Deutsche Bank
Markets Research

Asia
Japan

Quantitative Strategy
Quantiles

Date
30 September 2016

Machine Learning in Finance

Vincent Zoonekynd
vincent.zoonekynd@db.com

Khoi LeBinh
khoi.lebinh@db.com

Ada Lau
ada-cy.lau@db.com

Hemant Sambatur
hemant.sambatur@db.com

North America: +1 212 250 8983
Europe: +44 20 754 71684
Asia: +852 2203 6990

Machine learning is everywhere

"Machine learning" repeatedly appears in the news, from the game of go to autonomous cars: what can those algorithms do for us in finance?

Supervised learning and its pitfalls in finance

In this first report in the series, we focus on supervised learning and note that while machine learning is very relevant to us, there are dangerous pitfalls, sometimes specific to the type of data we deal with. In particular, we examine penalized regression (lasso and elastic net), decision trees, and boosting – we also mention, in passing, support vector machines and random forests.

Application to the Japanese equity market

To make things more concrete, we try to use those algorithms to combine the investment factors in our database in order to build a stock ranking system for the Japanese market; this shows the limitations and pitfalls of traditional machine learning practices in finance.

Takeaways

More specifically, we:

- Explain the ideas and the math behind some of the most popular machine learning algorithms;
- Show that the usual ways of preventing overfitting in machine learning fail in finance;
- Layout a recipe for the construction of a machine-learning-based investment strategy;
- Emphasize that investing is still an art in which we may want to remain on top of the recommendation suggested by the machine.

Deutsche Bank AG/Hong Kong

Table Of Contents

A letter to our readers

Statisticians want to turn humans into computers.
Machine learners want to turn computers into humans.
We meet somewhere in the middle.
N. Lawrence, 2012

Machine learning is all over the news, from self-driving cars to the game of go. In this report, we see what machine learning is, how it can be applied in finance, and what the pitfalls to avoid.

We have used machine learning in many of our past reports.

- For instance, when examining quality indicators [51], we used **penalized regression** to forecast future returns, **penalized logistic regression** to forecast future drops in prices and **random forests** to combine those indicators into an investment signal.
- Our report on insider transactions [7] relied on **boosting** to deal with large numbers of predictive variables.
- In our momentum trilogy [52, 53, 54], we used, among other tools, **generalized additive models** (GAM) and **state space models**.[1]
- When studying risk aversion [25], we combined risk indicators with a **non-negative matrix factorization** (NMF) and inferred relations between risk factors with **Bayesian networks**.
- The way we group investment factors in our Quantfucius reports, with "truncated graphs", can be seen as an application of **social network analysis**, or a crude form of **topological data analysis** [10, 9, 15].

This report examines machine learning more systematically.

We provide an introduction presenting some historical background, the main types of machine learning (supervised and unsupervised learning), and the main software tools. Then in the first part of this report, we examine in more detail some popular machine learning algorithms, such as penalized regression, support vector machines, decision trees, random forests and boosting.

In the second part of the report, we try out some of these algorithms to devise an investment strategy, using the Japanese market as an example. This shows more concretely what limitations those algorithms have and what dangers they present. In particular, we show that the traditional way of avoiding overfitting, cross-validation fails in finance: as a remedy, we suggest to check that the model remains interpretable.

[1] State space models may not sound like machine learning, but we use a rather broad definition: we call "machine learning" any algorithmic approach to statistical problems. Since the Kalman filter is an algorithmic procedure to efficiently compute conditional Gaussian distributions, it falls under that umbrella. You may have been doing machine learning without knowing it.

In the conclusion, we put forward a non-machine-learning procedure that performs even better, list the steps to follow should you want to use machine learning as part of your investment process, stress again the main pitfall (classical ways of preventing overfitting fail), and list other potential applications of machine learning.

Yours sincerely,

Vincent, Khoi, Ada, Hemant & the Global Quantitative Strategy Team.

Deutsche Bank Asia Quantitative Strategy Team.

Screen

Figure 1 shows some large caps in the long and short portfolios built from the final model examined in this report (a constrained regression), for Japan, as of this writing. Since the model is linear, we can decompose the score of each stock as a sum of contributions of the individual variables, and group those contributions according to the type of information each factor tries to capture. The largest weights (short-term reversal, 12-minus-1-month momentum, 5-year risk, sales diffusion, dividend yield, etc.) are shown in figure 81.

