Multi-Factor Portfolio Construction

Karim Layoun and Pranav Mehta

Georgia Institute of Technology

ABSTRACT

We explore the use of factor models to construct equal-weighted portfolios from assets in the Russell 1000 Index.

Keywords: Russell 1000, Factor Models

INTRODUCTION

The input sensitivity of mean-variance models have given rise to quantitative portfolio construction methods that focus on managing risk. In the absence of constraints, the minimum-risk portfolio is the solution to the following model:

$$\begin{aligned} \min_{x} \quad & \mathbf{x^T V x} \\ \text{s.t.} \quad & \mathbf{1^T x} = 1 \end{aligned}$$

When $\mathbf{V} = \mathbf{I}$, we obtain an equally weighted portfolio, as the optimal solution

$$\mathbf{x}^* = \frac{1}{\mathbf{1^T V^{-1} 1}} \mathbf{V^{-1} 1} = \frac{\mathbf{1}}{N}$$

In theory, such a portfolio should have superior risk-adjusted returns.

We propose to use 10-factor models to constrain the number of stocks in the constructed portfolios such that we only select assets with predicted next-month returns above a certain threshold. We use stocks in the Russell 1000 as our universe of assets, and select 10 variables (listed below) from a set of more than 20 factors, to build the prediction models. We form and evaluate portfolios, from Dec-2004 to Nov-2018, in four steps: data wrangling, monthly return prediction, monthly portfolio construction, and performance evaluation.

DATA

Input

We use 3 datasets to construct equal weighted portfolios: Russell 1000 Index monthly returns, stock monthly returns, and factor data.

Russell 1000 Benchmark Data

Benchmark data represents index monthly returns from Jan-2003 to Sept-2019, and did not require any wrangling.

DATE	MSCI EM Bench Return	Russell 1000 Bench Return	MSCI ACWIXUS Bench Return
2003-01-01	-0.033226	-0.056599	-0.032306
2003-02-01	-0.004354	-0.024234	-0.035107
2003-03-01	-0.026990	-0.015472	-0.020262
2003-04-01	-0.028355	0.010351	-0.019398
2003-05-01	0.089072	0.080728	0.096360

Figure 1. Russell 1000 Monthly Returns

Stock Returns

Stock data represents monthly returns for a universe of 41747 stocks, identified by the first 6 symbols of their respective SEDOLS, from Jan-1995 to Dec-2019. The data required some wrangling, as some stocks are not listed for the length of the data date range. We simply forward-filled null values, avoiding any look-ahead bias, and replaced the remaining nulls with 0.

DATE	200001	200169	200230	200247	200291	200369	200418	200582	200597	200784	...
1995-01-01	0.0	0.00332	0.00482	-0.00304	0.00432	0.0	0.0	0.0	0.00037	0.00848	...
1995-02-01	0.0	0.00346	0.00251	0.00436	0.00078	0.0	0.0	0.0	0.00295	-0.00040	...
1995-03-01	0.0	0.00307	-0.00149	-0.00564	-0.00358	0.0	0.0	0.0	0.00586	0.00794	...
1995-04-01	0.0	0.00033	0.00186	-0.00351	0.00011	0.0	0.0	0.0	0.00556	0.00044	...
1995-05-01	0.0	-0.00093	-0.00036	-0.00048	-0.00014	0.0	0.0	0.0	-0.00298	-0.00889	...

Figure 2. Stock Monthly Returns

Factor Data

Factor data required more data wrangling, as some factors were not available for the totality of the data date range, and it is essential we avoid any information leak to the training sets, including look-ahead bias. And so, we forward-filled null values and replaced the remaining nulls with 0. Moreover, when standardizing data for model training/prediction, we fit the ”standardizer” to the training set, before transforming both the train and test sets.

We selected a subset of the available columns/factors, a mix fundamental and return momentum indicators, to predict next month returns (TARGET in Fig. 3):

