

Analyzing Historical S&P 500 Components:

Application of
Regularization
Techniques for
Portfolio Optimization

Alessandro Rossato
N. 875067

What is the research question?

- Is it possible to replicate the S&P 500 returns with fewer asset?
 - Which companies are the primary drivers of the S&P 500's performance?
 - What are the main sectors represented within the S&P 500 index?
- Testing this hypothesis with 4+1 year rolling portfolios shifted by 1 month from 1995 to 2024

Outline

1. Data sources description and preprocessing
2. Rolling windows and split
3. Regressions on training
4. Metrics of performance in test
5. Weights exploration
6. Remarks

Data sources

- **INDEX**

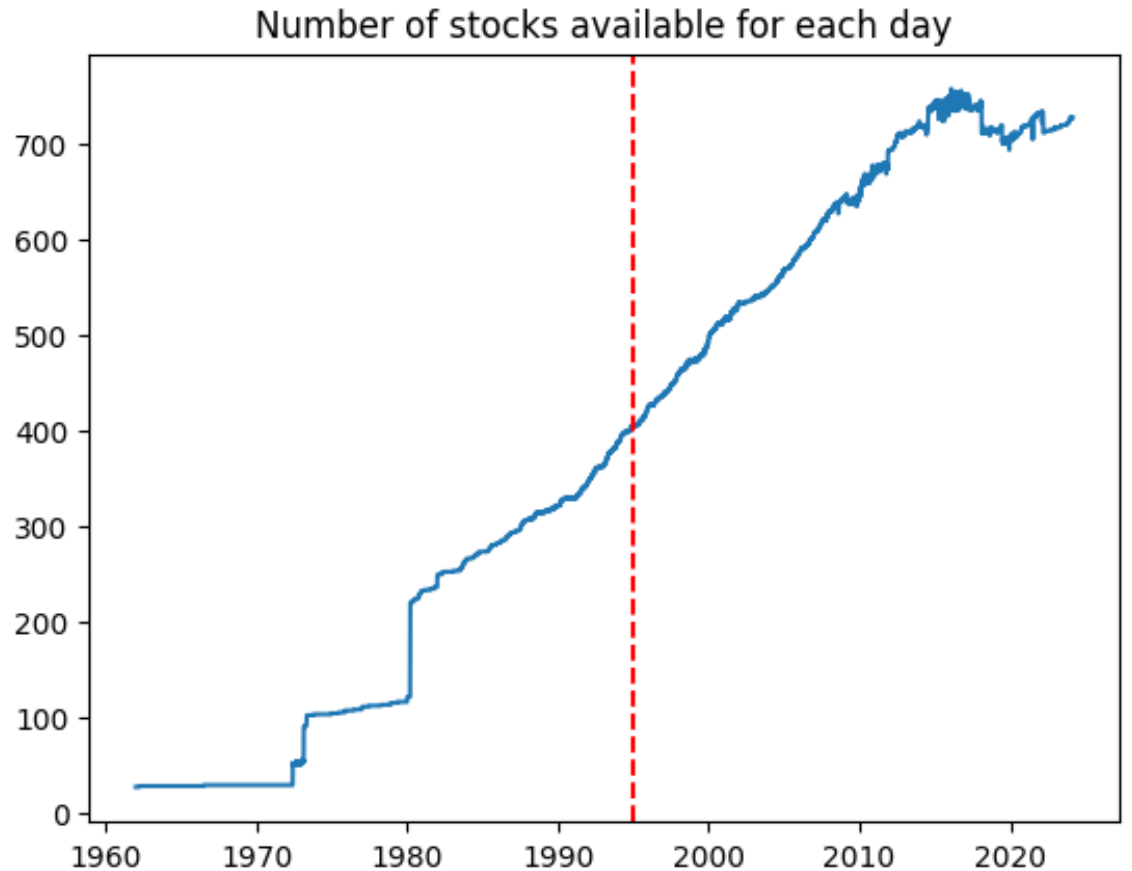
- All historical daily adjusted prices of the index from yahoo finance

- **STOCKS**

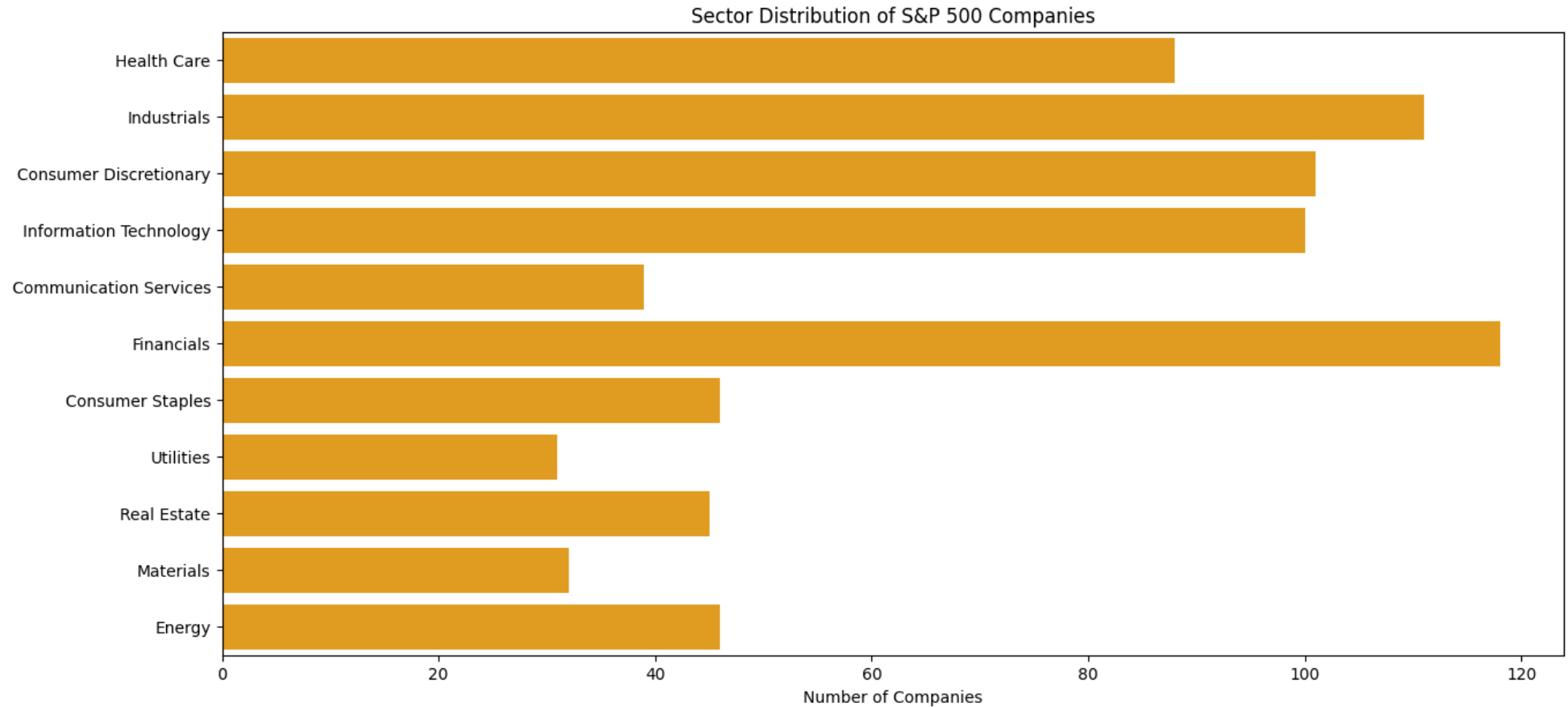
- Historical S&P500 components from 1996 to 2023 (thanks to [fja05680](#))
 - Example: 1996-01-02 AAL,AAMRQ,AAPL,ABI,ABS,ABT,ABX,ACKH,ACV,ADM,AD...
- Historical daily adjusted prices of the **currently listed constituents** from yahoo finance using of [yahooquery](#) package
- Stocks sectors are obtained from [financedatabase](#) package

Data preprocessing

- Filtering the index and stocks series **from 1995 to 2024**: limit of historical components and good number of price available
- Unstack the pivot dataframe
- Fill internal NaN with the mean and drop the stocks with more Nan
- Remove GOOGL for its perfect correlation with GOOG
- Exclude stocks with more than 50% change in one day (unstable result in regression).

