Predicting mutual fund returns using machine learning tools
Machine learning final project

László Jakab

December 10, 2015

1 Introduction

A typical finding in the finance literature is that while mutual fund investors chase past returns, flows fail to predict future performance. In general, past performance is not a strong predictor of future performance. A number of interpretations have been proposed. Perhaps it is an equilibrium phenomenon: as capital flows to talented, high-performing fund managers, their performance suffers due to decreasing returns to scale [Berk and Green (2004), Berk and van Binsbergen (2014)]. Essentially, large funds' trades move prices, decreasing the profitability of trades. Similarly, the profitability of potential investments will decrease if competing funds move prices by engaging in similar trades. There is empirical evidence of a negative relation between risk-adjusted returns and fund size — both own size, and the size of competing funds [for example, Pástor, Stambaugh and Taylor (2015), Jakab (2015)]. In general, if capital is slow-moving to talented managers, as learning about skill from noisy returns is slow, performance might persist, and returns might be predictable to some extent.

The aim of academic finance research is typically inference, not prediction. To the extent that academic researchers engage in prediction in this sphere, it is generally based on relatively primitive portfolio sort methods. In this approach, we sort funds into portfolios based on observable characteristics associated with returns, and see how well the portfolios do in the future.

In this project I attempt to answer the following questions. How well can we predict returns on actively managed mutual funds using accepted machine learning tools? What is the relative importance of different features, such as past returns, fund size, competitor size, and other fund characteristics? Can we construct portfolios that do better out of sample than randomly constructed portfolios or portfolios based on predictions from linear regression? Fund returns are extremely noisy, so building a model with any out of sample predictive capacity would be an achievement.

My study is based on a monthly panel of actively managed domestic equity funds. I assign a random subset of months to the "test" set; the train set consists of data from the remaining months. I build models on the train data, and predict on test. In particular, I compare the out of sample predictive performance of a simple linear regression with the performance of random forest and boosted tree models whose tuning parameters are chosen by cross-validation. Beside comparing the out of sample RMSE of the models and the correlation between predicted and actual returns, I also examine whether allocating resources to the funds with the highest predicted returns in fact increases portfolio performance.

I find that by conventional measures, the models are a failure: out of sample RMSE is high, and the correlation between actual and predicted returns is close to zero. *Nonetheless*, the models pick out enough signal to be potentially useful for informing investment decisions. In particular, I find that a strategy of investing in the fund with the highest predicted return each month outperforms picking funds randomly, often by a significant margin. For example, over the course of the test set, investing according to the boosted trees model outperforms (on a risk-adjusted basis) over 80% of simulated portfolios where funds are chosen randomly.

I find that in this context the machine learning models are not obviously superior to a simple

linear regression model. I can see a couple of reasons for the relatively favorable performance of the linear model. First, as in the marketing dataset we saw during the course (where logistic regression did surprisingly well), the variables in my dataset have appeared in a number of academic studies over time. Linear models are a popular modeling choice in these circles, and hence the variables are likely to have been selected to perform well in these models. Second, returns are very noisy, which might favor the simple, comparatively high bias/low variance technology of linear regression over more sophisticated models that might give more wiggle room for overfitting.

The rest of the paper is organized as follows. Section 2 discusses the source data, Section 3 describes the methods for model building and evaluation, and Section 4 presents results. Section 5 briefly concludes.

2 Data

From a previous project [Jakab (2015)],[1] I have a dataset covering actively managed domestic equity mutual funds over the period of 1980-2011. The original construction of this dataset was primarily based on (i) the CRSP Survivor-Bias-Free US Mutual Fund database which includes monthly information on fund returns, size, age, etc, and (ii) the Thompson Reuters S12 database, which includes detailed, generally quarterly, information on fund holdings. I linked the CRSP mutual fund data to the Thompson holdings data using MFLINKS, initially developed by Wermers (2000). These two main sources were supplemented by security-level data on prices and shares outstanding from CRSP, and monthly return factors from Ken French's data library.[2]

The resulting dataset is at the fund-month level. Variables included are returns (both raw and risk adjusted), expense ratios, fund size (TNA), turnover, fund age, the typical market cap of holdings, measures of portfolio concentration, a proxy for the overall size of the actively managed mutual fund industry, as well as a proxy for the size of competitor funds. For each fund, this proxy for the size of competitor funds was calculated as the weighted sum of the size of other funds, where the weights were equal to the cosine similarity between the funds' portfolio weights. [3]

For the present analysis, I drop turnover (it has some missing values and is unlikely to be a

[1] While I have used this dataset in previous research for inference, I have not tried to build prediction models before. Therefore, while the out of sample exercise will not be as pure as if I had never seen the dataset (and would begin by setting aside a test dataset sight unseen), I do not know what models would do well when it comes to cross-validation or out of sample testing. In this sense, I am not particularly concerned about data snooping, and I think that re-using the dataset for the project is an acceptable compromise.