Figure 1: Large caps with the highest and lowest scores

Id	Name	MCap	Score	Quality	Sentiment	Value	Growth	Momentum	Risk	Size
7751-JP	Canon	32	8.69	2.50	0.55	1.04	-0.56	-0.15	4.18	1.14
7267-JP	Honda Motor	54	8.13	0.83	-0.66	2.41	0.26	0.36	3.81	1.14
8035-JP	Tokyo Electron	15	7.16	1.59	2.13	-1.50	0.37	1.76	1.65	1.16
4901-JP	FUJIFILM Holdings	17	6.63	2.07	-0.38	0.82	0.22	-0.21	3.46	0.64
5802-JP	Sumitomo Electric Industries	12	6.30	0.80	-0.36	1.73	0.21	0.93	1.81	1.18
9735-JP	Secom	16	5.91	0.60	0.48	-1.79	0.15	1.20	4.20	1.07
3407-JP	Asahi Kasei	11	5.75	1.36	0.43	1.26	-0.06	1.14	0.68	0.93
7201-JP	Nissan Motor	26	5.67	1.47	-0.39	1.73	0.18	0.04	1.45	1.18
8031-JP	Mitsui	24	5.48	-1.99	1.83	2.26	-0.63	0.82	2.11	1.07
5020-JP	JX Holdings	10	5.17	-2.17	-0.54	4.15	-0.31	-0.28	3.15	1.18
9984-JP	SoftBank Group	61	-3.33	-0.29	1.50	-2.58	-0.27	1.47	-2.49	-0.67
6869-JP	Sysmex	13	-3.44	1.54	-0.40	-4.10	0.41	0.14	-1.92	0.90
3382-JP	Seven & I Holdings	38	-3.81	-0.20	-1.44	-0.89	0.11	-1.14	-1.41	1.15
8113-JP	Unicharm	12	-4.11	0.40	-2.40	-3.45	-0.04	1.22	-0.80	0.96
7974-JP	Nintendo	32	-4.51	-0.43	1.76	-3.85	0.13	0.21	-1.30	-1.02
4755-JP	Rakuten	12	-5.98	-0.99	-0.14	-3.95	-0.07	-0.75	0.67	-0.75
4528-JP	ONO Pharmaceutical	15	-9.01	1.17	-1.91	-3.58	0.22	-0.16	-4.08	-0.68

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

What is machine learning?

Is machine learning just a rebranding of statistics?

For several years, "machine learning" has been a recurrent news item: self-driving cars, machine translation, recommendation engines, machine learning competitions, deep dream (Figure 2 and Figure 3), alpha go, etc. What is behind this term?

Depending on who you ask, you will get different answers.

For some, it is a response to the failure of artificial intelligence (AI). In the 1960s, researchers were optimistic, and thought that tasks like machine translation were only a few months away. They tried to teach computers everything that could be known about the world, together with rules to deduce new facts. Unfortunately, as those rule systems became larger and larger, it became apparent that with the slightest inconsistency or error in those rules, the system would grind to a halt. The systems were not robust enough. After a long "AI winter" (Figure 6), researchers eventually managed to introduce uncertainty into the logic machinery: machine learning became a part of statistics, while retaining its algorithmic roots.

For some, "machine learning" is just a re-branding, or a re-discovery of statistics: indeed, most of the methods of machine learning were already known to statisticians and were simply re-developed, re-named or re-interpreted by computer scientists. Figure 5 gives a few examples.[2]

Our definition is somewhere in between: for us, **machine learning** is an empirical, algorithmic approach to the problems already tackled by Statistics.

Figure 2: Deep Dream

Source: J. Mullen, https://en.wikipedia.org/wiki/File:Deep_Dreamscope_(19822170718).jpg

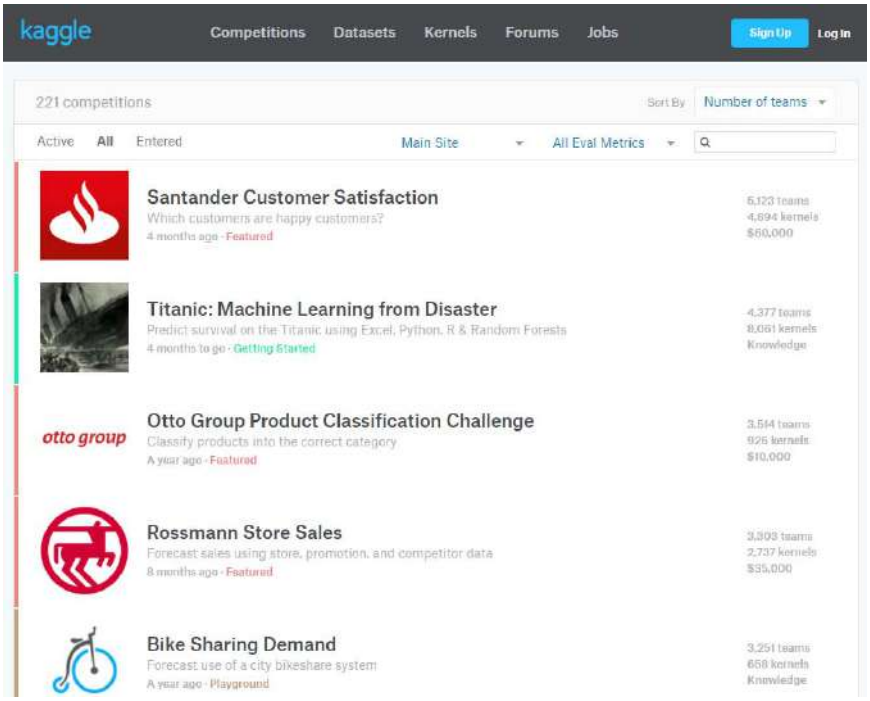
[2] Also see http://statweb.stanford.edu/~tibs/stat315a/glossary.pdf

Figure 3: Self-driving car

Source: S. Jurvetson, https://en.wikipedia.org/wiki/File:Google%27s_Lexus_RX_450h_Self-Driving_Car.jpg

Figure 4: Kaggle home page

Kaggle is a website hosting "data science competitions": participants use machine learning tools to solve a problem, e.g., automatically distinguishing between cats and dogs, identifying diabetic retinopathy in eye images, or making better customer recommendations; for some competitions, the best entry receives a cash prize.