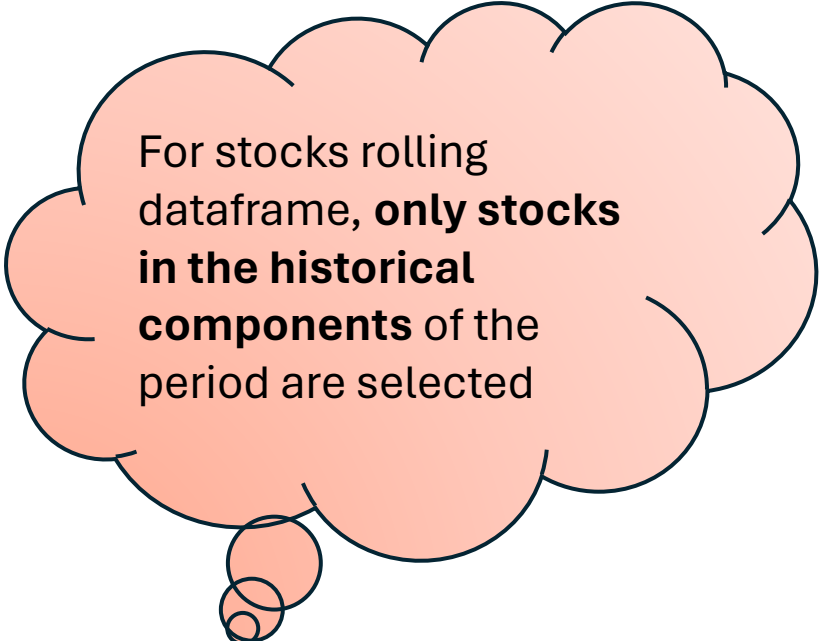


GICS Sectors distribution



Rolling windows creation

- Window information:
 - **5 years size**
 - **1-month shift**
- Range: **January 1995 – February 2024**
- Total windows: 290 index + 290 stocks
- Each window contains **daily prices**



For stocks rolling dataframe, **only stocks in the historical components** of the period are selected

Custom modules

Three simple modules are created to accomplish the following phases to create a replicable and extendible framework:

- **EF_regression.py**: computes the EF and the regressions
- **functions.py**: contains some useful functions to scale and split the data, get returns
- **plot.py**: plot some interesting graphs

Train and test dataset

- For loop for iterating across all the windows
- Scale the prices and calculate the returns
- Split each window in 4 years of train and 1 of test:
 - Avoid to go too far from the actual index composition
 - Monthly shift can balance and stabilize the variability of the short random time space
 - Other possible combination can be justified
- Obtain the risk-free rate of the test set at the start date for further use (same maturity)

Regressions on training

Portfolio optimizations for linear regression:

- Short selling is allowed (only sum of weights equal to 1 from budget constraint)

Portfolio optimizations for elastic net:

- Short selling is allowed (only sum of weights equal to 1 from budget constraint)
- Ratio l_1 equal to $\frac{1}{4}$
- Cross-validation to achieve best alpha → elastic net can handle correlated asset and bring the low coefficient to zero

Lasso was not included because his results were close to elastic net

Returns on test: the strategies

Returns on each test window with 4 different strategies:

- Index → ETF can do this easily and with low cost (great for retails)
- Equally weighted → Are larger capitalization firms equal to smaller ones?
- Linear regression weights → Elastic net comparison
- Elastic net weights → Interesting for those who wants to replicate index with fewer stocks (less costs of transactions and management)

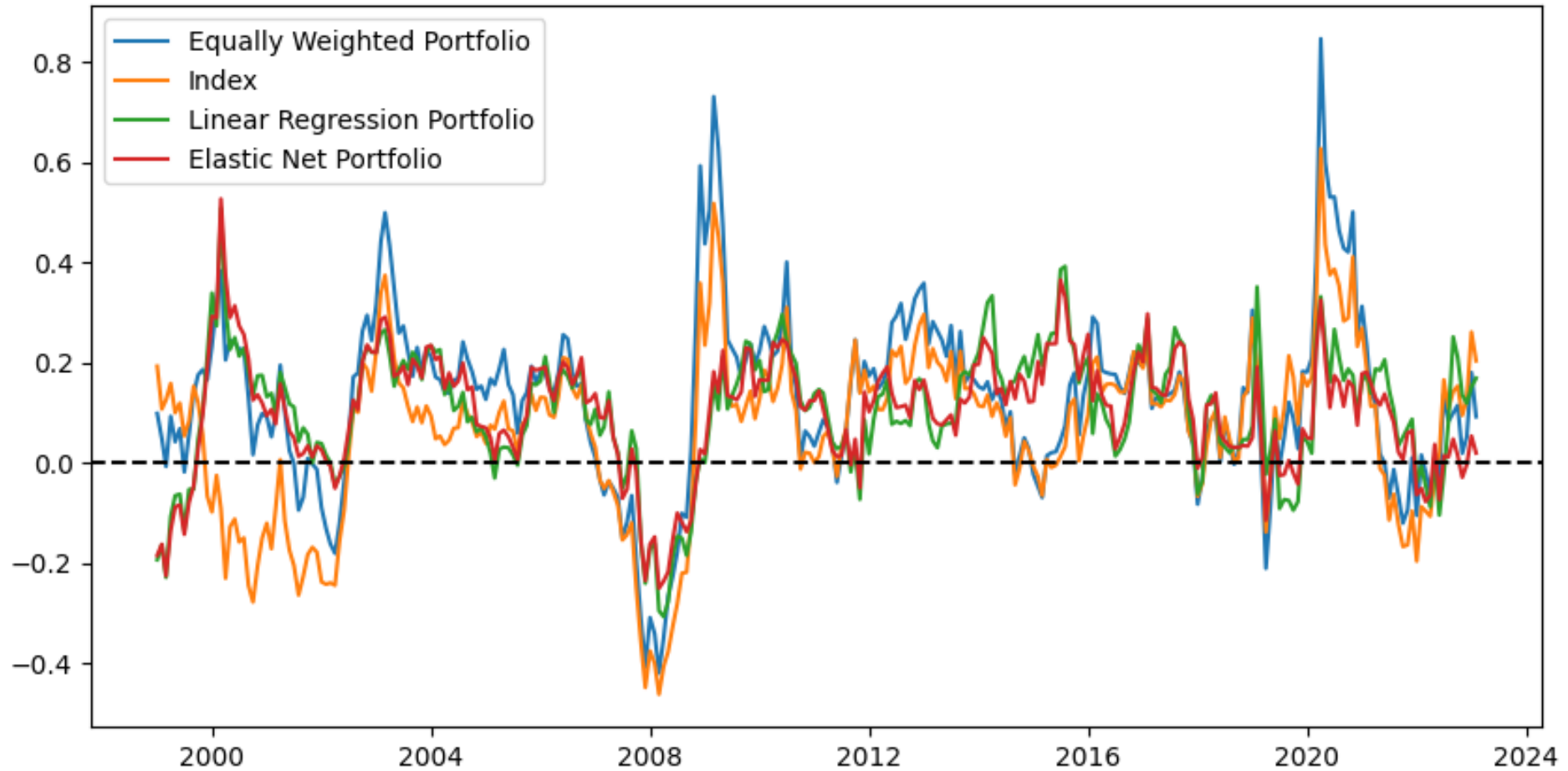
Metrics on the strategies

- Total returns for the year of investment
- Standard deviation
- Sharpe ratio (risk-free rate set as the Treasury Securities at 1-Year at start date of the windows)
- Maximum drawdown