[2] `http://mba.tuck.dartmouth.edu/pages/faculty/ken.french/data_library.html`

[3] In terms of equations, the competitor size proxy is defined as

$$CompetitorSize_{i,t} = \sum_{j \neq i} \underbrace{\psi_{i,j,t}}_{\text{cosine similarity bw fund i and j}} \underbrace{\frac{TNA_{j,t}}{TotalMktCap_t}}_{\text{size of competitor } j},$$

where $\psi_{i,j,t}$ is the cosine similarity of funds' lagged portfolio weights; the portfolio weight vector is denoted $\mathbf{w}$

$$\psi_{i,j,t} = \frac{\mathbf{w}_{i,t-k} \cdot \mathbf{w}_{j,t-k}}{\|\mathbf{w}_{i,t-k}\| \|\mathbf{w}_{j,t-k}\|}$$

strong predictor of returns), and observations with any missing values. Observations with missing values constitute a small fraction of the overall dataset. After these steps, I am left with 383,980 observations, spanning 376 months. The complete dataset includes 3,575 unique funds.

There is considerable cross-sectional correlation in (risk-adjusted) fund returns, but only limited within-fund correlation [e.g. Pástor, Stambaugh and Taylor (2015)]. Since observations within month are not independent, including observations from the same month in both the training and the test sample would lead to underestimating the out of sample error rate. Therefore, instead of randomly allocating individual *observations* to the train and test samples, I instead allocate individual *months* to the train and test samples.[4] Besides alleviating the concern of underestimating the test error, sampling months instead of observations allows me to evaluate the usefulness of the predictive models in an intuitive way through the construction of hypothetical portfolios, as discussed in the next section. I allocate a random 25% of months to the test set. The final training set consists of 287,425 observations, and the test set 96,555 observations.

3 Methods

I consider the following potential predictor variables in all models:

- `r_mean_1yr`: Mean of past one year of raw returns
- `r_sd_1yr`: Standard deviation of past one year of raw returns
- `IndustrySize`: The aggregate size of the actively managed fund industry, as a fraction of total market capitalization of all stocks.
- `CompetitorSize`: The size of the fund's competitors, relative to total market cap. As defined earlier.
- `tna` Fund size
- `logtna` Log fund size
- `fund_age` Age of fund, in years
- `stock_cap` Typical size of the stocks held by the fund (i.e. small stock vs large stock funds), relative to total market cap
- `hhi` Herfindahl index of the fund's portfolio holdings; measure of portfolio concentration

For models predicting risk-adjusted returns, I also add `ff_r_mean_1yr` and `ff_r_sd_1yr`, the mean and standard deviation of the last one year of risk-adjusted returns, respectively.

[4] To keep the analysis simple, I employ the usual k-fold cross-validation methods on the training sample. Nonetheless, I believe that in this environment the sampling for the folds should take into account the within-month dependency in the data by splitting the training set into random folds by randomly allocating observations belonging to different months to different folds.

To establish a simple baseline, I first produce, using the entire training sample, a linear regression model of the form

$$r_{i,t+1} = \alpha + \mathbf{X}'_{i,t}\mathbf{\Gamma} + \varepsilon_{i,t}, \tag{1}$$

where i indexes fund, t month, and $\mathbf{X}_{i,t}$ is a vector of variables including the appropriate predictors.

Next, I build random forest and boosted trees models. For tuning the parameters, I use five-fold cross-validation, repeated six times, and in each case choose the model with the lowest cross-validated RMSE. I estimate 500 trees in each case. For boosted trees, I search over shrinkage parameters $\{0.01, 0.1, 0.2\}$, and interaction levels $\{1, 2, 3\}$. Considering the large sample size and the potential for overfitting, I set the minimum node size at a relatively large 10,000.