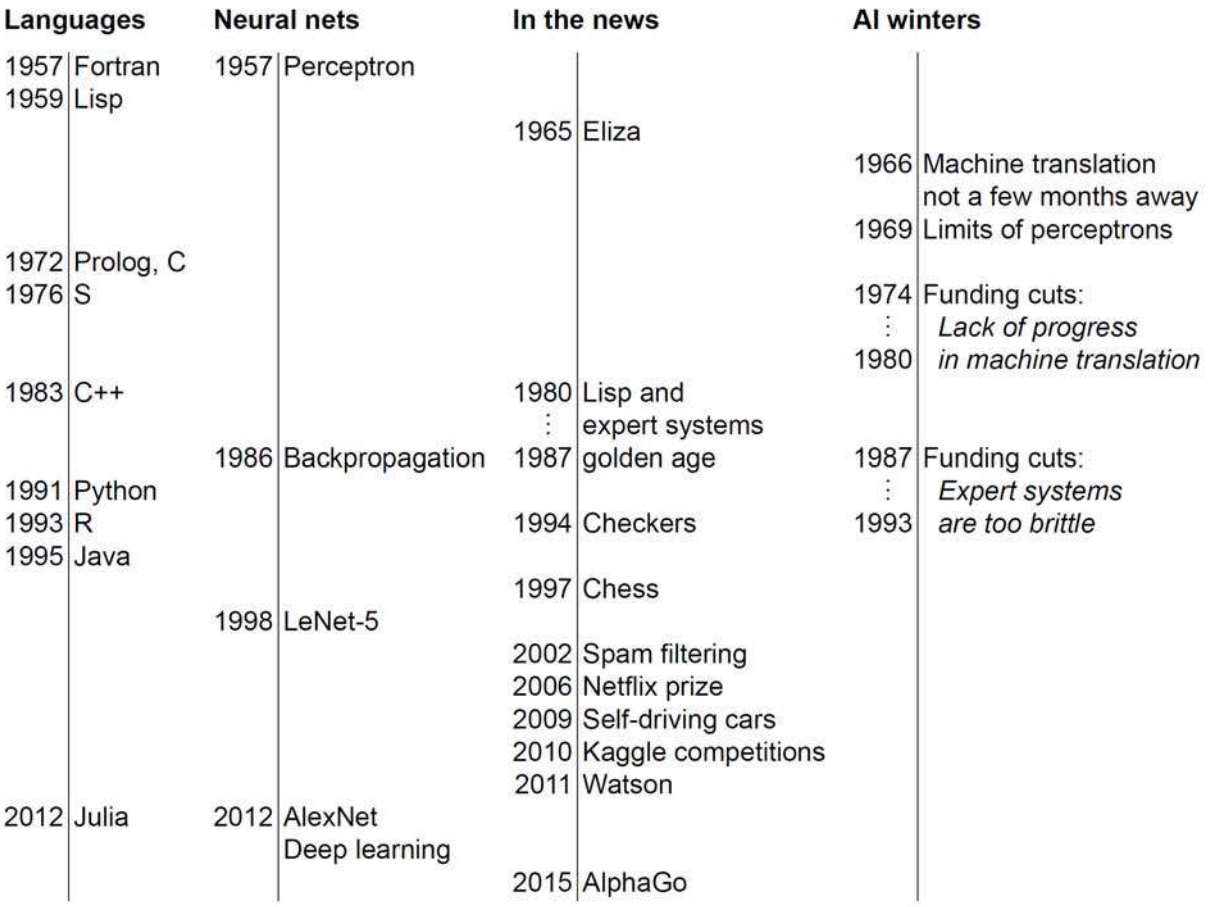
Source: https://www.kaggle.com/competitions

Figure 5: (Necessarily incomplete) lexicon from machine learning to statistics and finance

Machine learning	Statistics, finance
Supervised learning	Regression, classification
Unsupervised learning	Clustering, dimension reduction, density estimation
Feature	Predictive variable, dependent variable
Manual feature engineering	Financial ratios, valuation models
Training set	In-sample
Test set	Out-of-sample
Learn, train	Fit
Softmax classifier, cross-entropy classifier	(Multinomial) logistic regression
Large-margin classifier	Support vector machine
Weights	Parameters
Regularization	Penalty, Bayesian prior

Source: Deutsche Bank Quantitative Strategy

Figure 6: Artificial intelligence timeline



Source: Deutsche Bank Quantitative Strategy

Machine learning algorithms and tools

Machine learning problems are usually divided into two broad categories: supervised and unsupervised.

In **supervised learning**, we are trying to predict something ("labels"), for instance future returns, or whether a company will default in the near future; the training data contains the ground truth (past returns, past defaults). Regression and classification are supervised learning problems.

