

Picking stocks by maximizing the Sharpe ratio

MGT 924 Statistical Foundations, Problem Set 3 | Kai Brusseleers, Yale SOM | October 2026

2.50

Gross Sharpe ratio of the final model over the scored period (December 1989 to November 2023). The best benchmark, IPCA, earns 1.95. My first model, a ridge regression, earned 1.33.

1. The idea in one paragraph

Most return models answer the question *which stocks will go up?* and then rank stocks on the answer. I found that a different question works much better: *which combination of trading signals has been the steadiest source of profit?* Each of the 402 stock characteristics in the data defines a simple long-short strategy. My final model, Portfolio-ML, picks the mix of these strategies with the highest past Sharpe ratio, shrinks it to avoid overfitting, and adds nonlinear versions of the signals. Using the same data, this raises the Sharpe ratio from 1.33 to 2.50.

Gross Sharpe ratio, 1990 to 2023

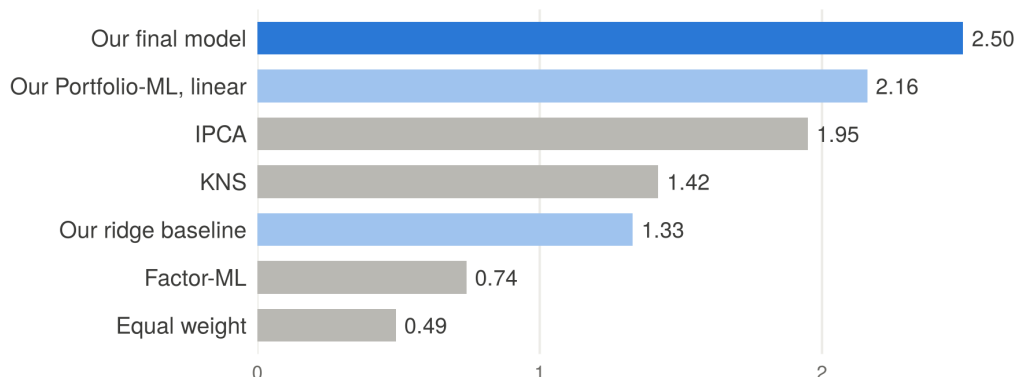


Figure 1. Gross annualized Sharpe ratio, 1990 to 2023. Grey bars are the benchmarks given in the assignment.

2. Data and how I avoided look-ahead

The data are 1,369,324 stock-months from WRDS (contrib_global_factor.ctff_chars), December 1951 to November 2023, with 402 characteristics and about 1,600 stocks per month. The target is each stock's excess return over the next month. I do not use the daily returns.

Every January I refit the model on all earlier months only. The newest return it sees is the one realized at the end of that January, which is exactly when the model is first used. So no weight ever depends on a return that was not yet known.

Period	Years	Used for
Training only	1952 to 1969	Fitting the first models
Validation	1970 to 1989	Comparing versions and choosing the final model
Scored	Dec 1989 to 2023	Reporting only, never used to choose anything

Table 1. Sample split. All periods use the same expanding window and yearly refit.

Characteristics come in different units, have extreme values and are often missing. Each month I therefore rank every characteristic across stocks and rescale the ranks to run from -0.5 to 0.5. A missing

value gets 0, the middle.

3. Three versions of the model

Version 1: ridge forecasts. I regress next-month return on the 402 ranks with a ridge penalty and use the forecasts as weights: demeaned within the month and scaled so the absolute weights sum to one. The penalty strength is chosen by cross-validation with whole years as folds. This model ranks stocks well, but it treats each characteristic as a separate bet. Many of them measure the same thing, for example several versions of value or momentum, so the portfolio stacks up the same few risks.

Version 2: Portfolio-ML. Following Jensen, Kelly, Malamud and Pedersen, I choose weights to maximize the Sharpe ratio directly. Each feature k gives a long-short portfolio with return $f_k = (\text{feature } k)(\text{returns}) / N$. Given T past months of these factor returns F , the model combines them with

$$\beta = (F'F / T + \lambda I)^{-1} \times (\text{average factor returns}), \quad w = Z\beta / N,$$

where Z holds this month's features and w is the weight of each stock. Without the penalty λ , β is the maximum-Sharpe mix of the factors (Britten-Jones, 1999). The term $F'F$ captures how the factors move together. That is exactly what version 1 ignores: if three signals always move together, the model treats them as one bet instead of three. Each January I choose λ by five-fold cross-validation over blocks of past years and keep the value with the highest out-of-fold Sharpe ratio.