RCP: $\frac{\text{current CF/P}}{\text{5 year avg CF/P}}$

RBP: $\frac{\text{current Book/P}}{\text{5 year avg Book/P}}$

RSP: $\frac{\text{current Sales/P}}{\text{5 year avg Sales/P}}$

REP: $\frac{\text{current E/P}}{\text{5 year avg E/P}}$

RDP: Relative dividend yield

RPM71: Reverse price momentum of month-7 price divided by month-1 price

RSTDEV: Standard deviation of the previous 12-month returns

ROA1: One year return on assets

9MFR: Return forecast by Mckinley 9-factor model

8MFR: Return forecast by Mckinley 9-factor model

		RCP	RBP	RSP	REP	RDP	RPM71	RSTDEV	ROA1	9MFR	8MFR	TARGET
DATE	SEDOL											
2003-01-01	200001	6.0	1.0	28.0	8.0	39.0	99.0	11.0	2.0	24.6395	6.1009	97.0
	200169	97.0	100.0	100.0	1.0	39.0	2.0	6.0	5.0	38.4409	47.6129	1.0
	200230	28.0	12.0	17.0	30.0	66.0	66.0	51.0	79.0	32.8892	26.0335	42.0
	200247	100.0	100.0	98.0	100.0	39.0	2.0	1.0	33.0	80.2576	100.0000	97.0
	200418	35.0	17.0	43.0	76.0	39.0	44.0	20.0	99.0	55.2264	52.9502	100.0

Figure 3. Filtered and Cleaned Factor Data

PREDICTION PIPELINE

The pipeline includes all steps required for accurate prediction of next-month returns to construct equal-weighted portfolios.

Model Description

We have chosen 5 models, leveraging different supervised learning methods, to predict next-month returns, where the independent variables are the 10-factors in Fig. 3.

Linear Regression

More specifically, a multiple linear regression, which attempts to fit a linear function of the 10-factors to the data.
The resulting factor coefficients may be interpreted as a measure of factor importance, where the higher the coefficient the more significant the factor.

CTEF

We simply use the given Consensus EPS/I/B/E/S forecast, revisions and breadth given in the factor data as the prediction of next-month returns.

Decision Trees

A non-parametric supervised learning method that predicts the value of a target variable by learning simple decision rules inferred from the data features.

AdaBoost

A meta-estimator that starts by fitting a regressor, typically a weak learner, on the original dataset and then fits additional copies of the regressor on the same dataset such that the weights of instances are adjusted according to the error of the current prediction. And so, subsequent regressors focus more on difficult observations.
AdaBoost has an associated measure of feature importance, Gini importance, which is computed as the (normalized) total reduction of the criterion brought by that feature; the higher the Gini importance the more significant the feature.

KNN

An instance-based learning algorithm, which simply stores instances of the training data, and prediction is computed from a simple majority vote of the nearest neighbors of each point.

Pipeline

Given a prediction model M, the pipeline consists of 2 steps:

1. hyperparamter tuning, training, next-month return predictions and output scaling
2. IC calculation and t-tests

Step 1

For each month in the desired date range, we tune model M's hyperparameters on the previous 12-month returns, with Hyperopt, a library for Distributed Asynchronous Hyperparameter Optimization, using 3-Fold cross validation as the scoring function. Furthermore, we have implemented a custom scoring metric, where model predictions are scaled as per their percentile within the set of predictions, to achieve better results; ideally the hyperparameter-tuning metric should mirror that of model testing. Indeed, we achieved better returns using our custom scoring function as opposed to typical regression scoring functions.

Note that model predictions are stored in the output/predictions.csv file.

			RETURN
DATE	MODEL	SEDOL	
2004-11-01	AdaBoost	200001	0.216371
		200169	0.602711
		200230	0.459854
		200247	0.459854
		200418	0.459854

Figure 4. Model Predictions Snapshot

Step 2

Model M is trained on a 12-month rolling window, and for each training cycle, we scale predicted returns as per their percentile within the set of predictions, then regress them onto the actual next-month returns. the resulting regression coefficient is the IC, and the t-statistic of the coefficient is the t-test.

Note that IC and t-statistics are stored in the output/IC_T.csv file.

		IC	T
DATE	MODEL		
2004-11-01	AdaBoost	0.320545	1.703083e-29
	CTEF	0.346796	2.609842e-34
	DecisionTree	0.306195	1.268584e-74
	KNN	0.302439	5.242643e-26
	LinearRegression	0.315524	2.652425e-28

Figure 5. IC and t-statistics

Factor Significance

Only two of the five selected models, linear regression and adaboost, have an associated measure of factor significance. For each date in the desired date range, we save both measures of feature importance in the output/feature_importance.csv file. You will find below a summary of factor importance measures across the desired date range.