In **unsupervised learning**, we are trying to find structure in the data, but there is nothing to predict: for instance, we can try to find clusters in the data (but we do not know beforehand what those clusters mean) or a low-dimensional representation of the data (but we do not know beforehand what those dimensions mean). Clustering, dimension reduction and density estimation are unsupervised learning problems.

In finance, unsupervised learning can be used to group stocks (you may end up with industries, or countries, or value versus growth stocks, or a combination of those attributes, or something entirely different), or investment factors, or as a pre-processing step to reduce the dimension of the data.

However, not all problems fit into those two main categories.

In **semi-supervised learning**, only part of the data is labelled: in the training set, we only know the variable to predict for some of the observations. The problem is often tackled by a mixture of supervised and unsupervised learning, using unsupervised methods to understand the structure of the (mostly-unlabelled) data, and supervised methods to leverage that structure.

For instance, one may want to teach the computer to distinguish between cat and dog pictures, from a small set of labelled pictures, but with a large number of unlabelled pet pictures: we can give the pet pictures to an unsupervised learning algorithm, to learn the features present in those images (the algorithm does not know what they are, but they could turn out to be pointy ears, flappy ears, fur, eyes, tail, etc.), and then use those features, with a supervised learning algorithm, on the (labelled) cat and dog pictures.

Active learning starts with a completely unlabeled dataset, and lets us choose which observations we want labeled, depending on the cost of those labels (e.g., if they require expensive or time-consuming physical or medical experiments or, in finance, human expertise to read and understand a press release or a contract) and how informative we hope them to be.

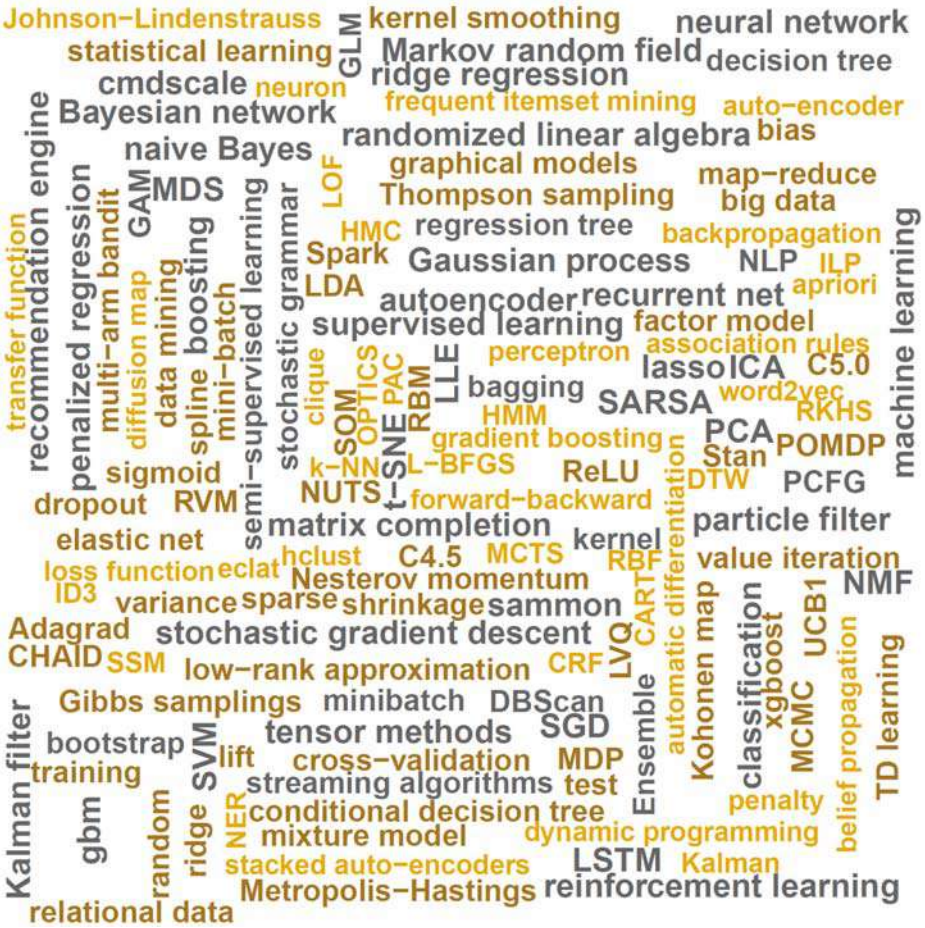
Online learning refers to algorithms that adapt over time, as data is progressively revealed. In addition, streaming algorithms must process the information without storing all of it.

Transfer learning learns a model on one dataset and applies it to a related but different one. This is common in medicine (the model is estimated on a few patients, for which we took the time to carry out all the imaginable medical tests, and applied on different patients, when we only have the time and money to do some of those tests), and image or text processing (the model learns what a photograph looks like from a large image database, what English

text looks like from a huge and diverse corpus, but then applies this knowledge to a much narrower domain, e.g., analyzing satellite images or understanding business contracts). In finance, we may sometimes want to fit a model for a country with enough historical data, and apply it, perhaps with a few modifications, to a market with not enough training data.