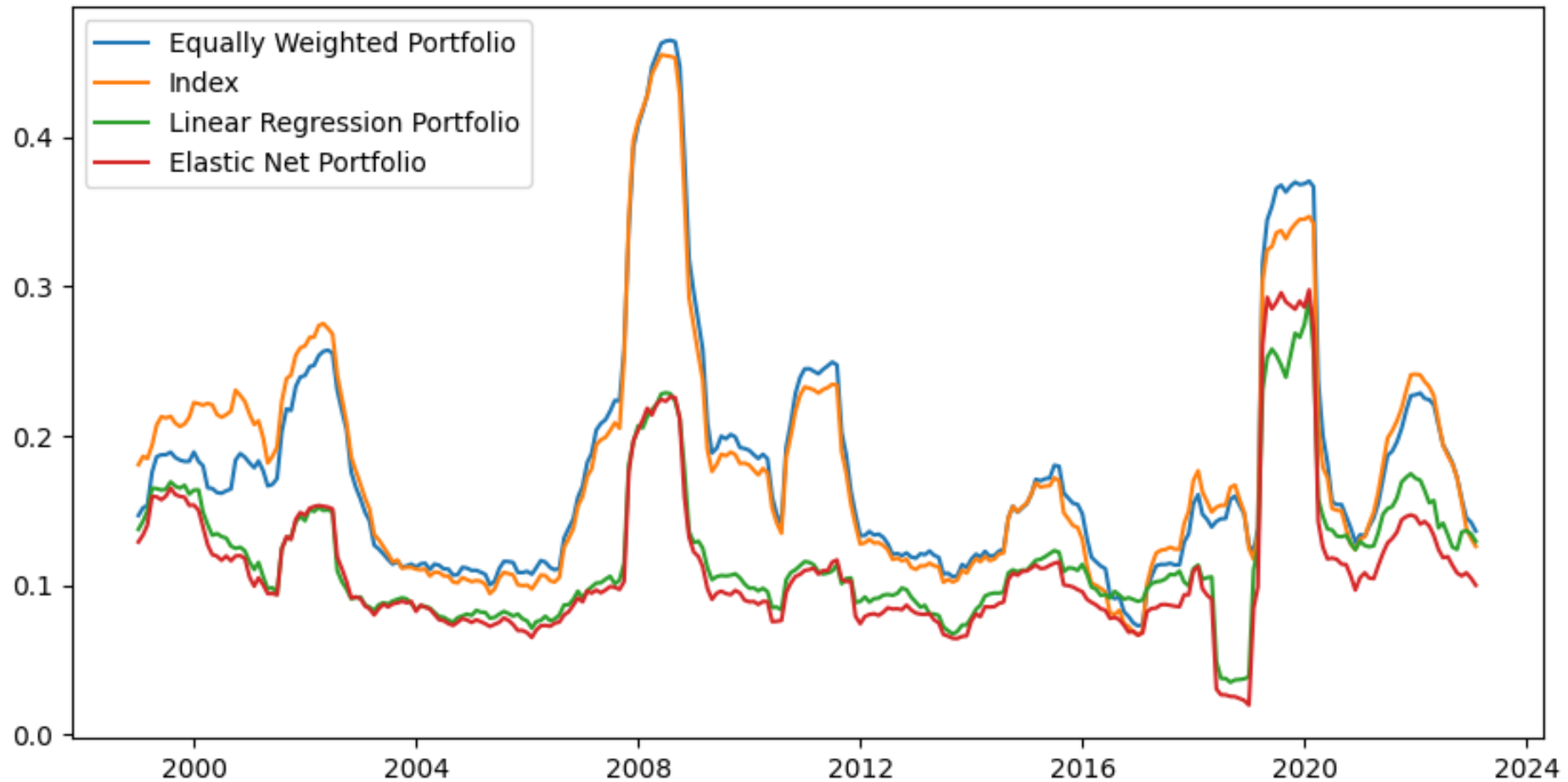
The table shows the average across all the test windows

Strategy	Avg Return	Avg Standard Deviation	Avg Sharpe Ratio	Avg Maximum Drawdown
Equally Weighted Portfolio	12.97	17.99	0.85	16.70
Index	6.52	17.99	0.50	17.15
Linear Regression Portfolio	10.54	11.85	0.92	11.64
Elastic Net Portfolio	10.20	11.13	1.00	10.79