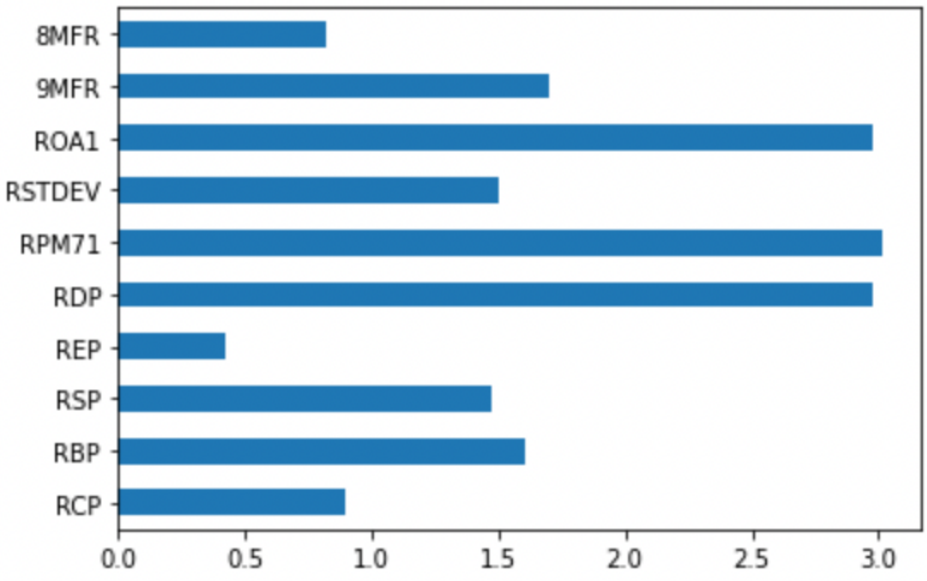

Figure 6. Linear Regression Feature importance

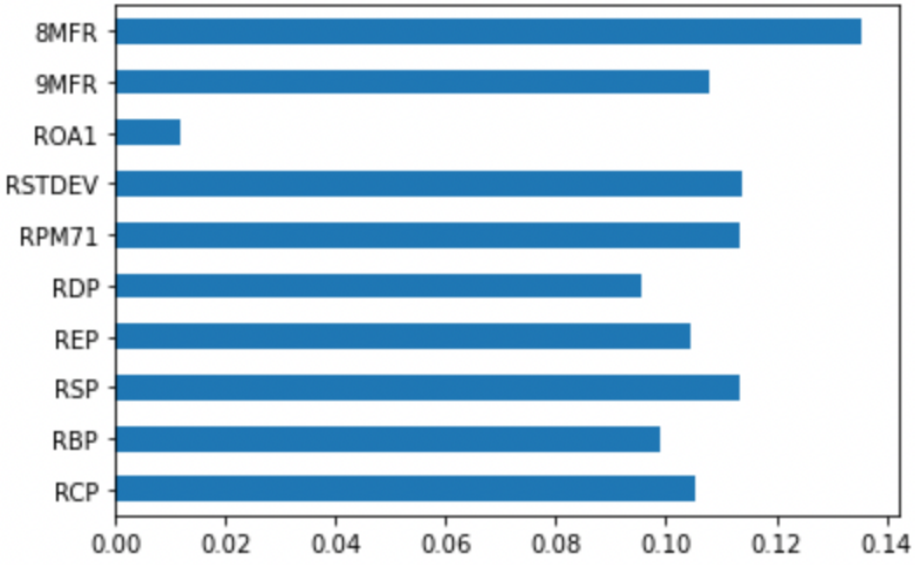

Figure 7. AdaBoost Feature importance

Model Evaluation & Comparison

The Information Coefficient ranges between 0 and 1, and measures the linear relationship between predicted and actual returns, where an IC of 1 indicates a perfect linear relationship. Moreover, the t-statistic measures the significance of the linear relationship between predicted and actual returns. Both measures depict a comprehensive picture of prediction accuracy and significance for each model.

We have found that all models have similar IC's of around 0.3, and t-statistics that tend to 0. Consequently, the models are adequate at predicting next-month returns.

MODEL	count	mean	std	min	25%	50%	75%	max
AdaBoost	169.0	0.305162	0.090702	0.071982	0.247556	0.312537	0.368495	0.544001
CTEF	169.0	0.302205	0.062018	0.129693	0.273387	0.303269	0.337733	0.533330
DecisionTree	169.0	0.294069	0.035022	0.132046	0.282064	0.295994	0.307717	0.458136
KNN	169.0	0.300463	0.062983	0.113952	0.263636	0.304665	0.339811	0.524093
LinearRegression	169.0	0.302205	0.087569	0.048915	0.246147	0.300956	0.352878	0.563768