Reinforcement learning [41] tries to find actions (moving a robotic arm, moving a chess piece, buying or selling a stock) that optimize some reward. However, the reward is only known much later, and we do not know which actions (chess moves) played a role and which were irrelevant. In finance, reinforcement learning can be used to devise trading strategies ("I want to sell *n* shares by the end of the day: should I place a big order now, or smaller orders through the day? Should I start with large orders and decrease their sizes or the opposite?") or multi-period investment strategies ("How should the balance between stocks and bonds change, in my retirement account, as I age?").

Figure 7: The machine learning vocabulary is often overwhelming



Source: Deutsche Bank Quantitative Strategy

Most quants do their computations through languages such as R or Python.

For machine learning, **Python** provides a unified interface to the most common algorithms, through scikit-learn. It is therefore very easy to try new algorithms, but only the most standard ones are available.

R presents a completely opposite picture: there is no unified interface (though packages such as caret or mlr try to provide one) but a gazillion independent packages, providing (almost) all the variants of all the algorithms you can think of – Figure 8 lists the packages we had in mind (or used) when writing this report, but there are many more.

Figure 8: A few R packages and functions for machine learning. This list is very incomplete: see https://cran.r-project.org/web/views/MachineLearning.html for more.

Supervised learning:

- Penalized regression: glmnet
- Boosting: xgboost, gbm, mboost
- Generalized additive models (GAM), splines, smoothing: gam, mgcv, fda, splines::ns, rms::rcs, Hmisc::rcspline, stats::loess
- Neural nets: net::multinom, mxnet
- k-nearest neighbours: class::knn
- Linear or quadratic discriminant analysis: MASS::lda, MASS::qda
- Support vector machines: e1071::svm
- Decision trees: rpart, partykit::ctree
- Random forests: randomForest
- Ensemble learning: ForecastCombinations

Unsupervised learning:

- Clustering: stats::kmeans, dbscan
- Principal component analysis (PCA) and variants: stats::prcomp, fastICA, NMF, stats::cmdscale, MASS::isoMDS, MASS::sammon, Rtsne
- Manifold learning: vegan::isomap
- Graphs: igraph, ape::mst
- Graphical models: bnlearn
- Topological data analysis: TDA

Source: Deutsche Bank Quantitative Strategy

We sometimes use separate programs. Here are a few examples.

- Vowpal Wabbit[3] is a command line tool for large linear models – useful if the data, or even the model, does not fit in memory.
- Stan[4] is a probabilistic programming language for Monte Carlo estimation of Bayesian models.

The "big data" ecosystems provide a rather limited number of machine learning algorithms – we do not recommend them unless you already have the required infrastructure and your data is too large for other tools.

- Hadoop[5] and its Mahout[6] machine learning library are limited by their reliance on the Map/Reduce paradigm: most machine learning algorithms are iterative and forcing them into that framework leads to poor performance;[7]
- Spark[8] and its MLlib[9] library address those problems;
- More recent tools, such as H2O[10] (machine learning on big data, from R or other languages) or Flink[11] (a Spark clone) may be worth investigating.

Prompted by the recent media coverage of "deep learning", some of you may be tempted by neural networks [12, 21, 33]. Here are the main libraries to estimate those models.

- Theano[12] is a low-level Python library for neural nets; you may prefer a higher-level interface such as Keras[13] or Lasagne[14];
- TensorFlow[15] is a more recent low-level Python library for neural nets, trying to replace Theano; it can also be used through Keras;
- MxNet[16] is a C++ library with interfaces in many languages (R, Python, etc.) – in R, this is probably your current best (or only) alternative;
- Torch[17] is a Lua[18] library for neural networks with a strong emphasis on image processing;
- Caffe[19] also focuses on image analysis; since it is not a library (you cannot program – you can only write configuration files), you may find it limiting;
- DeepLearning4j[20] is a Java and Scala library for neural networks.

[3] https://github.com/JohnLangford/vowpal_wabbit
[4] http://mc-stan.org/
[5] http://hadoop.apache.org/
[6] http://mahout.apache.org/
[7] See the benchmarks on http://spark.apache.org/.
[8] http://spark.apache.org/
[9] http://spark.apache.org/mllib/
[10] http://www.h2o.ai/
[11] https://flink.apache.org/
[12] http://deeplearning.net/software/theano/
[13] https://keras.io/
[14] https://github.com/Lasagne/Lasagne
[15] https://www.tensorflow.org/
[16] https://github.com/dmlc/mxnet
[17] http://torch.ch/
[18] Lua is a programming language used in embedded systems and the video games industry.
[19] http://caffe.berkeleyvision.org/
[20] http://deeplearning4j.org/

Why does machine learning work – or fail?

Here are the main reasons why machine learning succeeds, when it does.