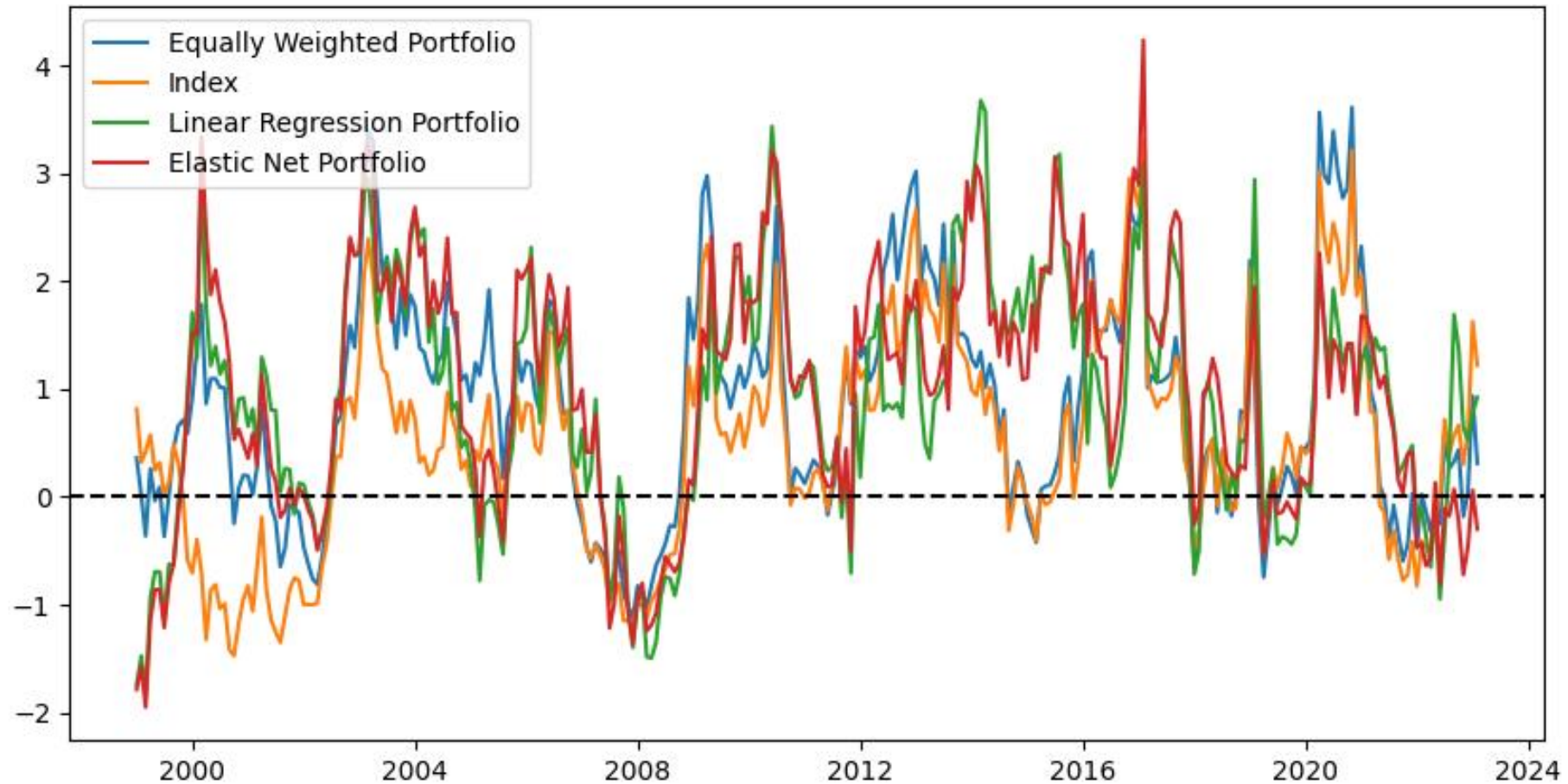
Strategies comparison: returns



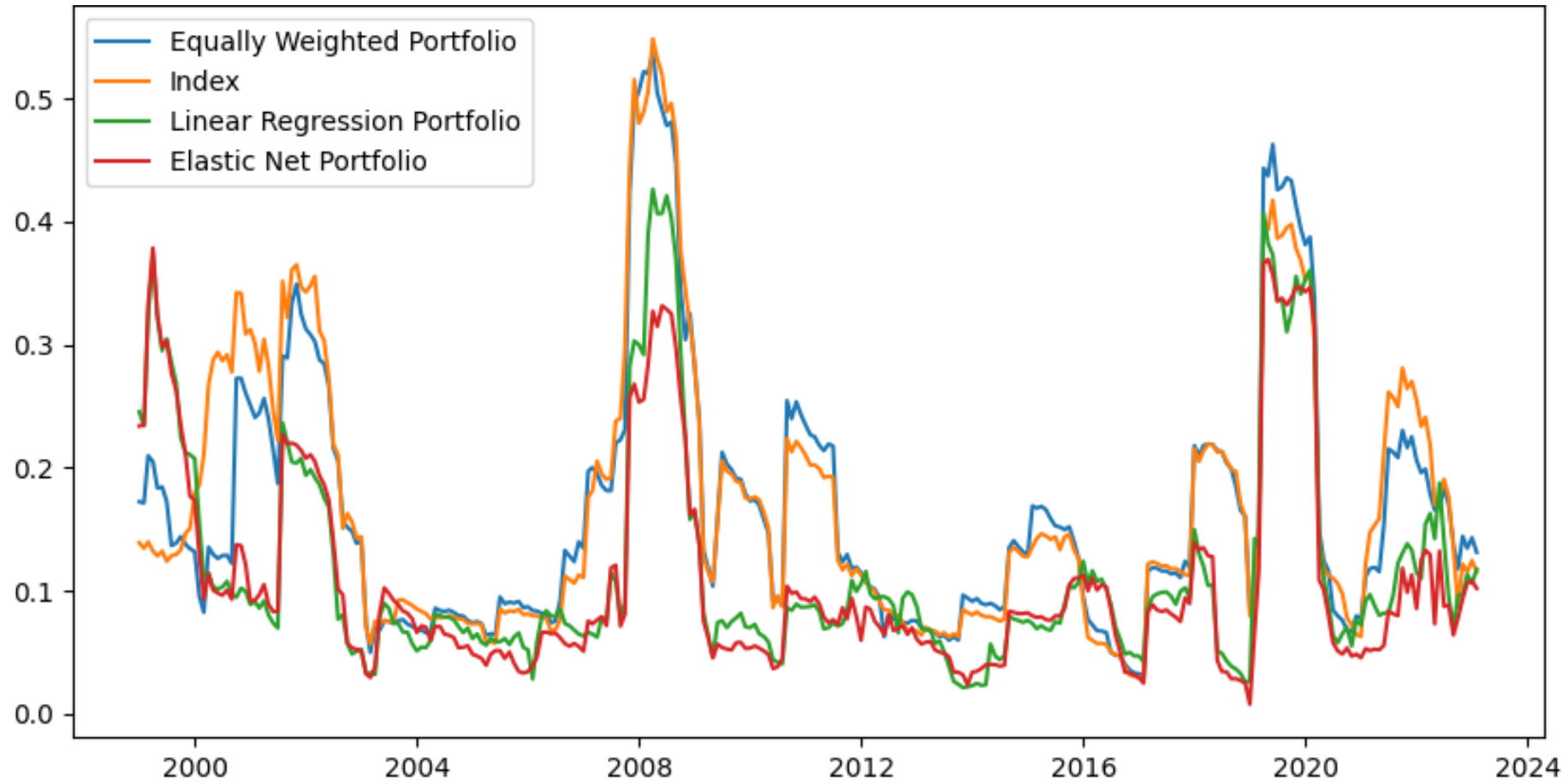
Strategies comparison: standard deviation