Figure 8. Model Prediction IC

MODEL	count	mean	std	min	25%	50%	75%	max
AdaBoost	169.0	7.038677e-05	7.984030e-04	1.225132e-106	1.312281e-42	4.223394e-30	4.233736e-19	1.032420e-02
CTEF	169.0	8.067801e-08	9.231084e-07	2.107785e-94	1.805299e-33	4.364803e-27	3.775003e-22	1.196711e-05
DecisionTree	169.0	7.017029e-10	7.035682e-09	1.339652e-88	1.501533e-78	1.636620e-75	1.003067e-69	8.544652e-08
KNN	169.0	3.193878e-07	4.072603e-06	2.930523e-85	5.886502e-34	7.359697e-27	1.224315e-20	5.294248e-05
LinearRegression	169.0	5.953000e-04	6.969039e-03	1.012829e-105	2.395754e-36	3.347023e-26	4.521596e-18	9.009188e-02

Figure 9. Model Prediction t-statistics

PORTFOLIO CONSTRUCTION

For each set of model predictions, for each time period, we filter for assets with predicted returns in the 70th percentile of all predicted returns. The initial equal weighted portfolio is constructed from the filtered stocks in October 2004, and subsequent portfolios are formed by replacing the 4 stocks in the previous portfolio with the lowest predicted returns in the next time period with the 4 stocks with the highest predicted returns in the next time period.

More specifically, let **B** be the set of stocks in the portfolio at time t-1, **B*** be the set of stocks in the portfolio at time t, and **A** be the set of filtered stocks at time t + 1.

$\mathbf{B}* = \mathbf{B} - \text{argmin}_4(\mathbf{B} \cap \mathbf{A^C}) + \text{argmax}_4(\mathbf{B^C} \cap \mathbf{A})$

As the constructed portfolios are equal-weighted, the weights of a stock s in **B*** at time t, $w_{st} = \frac{1}{|\mathbf{B}*_\mathbf{t}|}$

PORTFOLIO EVALUATION

The portfolio based on KNN predictions has consistently fared the best amongst constructed portfolios, yet has far lesser returns over the time period than the Russell 1000 Index. The CTEF, Decision Tree, and AdaBoost portfolios have similar returns, while the Linear Regression portfolio exhibits little to no returns.

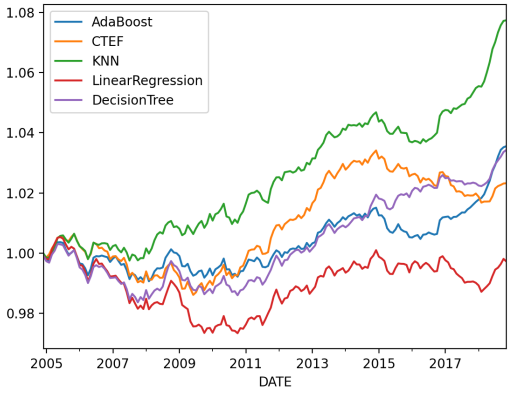

Figure 10. Model Cumulative Returns

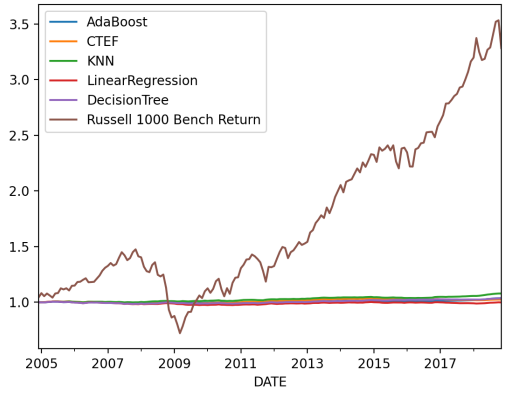

Figure 11. Cumulative Returns

Moreover, the returns of the constructed portfolios, while highly inter-correlated, exhibit close to no correlations with the benchmark.

	AdaBoost	CTEF	KNN	LinearRegression	DecisionTree	Russell 1000 Bench Return
AdaBoost	1.000000	0.872222	0.930784	0.860812	0.897851	-0.000576
CTEF	0.872222	1.000000	0.860796	0.859232	0.863831	0.026719
KNN	0.930784	0.860796	1.000000	0.756430	0.804608	0.054000
LinearRegression	0.860812	0.859232	0.756430	1.000000	0.948777	-0.048044
DecisionTree	0.897851	0.863831	0.804608	0.948777	1.000000	-0.044014
Russell 1000 Bench Return	-0.000576	0.026719	0.054000	-0.048044	-0.044014	1.000000

Figure 12. Return Correlations

Indeed, the constructed portfolios greatly outperform the Russell 1000 index during the Great Recession, as they continue to demonstrate steady positive returns, while the index loses 50% of it's value.