Version 3: random features. Versions 1 and 2 can only use each signal in a straight line. Following Kelly, Malamud and Zhou (2024), I add 3,000 random mixes of the 402 ranks, passed through sin and cos. This lets the model pick up curves and interactions, for example a signal that only works in small firms. The random mixes are drawn once with a fixed seed. With more factors than months, I solve the same formula in an equivalent form that only needs a months-by-months matrix, which keeps the yearly refit fast.

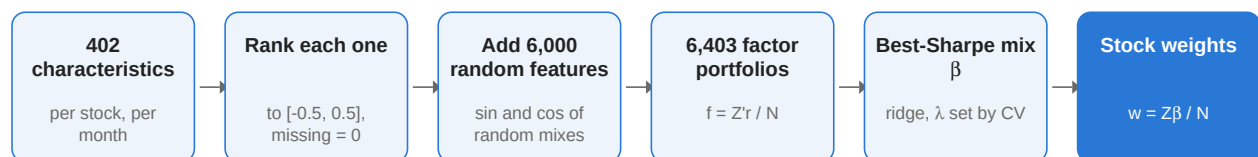


Figure 2. How the final model turns characteristics into portfolio weights each month.

	Version 1: Ridge	Version 2: Portfolio-ML	Version 3: + random features
Question	Which stocks go up?	Which mix of signals is steadiest?	Same, with curves
Learned parameters	402 coefficients	403 weights β	6,403 weights β
Hyperparameters	λ	λ	λ , number of features, γ
Sees overlap between signals	No	Yes	Yes

Table 2. The three versions side by side. λ is the ridge penalty, γ the scale of the random mixes.

4. Choosing the final model

I compared six versions on the validation period (Table 3). Each step helps, and the ranking is the same in the scored period, which suggests the gains are not luck. Moving to Portfolio-ML adds the most. Random features add about 0.3 more. The scale γ hardly matters, so I kept $\gamma = 1$, the value I fixed before running anything, and chose 3,000 features because they had the best validation Sharpe ratio.

Version	Validation 1970 to 1989	Scored 1990 to 2023
Ridge forecasts	2.90	1.33
Portfolio-ML, linear	3.55	2.16
Portfolio-ML, 1,000 random features, $\gamma = 0.5$	4.00	2.43
Portfolio-ML, 1,000 random features, $\gamma = 1$	4.03	2.46
Portfolio-ML, 1,000 random features, $\gamma = 2$	4.06	2.38
Portfolio-ML, 3,000 random features (final)	4.14	2.50

Table 3. Annualized gross Sharpe ratios of every version I ran.

5. Results

The weights have no natural scale, and scaling them does not change the Sharpe ratio, so I report returns at 10% annual volatility. At that level the final model earns 25% a year. Its worst month is -8.8%, its largest drawdown 12.6%, and 81% of months are positive.

Cumulative excess return, scaled to 10% volatility (%)

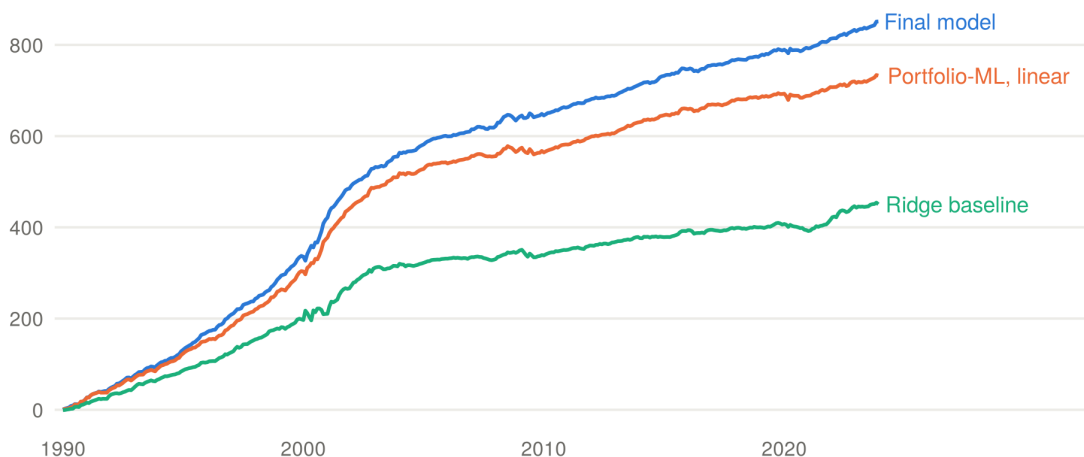
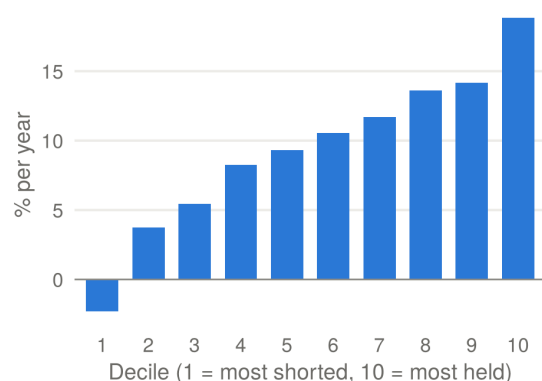