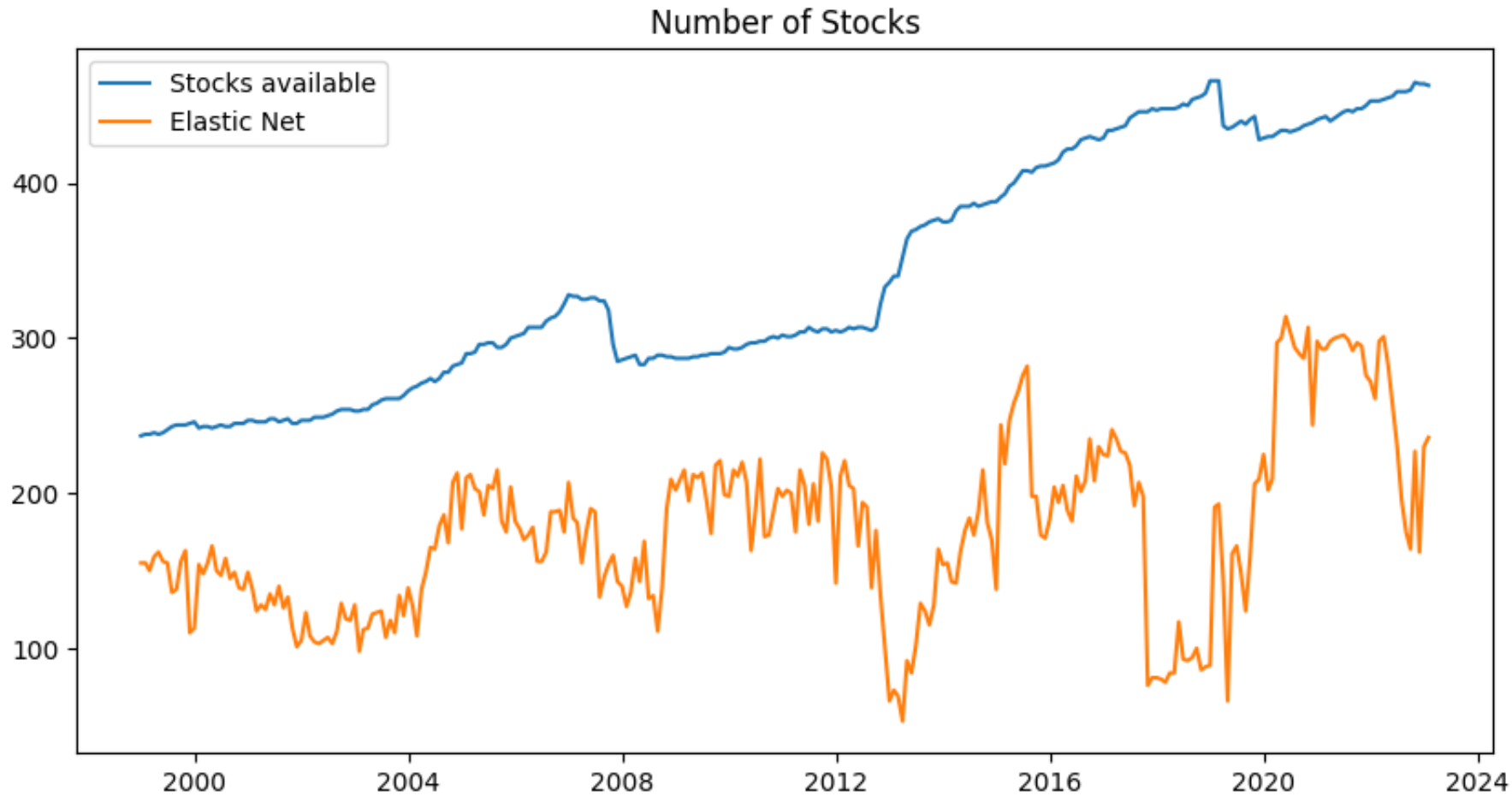
Strategies comparison: Sharpe ratio



Strategies comparison: drawdowns



Stocks selection power



Avg Stocks	Avg Elastic
340	176

Notable stocks in regressions

Average weights of stocks in different windows:

- “Heavier” stocks for both linear and elastic net (3-5%):

- CVG: Commercial Vehicle Group, Inc (Consumer Discretionary)
- SO: Southern Co (Utilities)
- MCD: McDonald's Corporation (Consumer Discretionary)
- JNJ: Johnson & Johnson (Healthcare)
- ED: Consolidated Edison, Inc. (Utilities)

AAPL
GOOGL
TSLA



- Stocks with lowest (negative) weights:

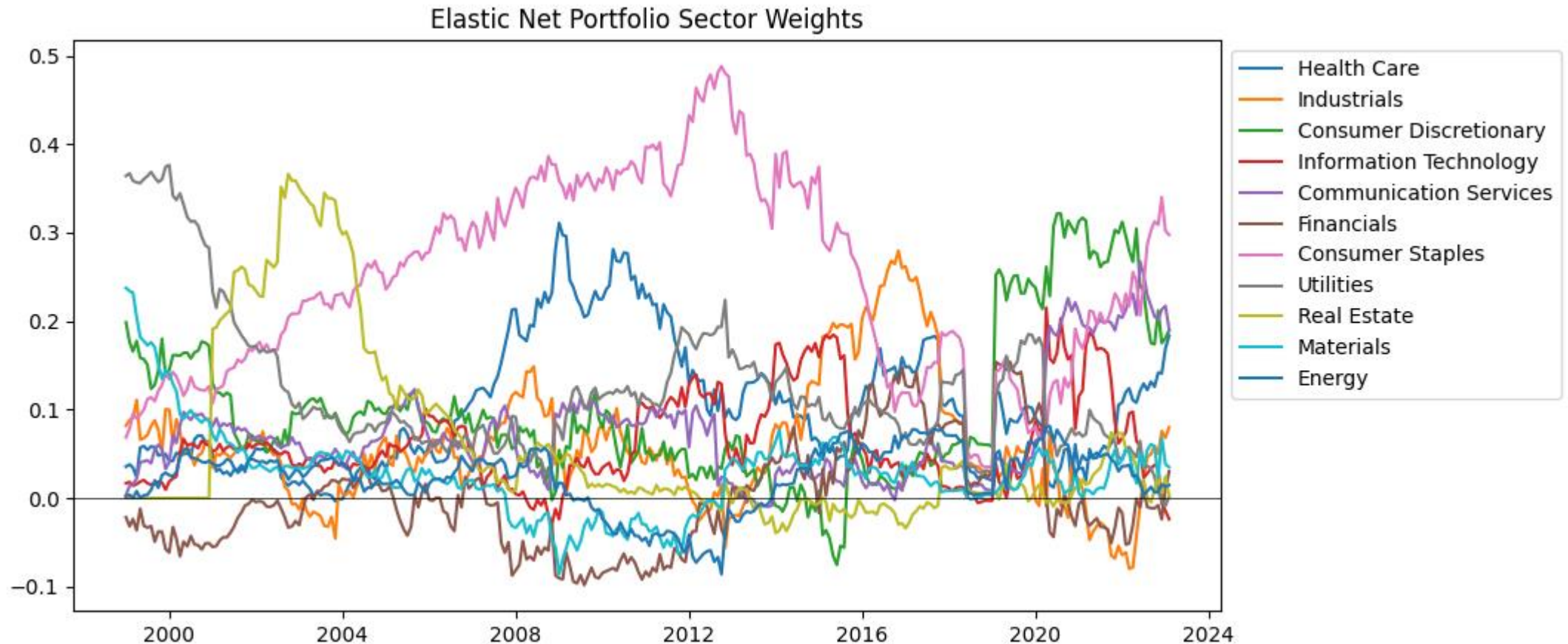
- Linear: AMP, PRU, PEG (Financial Services and Utilities)
- Elastic net: IVZ, TROW, AMP (Financial Services)

Sector weights in 2023 and historical averages

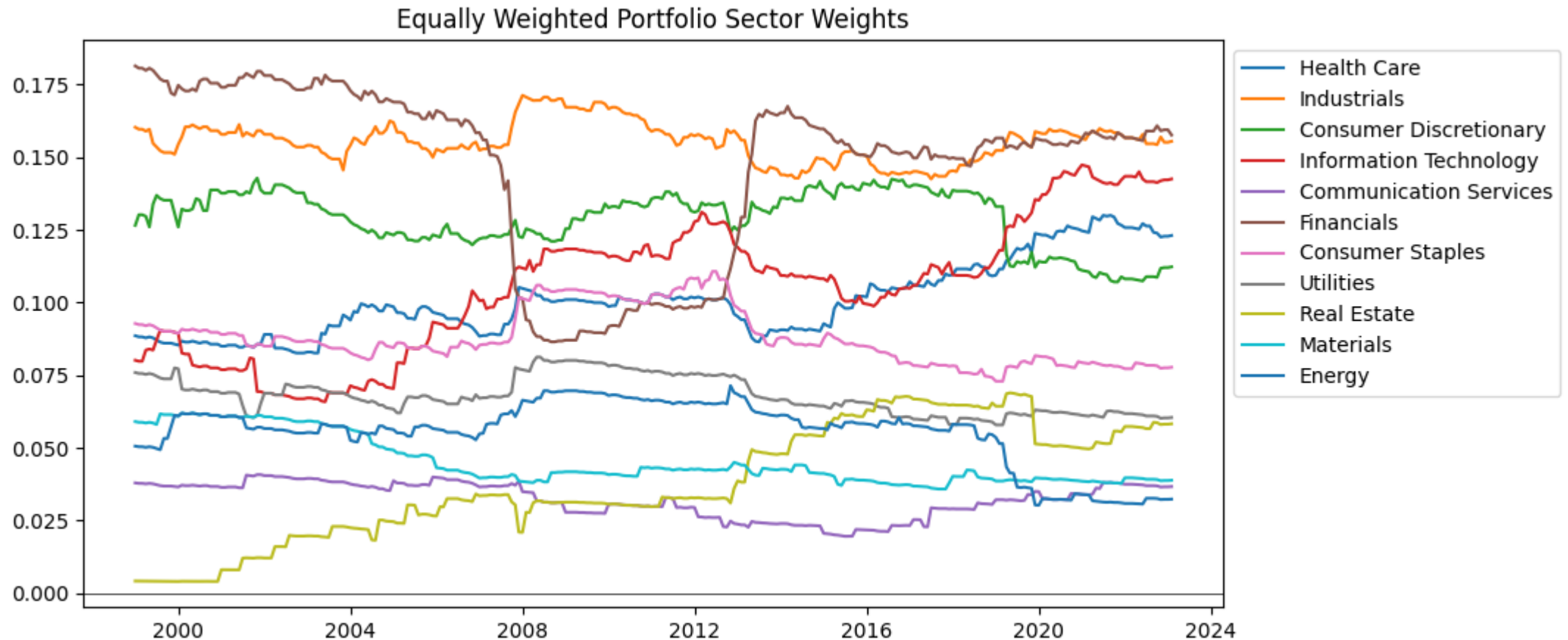
Original SP500 (1957):
425 industrials, 50 utility
and 25 railroad