- **Manual feature engineering.** Although "deep learning" shows promises in some domains, machine learning often still struggles to combine pieces of information in complex ways. Using some domain knowledge to define and precompute variables likely to have some predictive power (or, simply, interpretable quantities) still proves useful. In finance, this includes, for instance, financial ratios, technical indicators and valuation models (dividend discount model, etc.).
- **Ensembling.** Instead of learning a single model, it is often preferable to learn several and combine their forecasts; since they are unlikely to make the same errors, those errors will cancel out, to some extent, giving a more precise (lower variance) forecast.
- **Regularization.** While one may be tempted to consider ever complex models, this leads to overfitting. To control overfitting, one can add a penalty for complexity, so that complex models will only be selected if the added complexity is worth it, i.e., if the improvement they bring compensates the complexity penalty they incur.
- **Optimizing the correct metric.** While statistics have a fairly limited number of ways to measure how good a model is (basically, the log-likelihood or some approximation of it), machine learning is more empirical and accepts virtually anything as a loss function: in particular, one can take a measure meaningful from a business point of view, a measure reflecting the costs and benefits of the solution. For instance, in finance, we often follow a multi-step procedure: forecast future returns, use the forecasts to build a portfolio and look at its performance – what we optimize, the quality of the return forecasts, and what we care about, the performance of the portfolio, are different things. Machine learning allows a single-step approach, in which we directly optimize the portfolio performance.

But machine learning does not always succeed. Here are the main causes for failure.

- **Complicated models.** Even though your model may give consistently good forecasts, it may be too complex to be of practical use and put into production. This is particularly the case with ensemble models: if the model is the average of thousands of already large models, computing the forecasts may be too time- or resource-intensive.
- **Overfitting.** Even though there are methods to estimate, or at least control, overfitting (regularization, cross-validation), the pitfalls are numerous and it remains a very real problem. We note that the situation is actually much worse in finance.

Theory

In this section, we review a few algorithms one could use to forecast forward returns. If you are unfamiliar with those methods, you can find more details in [22] or [19].

In particular, we focus on the following.

- **Penalized regression** (lasso or elastic net) is an alternative to ordinary least squares when there are too many predictors, or when we suspect that only some of them are relevant; it tries to select a sparse model, which will be less noisy and easier to interpret.
- **Support vector machines** (SVM, large margin classifier, hinge loss classifier) have a good reputation for classification problems (i.e., to predict qualitative variables) but can also be used for regression problems (i.e., to predict quantitative variables). Kernel SVMs can capture non-linear relations and interactions, and are insensitive to the number of predictors but, unfortunately, not to the number of observations – using them for large datasets is problematic.
- **Decision trees** are easy to interpret and can capture non-linear relations, and even interactions, but they are often too tied to the training dataset to be relied on.
- **Random forests** are sets of decision trees that try to address that problem – unfortunately, they are no longer interpretable and the models can become extremely large. **Boosted decision trees** use a similar idea, but are less unwieldy.
- **Generalized additive models** (GAM) are a generalization of linear models that allows for non-linear relations – indeed, we often notice that extreme values behave differently, and V- or Λ-shaped relations are not uncommon. Boosting adds a lasso flavour to GAMs and allows them to robustly deal with a large number of variables.

Figure 9 summarizes the benefits and limitations of those algorithms.

While we have a preference for the lasso (fast, robust, interpretable) and the boosted GAM models (robust, interpretable, non-linear), there is no best algorithm: the "no free lunch theorem" (yes, it is a theorem: see [49, 48] for a precise statement and proof) states that no algorithm is good for all problems. In other words, you will need different algorithms for different situations.

Figure 9: Advantages and limitations of the models presented in this report

	Interpretation	Number of variables	Nonlinearities	Interactions	Overfit	Speed	Size
Linear model	O	X	X	X	X	O	O
GAM	O	X	O	O	X	O	O
Lasso	O	O	X	X	O	O	O
Linear SVM	X	O	X	X	O	O	O
Kernel SVM	X	O	O	O	O	X	X
Decision tree	O	X	O	O	X	O	O
Random forest	X	O	O	O	O	X	X
Boosted trees	X	O	O	O	O	O	O
Boosted GAM	O	O	O	O	O	X	O

Source: Deutsche Bank Quantitative Strategy

Penalized regression and sparse models

The most straightforward method to predict future returns is the linear regression.

$$\text{forward returns} = \alpha + \sum_i \beta_i \times \text{predictor}_i + \text{noise}$$

The coefficients β can be estimated by minimizing the sum of squared residuals. More generally, many statistical models are estimated by minimizing some loss function: a sum of squares for ordinary least squares, a sum of absolute values for robust (LAD, least absolute deviation) regression, minus the log-likelihood for logistic regression, etc.

Unfortunately, this does not scale: if there are many predictors, and if several contain similar information (e.g., different measures of "value", or the same financial ratio computed from different sources), we often end up with very large positive or negative coefficients (Figure 10 and Figure 11). In-sample, they cancel each other out and do no harm, but out of sample, they can turn the investment signal into pure noise.

Figure 10 gives a worked out example: a linear regression with two variables containing almost the same information. In this example, we attempt to explain future returns from two related financial ratios, two forms of Debt/Equity, where the equity comes from accountants or from the market. The linear regression gives large coefficients to those two variables, with opposite signs that almost cancel each other out,

return = −0.52 **− 2.49** × Debt/Equity_1 **+ 2.67** × Debt/Equity_2 + residual.