I use the models for performing out of sample predictions on the test set. I examine the out of sample RMSE and the correlation between predicted and actual returns. I then investigate whether the information from the predictions is actionable for investing. Each month in the test set, I pick the fund with the highest predicted returns, and cumulate the returns for this hypothetical portfolio across the months in the test set, ordering months chronologically, as if they were consecutive months. I also compute descriptive statistics of the returns on this portfolio. To produce a baseline comparison for the cumulative returns, I simulate 100 portfolios by randomly picking funds each month. As a robustness check, I evaluate the performance of investment strategies where each month I construct equally weighted portfolios of the five funds that are predicted to have the highest returns next month (as opposed to just picking the single fund with the highest predicted return). For these strategies, the comparison group consists of a hundred simulated portfolios, each constructed by equally weighting five randomly picked funds each month.

4 Results

Cross-validation picks random forest tuning parameter $m = 9$ for modeling raw returns, and $m = 11$ for modeling risk adjusted returns. For boosted trees, shrinkage parameter 0.2 and interaction level 3 are chosen for both raw and risk adjusted returns.

Since they are arguably the easiest to interpret, let's begin by examining the linear regression models estimated on the training set, and presented in Table 1. [5] Clearly, past returns (both mean and standard deviation) feature prominently in the linear models. IndustrySize is also important for the raw return model, and CompetitorSize, as well as hhi, might carry some weight too. Although not reported in the interest of space, variable importance in the machine learning models is strongly skewed toward past returns and IndustrySize. As demonstrated by the extremely low R^2, there is a lot more noise than signal in this environment: the linear model only explains about half a percent of the in-sample variation in next month's returns.

Table 2 reports conventional measures of the various models' out of sample predictive perfor-

[5] Note that the reported t-statistics are based on standard errors uncorrected for cross-sectional correlation in the errors, and therefore massively overstate the statistical significance of the estimated coefficients. However, since I only care about out of sample prediction here, I didn't bother correcting the standard errors.

Table 1: Linear regression models for raw and risk adjusted returns

	Raw		Risk adjusted	
	Estimate	t value	Estimate	t value
(Intercept)	7.17E-01	10.938	4.63E-02	1.584
ff_r_mean_1yr			1.82E-01	26.576
ff_r_sd_1yr			-3.75E-02	-8.684
r_mean_1yr	3.05E-02	4.828	7.38E-03	2.307
r_sd_1yr	1.40E-01	31.201	2.80E-03	1.12
fund_age	-1.97E-03	-2.349	1.09E-03	2.991
tna	2.03E-06	0.769	6.67E-07	0.581
logtna	-2.56E-02	-3.078	-9.06E-03	-2.496
IndustrySize	-5.84E+00	-17.841	2.71E-01	1.888
CompetitorSize	-4.56E+00	-2.801	-4.22E+00	-5.837
stock_cap	1.93E-01	1.897	3.33E-02	0.749
hhi	-7.83E-01	-6.718	-2.08E-01	-4.049
R^2	0.005		0.004	

mance. The models fail miserably by these measures. In the test set, the unconditional standard deviation of raw returns and risk adjusted returns are 5.72 and 2.55, respectively. The RMSE's of the models are actually higher! Essentially, we would do better by just predicting the mean. Correlations between predictions and actual out of sample outcomes are only slightly less bleak. In two of the six cases, the correlations are negative. For risk-adjusted returns, they are very close to zero. Only random forest does well for raw returns — it fails for risk-adjusted returns, however. For a more detailed look, Figure 1 presents scatter plots of actual versus predicted returns. There is no obvious upward sloping tendency. The random forest predictions are a bit clumpy, which might have been helped by increasing the size of the forest or decreasing the minimum node size. On the other hand, going from 100 to 500 trees did not help much in my experience, and decreasing minimum node size might have encouraged overfitting — not to mention that both would have increased computation time.