Sector	Elastic net	Equally	Actual
Health Care	16.9 (10.2)	12.3 (10.0)	14.5
Industrials	6.8 (6.1)	15.5 (15.5)	8.6
Consumer Discretionary	18.1 (10.0)	11.2 (12.9)	9.9
Information Technology	-0.2 (6.3)	14.2 (10.6)	26.1
Communication Services	21.7 (7.3)	3.7 (3.2)	8.2
Financials	1.1 (> 0.1)	15.9 (14.8)	12.9
Consumer Staples	30.1 (24.8)	7.8 (8.8)	7.4
Utilities	-0.1 (12.3)	6.0 (6.8)	2.9
Real Estate	2.4 (5.9)	5.8 (3.7)	2.5
Materials	3.8 (2.7)	3.9 (4.5)	2.6
Energy	1.2 (2.3)	3.2 (5.5)	4.5

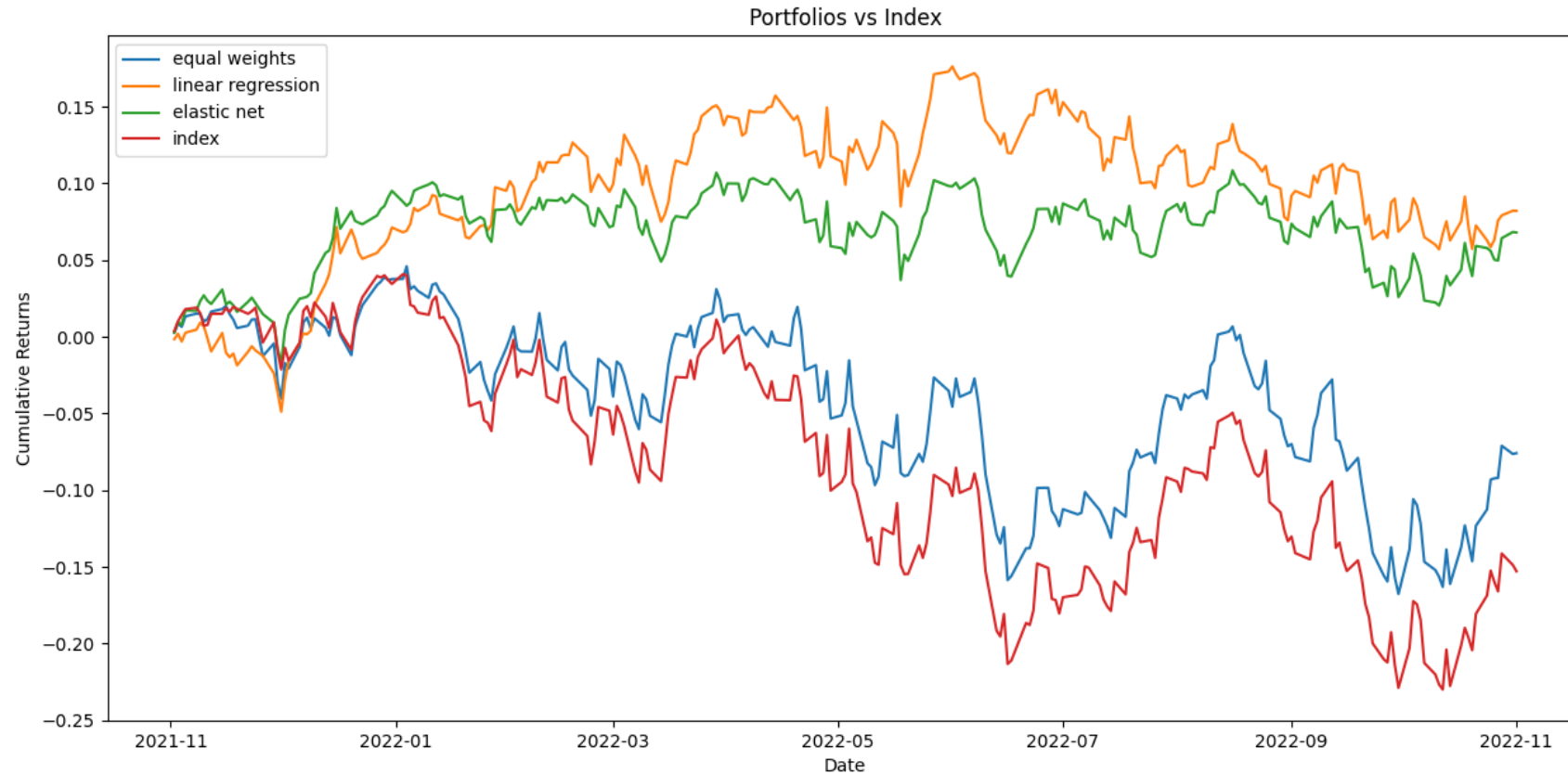
Sector weights: elastic net



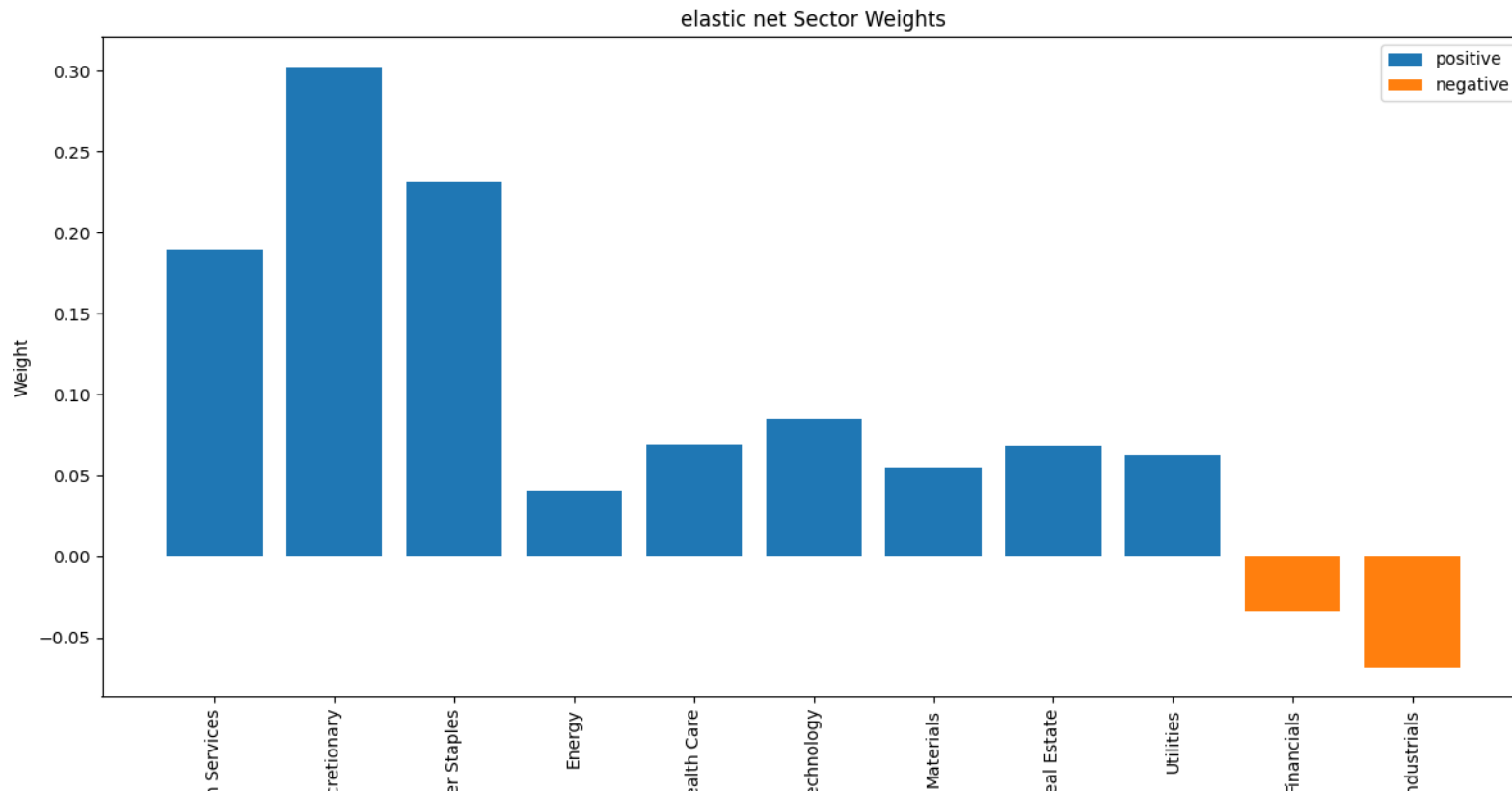
Sector weights: equally weighted



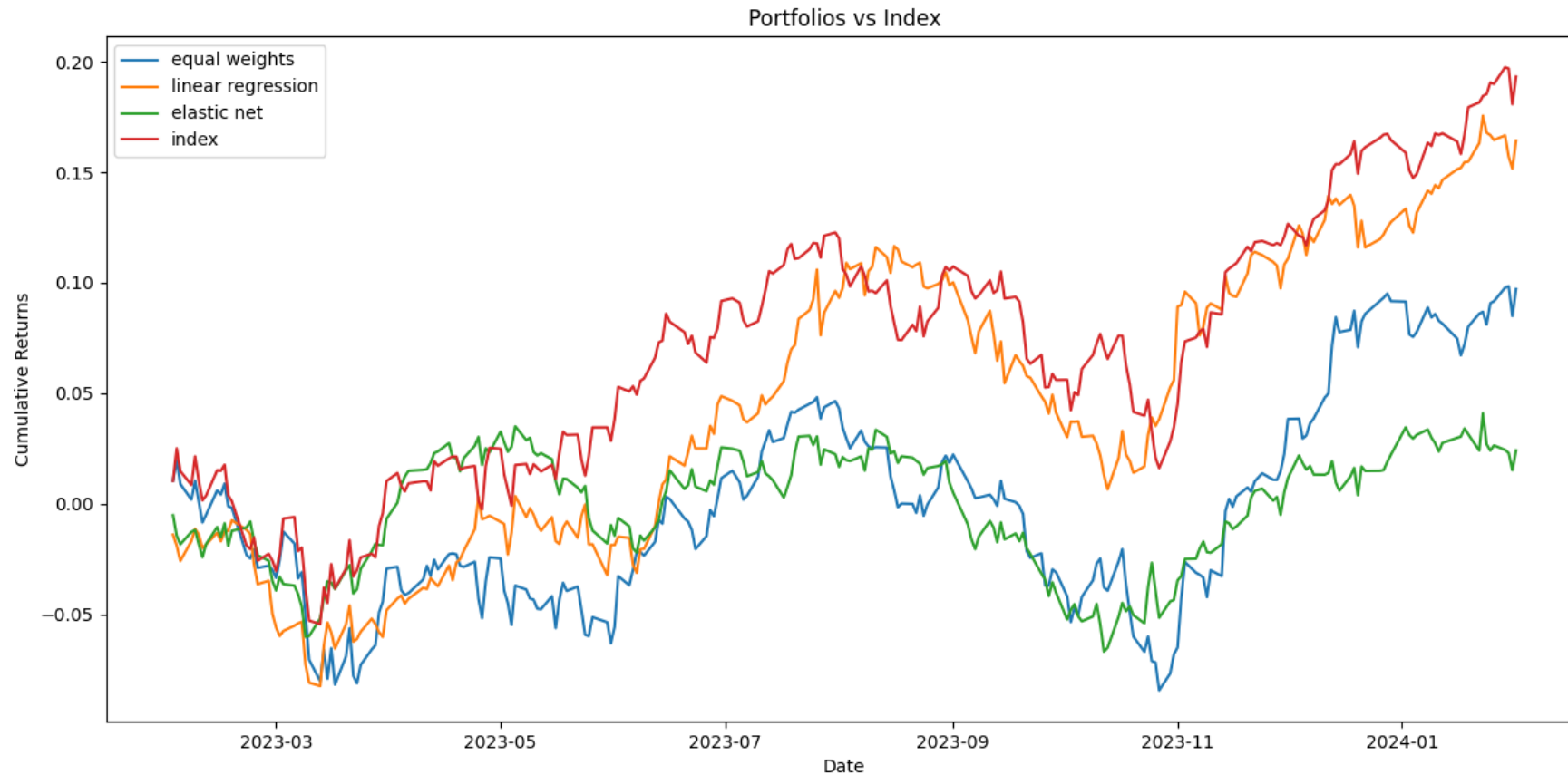