The penalized regression, in contrast, starts with an empty model and progressively increases the weights: we can stop before the coefficients get too large or end up with a counter-intuitive sign, e.g.,

return = intercept + **0** × Debt/Equity_1 + **0.18** × Debt/Equity_2 + residual.

Figure 10: Linear regression when two variables contain almost the same information.

```
> library(glmnet); library(magrittr); library(Matrix)
```

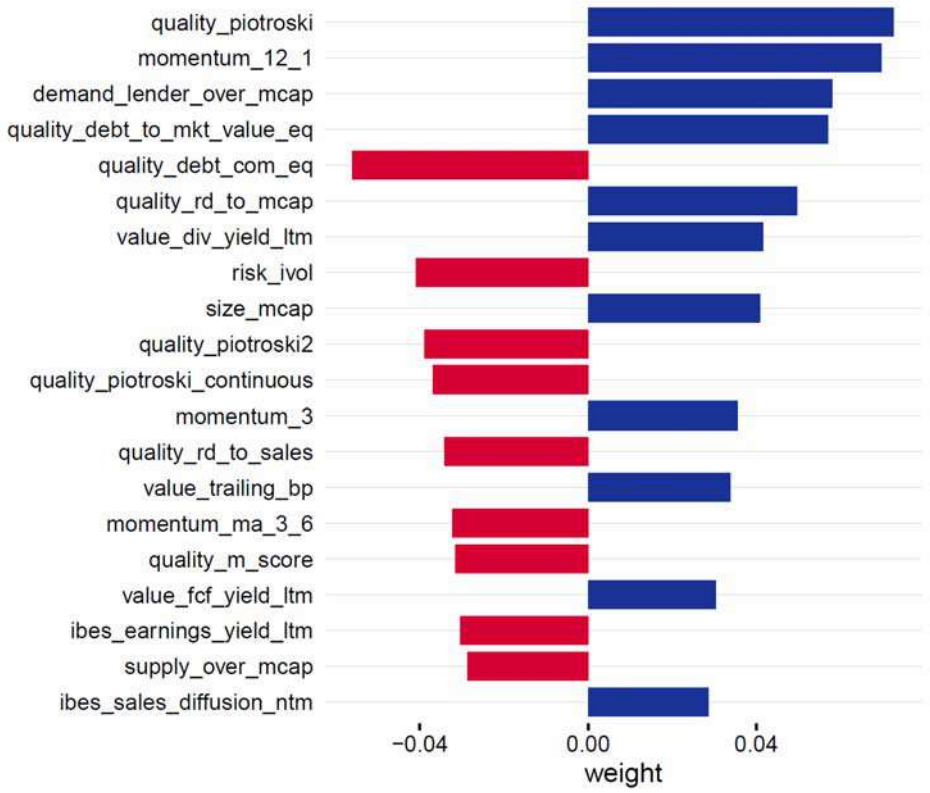
```
> str(d)

'data.frame': 188125 obs. of  3 variables:
 $ returns                    : num  39.122 -11.559 -2.703 0.632 1.071 ...
 $ quality_debt_com_eq        : num  -0.663 -0.7723 -0.3735 0.0739 -0.7075 ...
 $ quality_debt_to_mkt_value_eq: num  -0.767 -0.65 -0.179 0.113 -0.476 ...
```

```
> lm( returns ~ ., d ) %>% summary


Call:
lm(formula = returns ~ ., data = d)

Residuals:
    Min      1Q  Median      3Q     Max
-71.356  -5.481   0.235   5.721  69.835

Coefficients:
                           Estimate Std. Error t value Pr(>|t|)
(Intercept)                -0.52400    0.02524  -20.76   <2e-16 ***
quality_debt_com_eq        -2.49380    0.11658  -21.39   <2e-16 ***
quality_debt_to_mkt_value_eq  2.67273    0.11579   23.08   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 10.95 on 188122 degrees of freedom
Multiple R-squared:  0.002824,Adjusted R-squared:  0.002814
F-statistic: 266.4 on 2 and 188122 DF,  p-value: < 2.2e-16
```

```
> glmnet(x=X, y=Y) %>% .$beta %>% t %>% head(10) %>% signif(2)

10 x 2 sparse Matrix of class "dgCMatrix"
   quality_debt_com_eq quality_debt_to_mkt_value_eq
s0                   .                           .
s1                   .                       0.034
s2                   .                       0.065
s3                   .                       0.093
s4                   .                       0.120
s5                   .                       0.140
s6                   .                       0.160
s7                   .                       0.180
s8              -0.034                       0.230
s9              -0.250                       0.440
```

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 11: Linear regression to forecast future returns: variables with the largest weights. The performance of the model is not worrying: the out-of-sample information ratio (IR) is 1.8. The amplitude of the weights is not worrying: there are 124 predictors and both predictors and variable to predict are uniformly distributed on [0, 1]. However, we notice pairs of positive and negative coefficients for similar variables that almost cancel each other out: variants of the Piotroski signal, Debt/Equity ratio, R&D, etc.: in spite of its apparent good out-of-sample performance, the model is clearly overfitting the data.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

To control this phenomenon, we can add a penalty (or regularizing term) for large coefficients: instead of looking for the parameters β that minimize the loss, we can minimize the sum of the loss and a penalty.