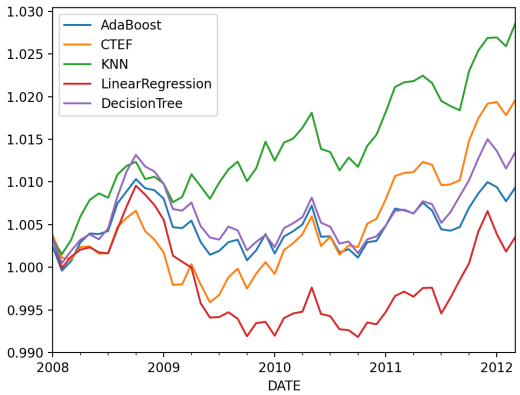

Figure 13. Model Cumulative Returns - Crisis

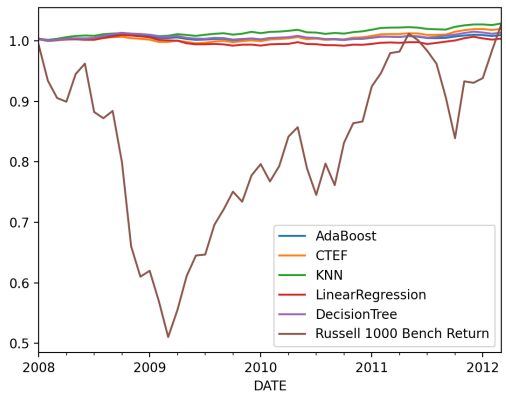

Figure 14. Cumulative Returns - Crisis

Evaluation Metrics

We further evaluate portfolio performance with metrics beyond absolute cumulative returns. As per Fig. 16, the KNN-portfolio has a significantly higher information ratio than the benchmark, thus better risk-adjusted alpha. Furthermore, all constructed models have far better max draw downs than the benchmark, namely one to two orders of magnitude lower (Fig. 17).

	RETURN
AdaBoost	1.035549
CTEF	1.023244
KNN	1.077383
LinearRegression	0.997273
DecisionTree	1.034288
Russell 1000 Bench Return	3.283606

Figure 15. Returns

	IR
AdaBoost	0.492154
CTEF	0.297336
KNN	0.970785
LinearRegression	-0.029900
DecisionTree	0.467892
Russell 1000 Bench Return	0.685119

Figure 16. Information Ratios

	MDD
AdaBoost	0.012955
CTEF	0.020111
KNN	0.009845
LinearRegression	0.031763
DecisionTree	0.019325
Russell 1000 Bench Return	0.511259

Figure 17. Max DD

More Comparison

We thought it interesting to compare portfolio size across models. We have found that the Decision Tree portfolio contains significantly more assets, while the other constructed portfolios have sizes of around 300 assets.

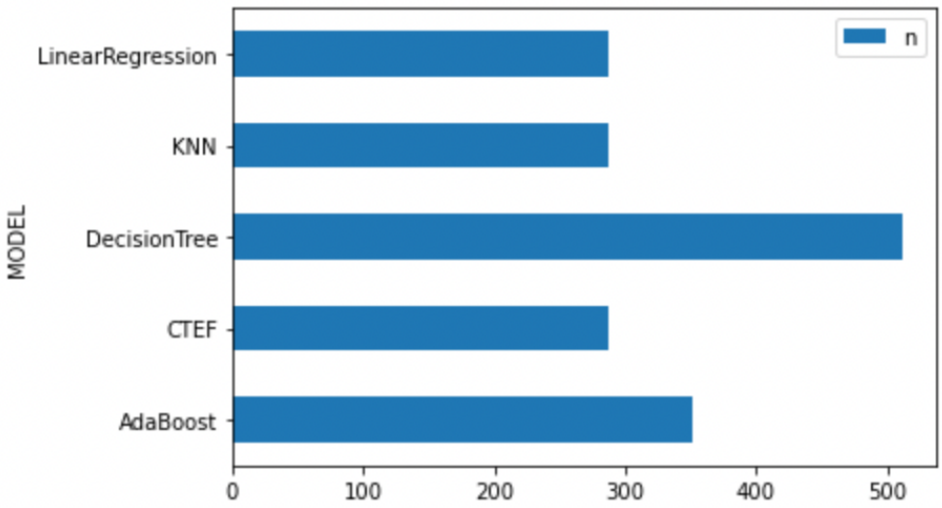

Figure 18. Portfolio Sizes

CONCLUSION

The model-filtered equal-weighted portfolios have stable returns, uncorrelated to benchmark returns, and are opportunities to realize risk-adjusted alpha across market cycles.

APPENDIX

Contributions

Pranav Mehta

- data wrangling
- portfolio construction
- graphs and plots

Karim Layoun

- prediction pipeline
- portfolio construction
- graphs and plots (to a lesser extent than Pranav)