BANK AND BARODA = Rs.1345.533
5. HDFC AND BARODA = Rs.13973.648
TOTAL PROFIT = Rs. 44,232.75

Loss:
6. HDFC AND KOTAK= -13932.801
7. ICICI AND BARODA= -1358.500
8. PNB AND BARODA = -3723.402
TOTAL LOSS = Rs. -19,017.7
OVERALL PROFIT AND LOSS = Rs. 25,215.05

IV. CONCLUSION

Pair Trading using Supervised Learning and Un-Supervised Learning has been explained and implemented successfully. In Un-Supervised Learning the clusters were formed using PCA and DBSCAN algorithms. In Supervised Learning method Profit and Loss for 8 banking pair stocks were calculated and the overall profit and loss was Rs.25000. For Un-Supervised Learning method fundamental data and stocks prices of US companies can be taken from quantopian pipeline, since fundamental information of the Indian companies are difficult to find in the internet. In Supervised Learning method, stock prices were download from yahoo finance website. It is evident that this strategy is profitable if it is executed properly.

V.REFERENCE

[1] Hakon Andersen & Hakon Tronvoll (2005). Statistical arbitrage trading with implementation of machine learning
[2] Implementation of Machine Learning. Norwegian School of Economics Bergen, Spring 2018
[3] Kathik Rangappa,'Module 10:Trading Systems, [Online]. Available: https://zerodha.com/varsity/module/trading-systems/
[4] Gopal Rao Madhavaram (2013) Statistical Arbitrage Using Pairs Trading With Support Vector Machine Learning, Saint Mary's University