Cumulative returns November 21-22



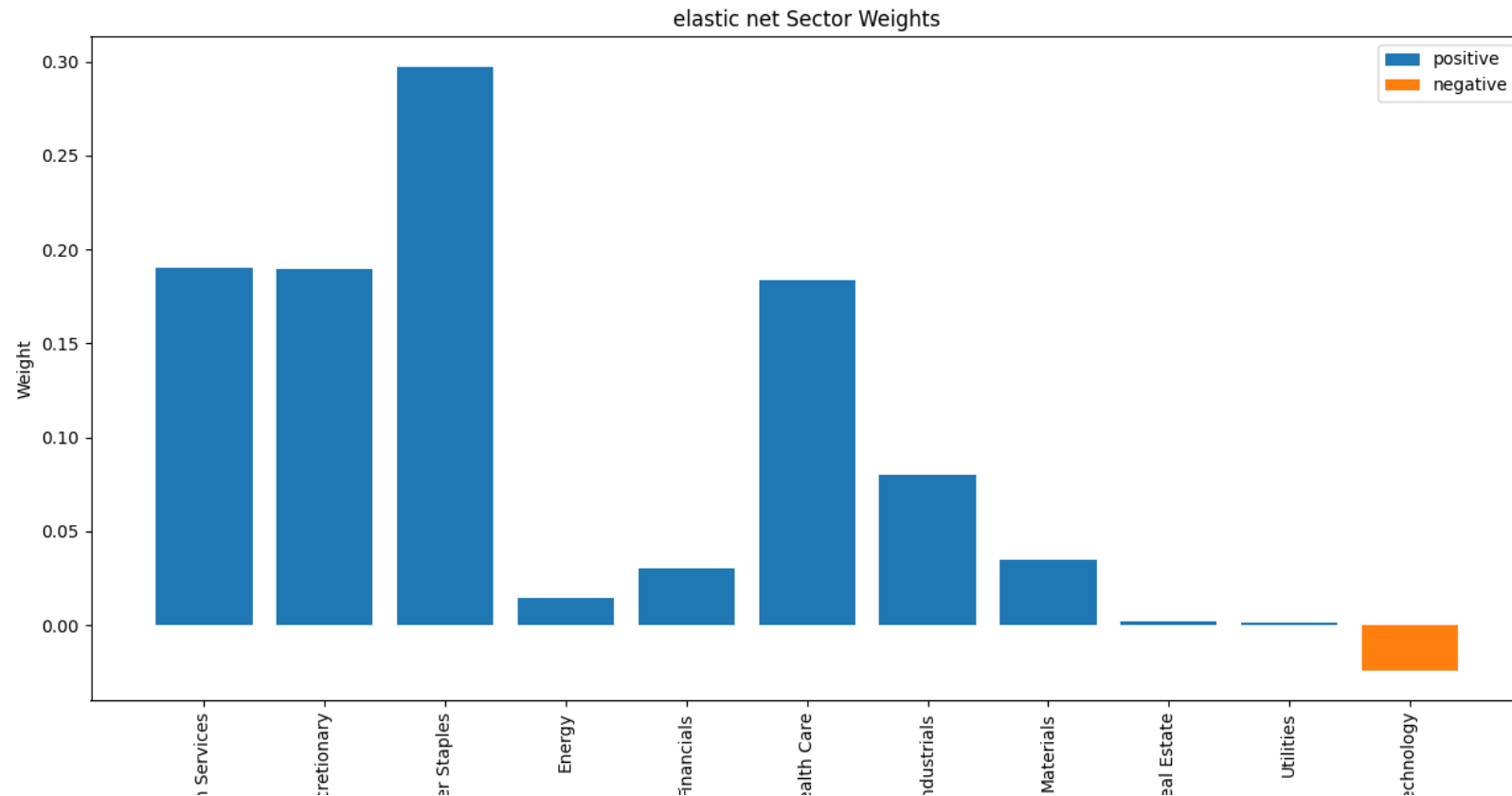
Elastic net weights November 21-22



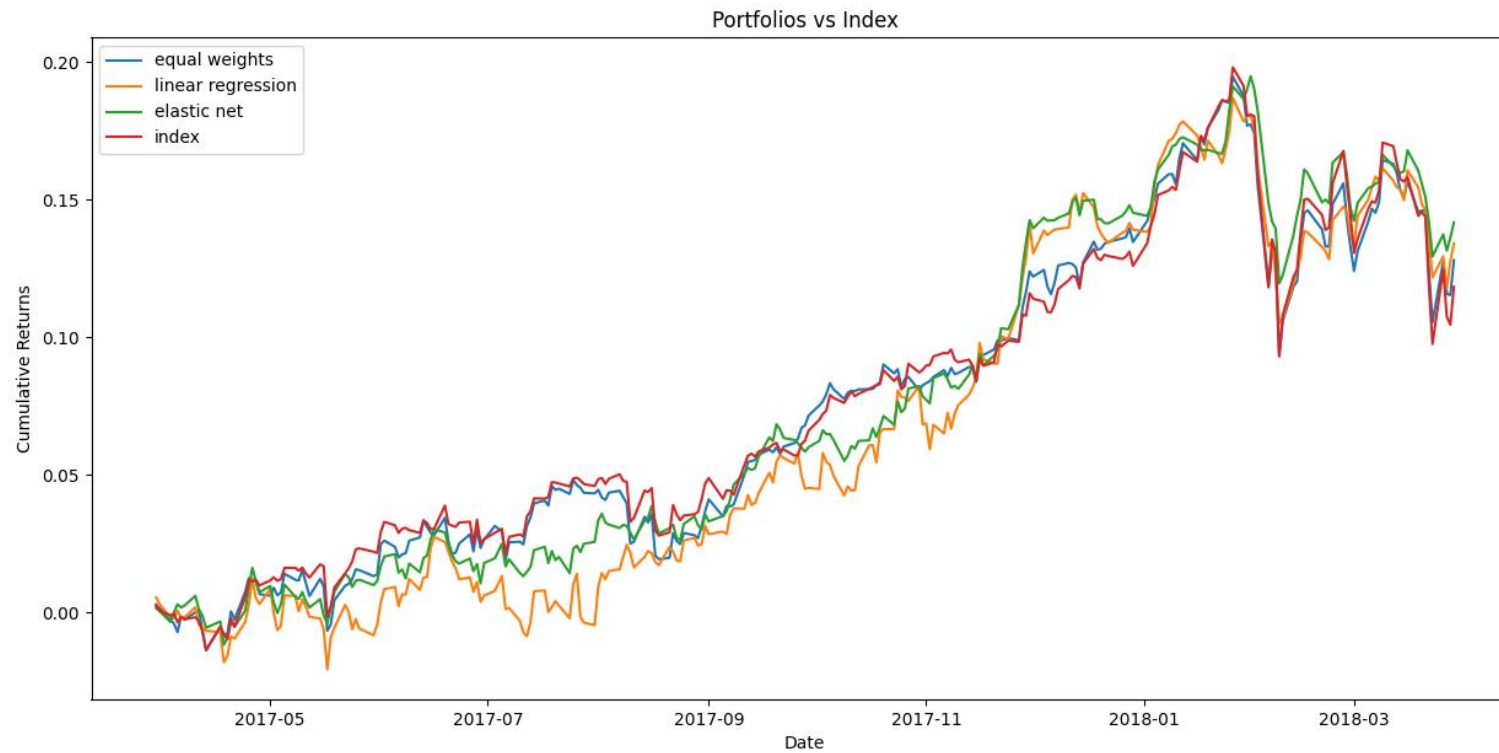
Cumulative returns February 23-24



Elastic net weights February 23-24



Sometimes models work



Takeaways

- Replicating (even better 😊) with fewer stocks the index is possible with a simple model: with more accurate data and more powerful models, replication can achieve a high level of performance. For this reason, financial companies may be interested in using this type of application.
- For the average person, the best way for replicate index remains to buy ETFs for at least two reasons:
 - The model halves the number of stocks, but they are still too large for retails
 - The ETF guarantees replication with reduced cost and effort

Drawbacks and extension

- Not so strong consideration of historical components (survival bias)
- Fixed components in the windows e no consideration of the period situation
- Non-random starting point: more rigid offsets (daily simulation)
- In cumulative graphs of each window, sometimes index and portfolio returns diverge greatly (adding nonnegativity constraints or limits to stock and sector weights can reduce this variability)
- Different combinations of training and testing intervals
- Conducting a more systematic and accurate analysis on the simulated portfolio

Thank you