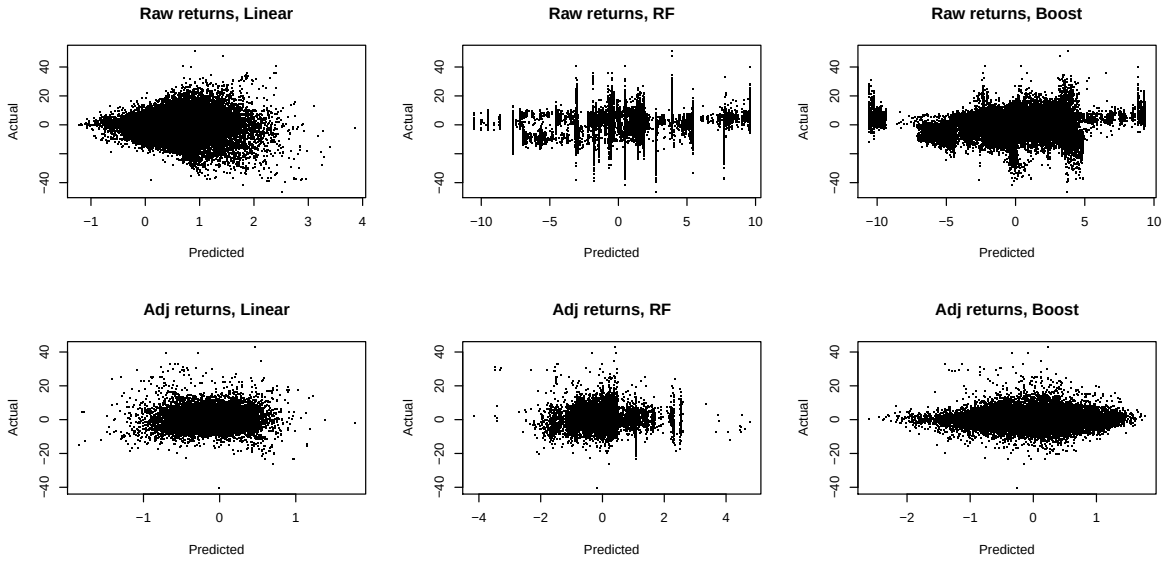
Table 2: Evaluating model performance out of sample

RMSE			
	linear regression	random forest	boosting
raw returns	5.74	6.19	6.2
risk adjusted	2.55	2.57	2.58

Correlation between actual and predicted			
	linear regression	random forest	boosting
raw returns	-0.0043	0.103	0.0627
risk adjusted	0.0228	-0.001	0.0052

It appears that the models are an utter failure for predicting typical fund returns. Can they nonetheless have any practical use? After all, for a successful investment strategy all we really need is to pick out a couple of funds that tend to perform better than the average fund. Motivated by

Figure 1: Actual versus predicted out of sample returns



this observation, I evaluate the profitability of an investment strategy where each month I pick the fund with the highest predicted returns according to the model at hand. [6] The left panel of Table 3 presents descriptive statistics of the returns of such investment strategies. In the right panel, I first subtract the return on the median fund each month to establish a direct comparison between the returns of this strategy relative to simply investing in the typical fund. These investment strategies are fairly lucrative. The mean returns relative to the median fund are positive across the board. Even more impressively, the linear and the boosted tree models produce about 3% mean annualized risk adjusted returns! On the downside, the realized returns on these strategies are very volatile. For instance, the interquartile ranges of the strategies' returns tend very similar the unconditional interquartile range of returns in the test set. At any rate, it does appear that using the models to pick funds shifts the return distribution to the right by a small amount.

Another natural extension of this thought experiment is an examination of how well the strategy would do over longer periods of time. Since risk-adjusted returns are reasonably uncorrelated across time, we can construct a hypothetical time series of returns from the test set by ordering the months in the test set in chronological order and pretending that they are in fact consecutive. [7] Using this hypothetical time series of returns, I can compute the cumulative returns on the investment strategy of picking the best fund according to the predictive models. In order to establish a null, I

[6] My analysis ignores transaction costs. The strategy of picking the fund with the best predicted returns each month requires monthly rebalancing, transaction costs are likely to be large in practice. Regardless, I find the thought exercise illuminating.

[7] The temporal independence assumption is really only reasonable for risk-adjusted returns. This fact is evident in the plots of the simulated raw returns, which bear the tell-tale signs of market crashes and subsequent recoveries, whereas the simulated risk adjusted return series resemble random walks. Raw returns partially reflect time-varying, somewhat predictably mean-reverting risk premia. Interrupting the time-series structure of raw returns by pretending that non-successive months are adjacent in this manner is somewhat questionable practice. Risk-adjusted returns filter out compensation for bearing systemic risk (market risk, etc).

