Kyan Agdassi

20 December 2024

IEOR 198

Trading Algorithm Results

Abstract:

In my project, I looked for correlated stocks, and long and shorted those stocks. More specifically, I sorted the covariance between every pair of stocks and then executed long and short purchasing based on whether or not the behavior of the two stocks during the current time was reflective of that relationship. I used 2023 data and tested my algorithm on 2024 data. My scale was by week, so I used the stock data from noon on every Tuesday in 2023 and 2024. I used pyfinance for my data.

For my project, I created an algorithm that details recommendations for executing long and shorts for stocks from the S&P 500. My approach for the project was to backtest an algorithm that I developed for 2023 using practical 2024 data.

To start, I began implementing an algorithm using how correlated two stocks were during the 2023 year. I did this by using data such as the mean, how the stocks moved together, and the frequency at which they moved together. One potential downside of this approach is that it was difficult to find the mean of a certain stock because when stocks split, it makes it seem as if the stock went down a lot according to the pyfinance data. However, in reality, this can actually be a sign that the stock went up dramatically. Nonetheless, another issue with calculating the mean was that averaging all the data from multiple weeks is not an accurate representation of the mean of the stock, for I only sampled from Tuesday's at midday.

The frequency of my trading was weekly, so I collected weekly data starting from 2023 until the present. To calculate the covariance and maintain it effectively, I calculated the expected value of the product of the difference between stock 1's price and its expected value and the difference between stock 2's price and its expected value. This resulted in 500 choose 2 rows. After calculating the covariance between two stocks, I sorted the data, while still maintaining which stocks they were. If I hadn't I could have potentially run into issues where I'd lose the data I had collected. Nonetheless, I sorted the stocks in order of covariance. For stocks with high (and positive) covariance, it is logical to ascertain that if stock A increases dramatically and stock B plummets dramatically, then shorting stock A and longing stock B would be optimal. My portfolio implemented this strategy.

sorted_covariance_pairs

Stock 1	Stock 2	Covariance
EL	SMCI	0.018425922862110100
CE	SMCI	0.01565456759626900
QRVO	SMCI	0.015086277138027700
CEG	SMCI	0.009881917104610470
AMTM	INCY	0.009798171222255300
SMCI	WYNN	0.009707894943056240
APTV	SMCI	0.009511480900204240
CVS	SMCI	0.009480179149189260
HII	SMCI	0.009439630655620510
AMTM	LDOS	0.009382757071811690
MGM	SMCI	0.008616267326830700
AXON	EPAM	0.008591447683736370
MPWR	SMCI	0.007833610415950390
EPAM	FTNT	0.007781439726750670
AXON	FTNT	0.007669426126464140
CE	QRVO	0.0076556854513701700
EL	QRVO	0.00763531320449212
CZR	SMCI	0.007556128648365680
AMD	SMCI	0.00740067574579783
NXPI	SMCI	0.007315428669291950
CE	EL	0.0073022578737482200
AMTM	CRL	0.007076452274833080

One potential downside of only looking at covariance is that I did not consider the variance of the stocks individually, meaning I did not consider how volatile and risky the stocks were. This is potentially negative because I was not managing how unpredictable stocks were, which could potentially lead to massive portfolio losses.

From tracking my data, I found that the stocks with the highest covariance were EL and SMCI. After checking the empirical data, this looks to be logical, as both stocks have decreased in price in tandem during the last few years. Furthermore, it is worth noting that the vast majority of the stocks in the S&P 500 appear to be uncorrelated, as the magnitude of their covariance is quite low. This does not imply independence, yet I was expecting the stocks to move more in tandem (or in contrast) with each other.

As a result of considering the top covariance pairs from 2023, I tested my algorithm on 2024 data. My algorithm chose the stocks with the highest covariance each week to guide how I should buy/sell stocks based on how their “pair” stock performed. For instance, if stock A and stock B have a high positive covariance, and if stock A dipped and stock B increased, then my algorithm would long stock A.

After backtesting my algorithm on 30 weeks of 2024 data using the 2023 covariances, my test closed at a 1.85% profit. Unfortunately, the S&P 500 closed at an approximately 25% profit margin during that same interval of time (first thirty weeks of 2024), which indicates that my algorithm is faulty. One potential problem with my algorithm is I did not budget my money using Kelly betting, for instance. It appears as though I profited on most days, but massive losses on key days resulted in an overall dismal performance.

If I were to complete this project again, I would incorporate the variance of stocks to minimize massive dips in my performance. This was the main issue with my approach. Below is a table of the results.

30_weeks_trade_results

Stock	Open Price	Close Price	Final Value	Profit/Loss
EL	144.33999633789100	137.3000030517580	95.12263165806610	-4.877368341933890
EL	136.7100067138670	134.80999755859400	98.61019013827560	-1.389809861724420
EL	133.74000549316400	125.83000183105500	94.08553661042460	-5.914463389575450
EL	126.25	130.8000030517580	103.60396281327400	3.603962813273530
EL	130.77000427246100	134.1199951171880	102.56174255202000	2.5617425520204300
EL	159.4600067138670	143.33999633789100	89.89087564451060	-10.109124355489400
EL	143.9199981689450	146.3699951171880	101.70233253155400	1.7023325315542200
EL	144.0	149.99000549316400	104.15972603691900	4.15972603691948
EL	149.0	148.8300018310550	99.88590726916420	-0.11409273083577900
EL	148.5500030517580	149.5	100.63951324720700	0.6395132472068500
EL	150.0800018310550	149.75	99.78011605341920	-0.219883946580822
EL	149.27999877929700	143.17999267578100	95.91371506337280	-4.086284936627170
EL	143.82000732421900	154.14999389648400	107.1825796455280	7.182579645527600
EL	154.17999267578100	144.42999267578100	93.67622229655770	-6.323777703442290
EL	144.75999450683600	138.8000030517580	95.88284631027900	-4.1171536897210100
EL	140.63999938964800	144.41000366210900	102.68060600741000	2.6806060074104500
EL	145.3699951171880	147.4499969482420	101.430832978551	1.4308329785510000
EL	148.22000122070300	132.94000244140600	89.69100077354300	-10.308999226457000
EL	133.0	132.0	99.24812030075190	-0.7518796992481210
EL	132.99000549316400	134.75	101.32340358984800	1.3234035898482700
EL	134.4499969482420	126.05999755859400	93.75976230562630	-6.240237694373660
EL	126.0199966430660	123.36000061035200	97.88922702462140	-2.1107729753785800
EL	124.83000183105500	120.47000122070300	96.50724942209610	-3.4927505779038600
EL	120.0999984741210	113.9000015258790	94.83763777933920	-5.162362220660770
EL	112.91999816894500	113.8499984741210	100.82359220709900	0.8235922070990110
EL	113.8499984741210	106.4000015258790	93.45630474475970	-6.54369525524028
EL	108.37000274658200	106.3499984741210	98.13601160721140	-1.8639883927885700
EL	106.58999633789100	103.36000061035200	96.96970087389820	-3.0302991261018200
EL	102.33000183105500	99.18000030517580	96.92172239859870	-3.078277601401310
EL	98.88999938964840	100.72000122070300	101.85054286818600	1.8505428681863800