Figure 3. Cumulative excess return of three versions over the scored period, each scaled to 10% annual volatility.

Two checks show the result is broad, not driven by a few stocks. Sorting stocks each month into ten groups by their weight, returns rise almost step by step from -2.3% a year in the most-shortest group to 18.8% in the most-held group (Figure 4, left). A regression of next-month return on the standardized weight, with standard errors clustered by month, has a t-statistic of 12.9: a stock one standard deviation higher in the ranking earns about 0.5 percentage points more the next month.

Performance has fallen over time, from 4.73 in the 1990s to 2.2 to 2.5 in the 2000s and 2010s and 1.68 since 2020. All three versions show this, so it comes from the signals, not the model. It matches the finding that anomalies weaken once they are published (McLean and Pontiff, 2016).

Return by decile of portfolio weight



Sharpe ratio by decade

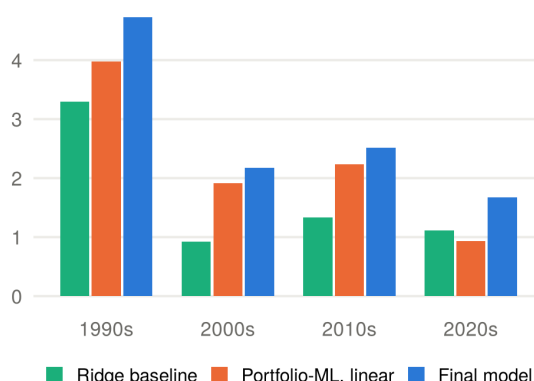


Figure 4. Left: average annualized excess return of ten groups sorted on the final model's weight. Right: Sharpe ratio by decade (2020s = 2020 to 2023).

6. Limits, and would I invest?

Limit	Why it matters
No trading costs	The score is gross. The model trades many stocks every month; costs would cut the Sharpe ratio a lot.
Small stocks	Every stock gets equal say in the signals, including micro-caps, which are the hardest and costliest to trade.
Exposure moves around	Total long and short size changes month to month. A real fund would target a fixed volatility.
Decay	The Sharpe ratio has trended down; the future likely looks more like the 2020s than the 1990s.
Model search	Six versions were compared. Choosing on 1970 to 1989 limits the damage, but small gaps between versions are noise.

Would I invest in it as it stands? No. I would first add a trading cost penalty, which Jensen, Kelly, Malamud and Pedersen show is needed for an implementable version, and drop the smallest stocks. The main lesson I take away is that accounting for how signals overlap added more than better return forecasts did.

7. Implementation

The submission is one R file that uses base R only, so the lock file pins R 4.4.2 and no packages. The random mixes are drawn with a fixed seed inside main, so runs are identical. I checked that removing later data leaves earlier weights unchanged and that the code runs on a 123-month sample.

References

- Britten-Jones, M. (1999). The sampling error in estimates of mean-variance efficient portfolio weights. *Journal of Finance* 54(2), 655-671.
- Jensen, T. I., Kelly, B., Malamud, S. and Pedersen, L. H. Machine learning and the implementable efficient frontier. Working paper.
- Kelly, B., Malamud, S. and Zhou, K. (2024). The virtue of complexity in return prediction. *Journal of Finance* 79(1), 459-503.
- McLean, R. D. and Pontiff, J. (2016). Does academic research destroy stock return predictability? *Journal of Finance* 71(1), 5-32.