Table 3: Descriptive statistics for realized returns of funds predicted to have the highest performance next month. In the right panel, I subtract the median fund return each month for a more informative comparison. Returns are in monthly % units.

Raw returns						
				Relative to month median		
	linear	RF	boost	linear	RF	boost
Min.	-29.16	-28.88	-15.1	-21.68	-5.281	-13.48
1st Qu.	-1.68	-2.27	-2	-1.081	-1.207	-1
Median	1.24	1.26	1.23	0.022	-0.042	0.21
Mean	0.74	0.96	0.83	0.058	0.273	0.15
3rd Qu.	4.47	4.46	3.8	1.413	1.607	1.39
Max.	16.54	13.43	11.19	11.05	6.825	13.87

Risk adjusted returns						
				Relative to month median		
	linear	RF	boost	linear	RF	boost
Min.	-4.97	-8.052	-3.8	-5.6	-7.763	-4.047
1st Qu.	-0.6	-0.976	-0.85	-0.72	-1.01	-0.809
Median	0.14	-0.141	0.13	0.31	0.0759	0.047
Mean	0.24	-0.018	0.26	0.26	0.0094	0.283
3rd Qu.	1.19	1.16	1.44	1.24	0.9946	1.264
Max.	7.56	9.761	6.91	6.43	9.385	7.612

also simulate a hundred return series by picking a fund at random each month.

Figure 2 illustrates the results. The top half plots cumulative returns over the course of the synthetic time series provided by the test set. As we can see, raw returns exhibit some temporal dependence (see footnote 7), whereas risk-adjusted returns are closer to a random walk. The top half also shows that the investment strategies do quite well over time. It is not impossible that the good performance is due to luck (a number of simulated strategies do equally well), but the returns from the investment strategies tend to be one of the better performing portfolios. To draw this fact into sharper focus, the bottom half of Figure 2 plots the proportion of simulated return series that do worse than each investment strategy at each point in time. The investment strategies tend to be above the median, especially for risk-adjusted returns. For risk-adjusted returns, the linear model is consistently above the median, and by the end of the sample it is one of the best performing portfolios. Interestingly, the performance of the linear model relative to the machine learning models is better for risk-adjusted returns. One reason might be that in the context of mutual funds, academics primarily study risk-adjusted returns (which are, after all, what we really care about), not raw returns, and so the variables are already essentially pre-processed for use in linear models predicting risk-adjusted returns. The other reason might be that raw returns provide more structure for the relatively sophisticated models to latch on to, whereas noisy risk adjusted returns favor simpler models with less freedom for overfitting.

Out of concern that the above relatively good results might just be lucky, I re-do the exercise

by picking the top five (as opposed to just one) funds each month, and stacking them into an equally weighted portfolio. For the baseline simulations, I pick five funds at random each month, and construct an equally weighted portfolio, repeated one hundred times. The results are shown in Figure 3. The portfolios do worse for raw returns. Since raw returns partially reflect compensation for risk, risk-adjusted returns are much more interesting. Here, the models do quite well. Again, the linear and boosting models are consistent good performers. In fact, boosting is overall beat by only one simulated return series. Random forest takes a dive halfway through, but recovers by the end.

5 Conclusion

My project studies the out of sample predictability of actively managed mutual fund returns using conventional linear regression methods, as well as sophisticated boosting and random forest machine learning tools that rely on cross-validation for tuning model parameters. I find that the performance of all considered methods is poor for predicting returns across all funds. However, in the extremes, the models carry information relevant for investment decisions: in any given month, the fund predicted to have the highest returns typically performs better than a randomly chosen fund. However, substantial uncertainty remains, as the performance of even the supposedly best fund is highly volatile month-to-month. My study also ignores transaction costs, which I leave for future research.

Figure 2: Cumulative returns of picking fund with highest predicted returns each month

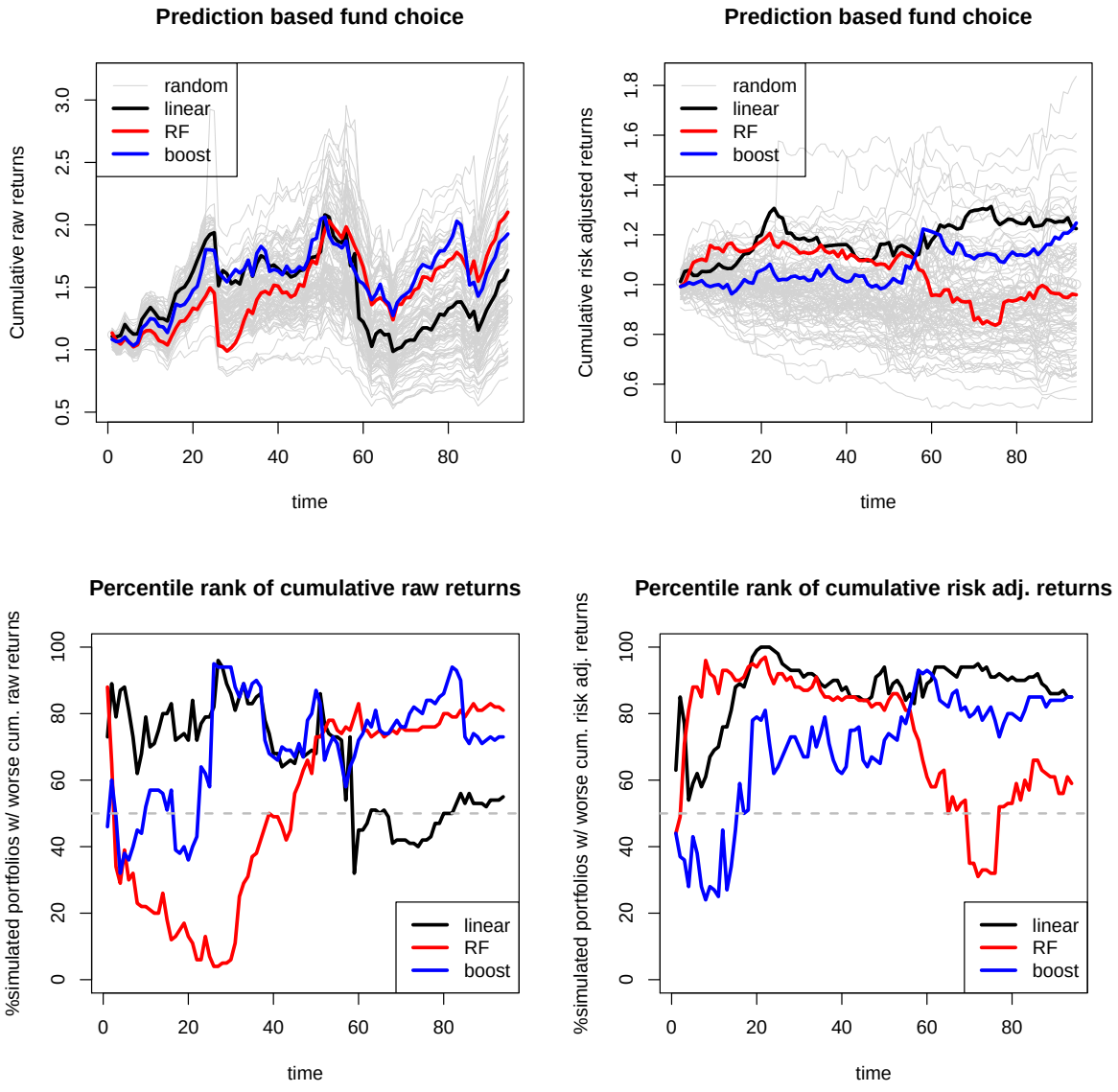
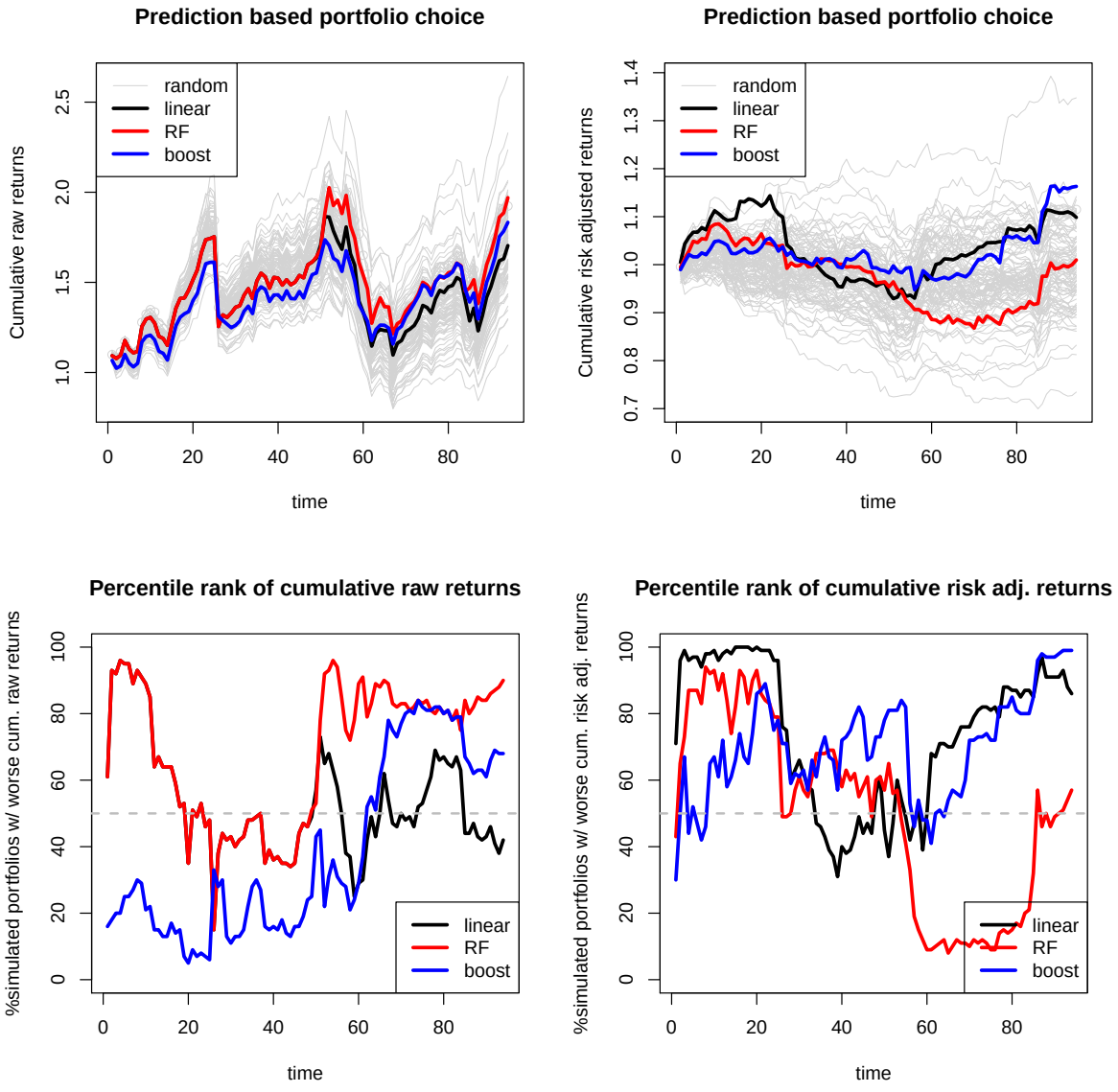


Figure 3: Cumulative returns of picking fund with highest predicted returns each month



References

[1] Berk, Jonathan B. and Richard C. Green. 2004. "Mutual Fund Flows and Performance in Rational Markets." *Journal of Political Economy*, 112(6): 1269–95.

[2] Berk, Jonathan B. and Jules H. van Binsbergen. 2013. "Measuring Skill in the Mutual Fund Industry." *NBER Working Paper 18184*.

[3] Jakab, László. 2015. "The Effect of Industry Size on Mutual Fund Returns: Evidence From Fund Holdings." *University of Chicago Booth School of Business Finance Ph.D. Curriculum Paper*.

[4] Pástor, Ľuboš, Robert F. Stambaugh and Lucian A. Taylor. "Scale and Skill in Active Management." 2014. Forthcoming, *Journal of Financial Economics*.

[5] Wermers, Russ. 2000. "Mutual Fund Performance: An Empirical Decomposition into Stock-Picking Talent, Style, Transaction Costs, and Expenses." *Journal of Finance*, 55(4): 1655–95.