$$\begin{array}{ll} \text{Find} & \beta \\ \text{To minimize} & \text{Loss}(\text{data} \mid \beta) + \text{Penalty}(\beta) \end{array}$$

Common penalties include:

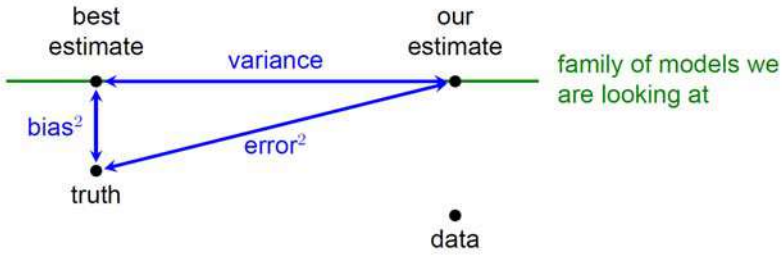
- The sum of the squared coefficients (called L^2 penalty, or ridge regression penalty, or Tikhonov regularizer);
- The sum of the absolute values of the coefficients L^1 (**lasso**);
- A linear combination of L^2 and L^1 penalties (**elastic net**).

Penalization can often be interpreted as a Bayesian prior (a Gaussian prior for the L^2 penalty, a Laplace prior for the L^1 penalty)[21] or as a shrinkage towards a simpler model (the constant model, where all the coefficients β are zero).

The benefits of the penalty are manifold:

- It allows models with a large number of parameters, even models with more parameters than data points;
- It helps avoid extreme values in the coefficients;
- It reduces variance – it also increases bias, but the resulting error, which combines variance and bias, is often lower;
- The L^1 penalty can generate **sparse models**, i.e., models in which most of the coefficients are zero, i.e., models that only use a small number of variables – such models are particularly easy to interpret.

Figure 12: Bias-variance trade off. The **bias** is error coming from the fact that our model cannot capture the complexity of the data. The **variance** error comes from the fact that the training data, being random, does not fully reflect the properties of the data.



Source: Deutsche Bank Quantitative Strategy

[21] A Bayesian MAP (maximum a posteriori) estimator maximizes the posterior model log-probability, which is the sum of the model log-likelihood and the prior log-probability. For a Gaussian prior, this prior log-probability is an L^2 norm, which can be interpreted as an L^2 penalty; for a Laplace prior, the log-probability is an L^1 norm. See [20], section 2.9 for more details.

To understand why the lasso yields sparse models, let us notice that the penalized optimization problem

$$\begin{array}{ll} \text{Find} & \theta \\ \text{To minimize} & \text{Loss}(\theta) + \lambda \|\theta\|_1 \end{array}$$

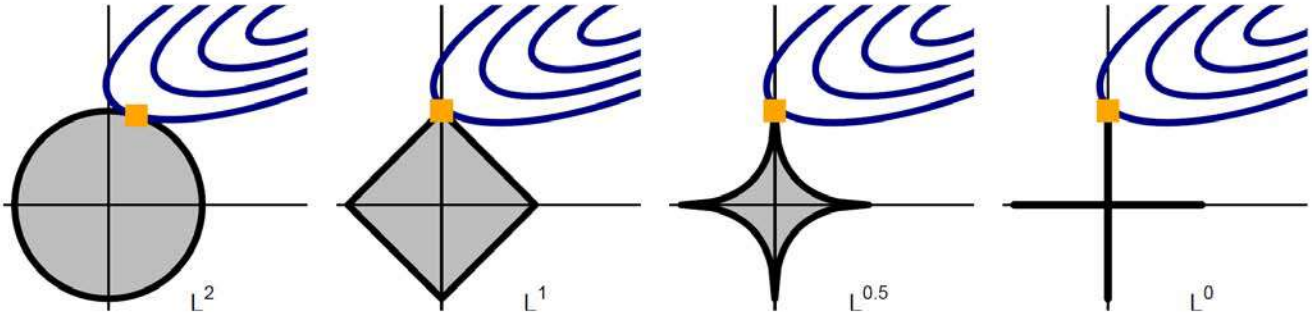
is the Lagrangian dual of the constrained optimization problem

$$\begin{array}{ll} \text{Find} & \theta \\ \text{To minimize} & \text{Loss}(\theta) \\ \text{Such that} & \|\theta\|_1 \leqslant c. \end{array}$$

Since the loss function is often smooth, its level lines (or hypersurfaces) can be thought of as balloons (blue, in Figure 13), inflating as we move away from the unconstrained minimum. Since the feasible area (grey) is an L^1 ball, it has sharp edges. The solution of the constrained optimization problem is the smallest level line that touches the feasible area.

When a balloon touches an object with a lot of sharp edges, it tends to touch it at a vertex, an edge or a low-dimensional facet – those points correspond to models with all but one, two, or a small number of non-zero coefficients.

Figure 13: Optimization with an L^2 (ridge), L^1 (lasso), L^p ($0 < p < 1$) or L^0 constraint. The grey area is the set of feasible solutions, the blue lines the level curves of the objective, and the orange square the optimal solution.



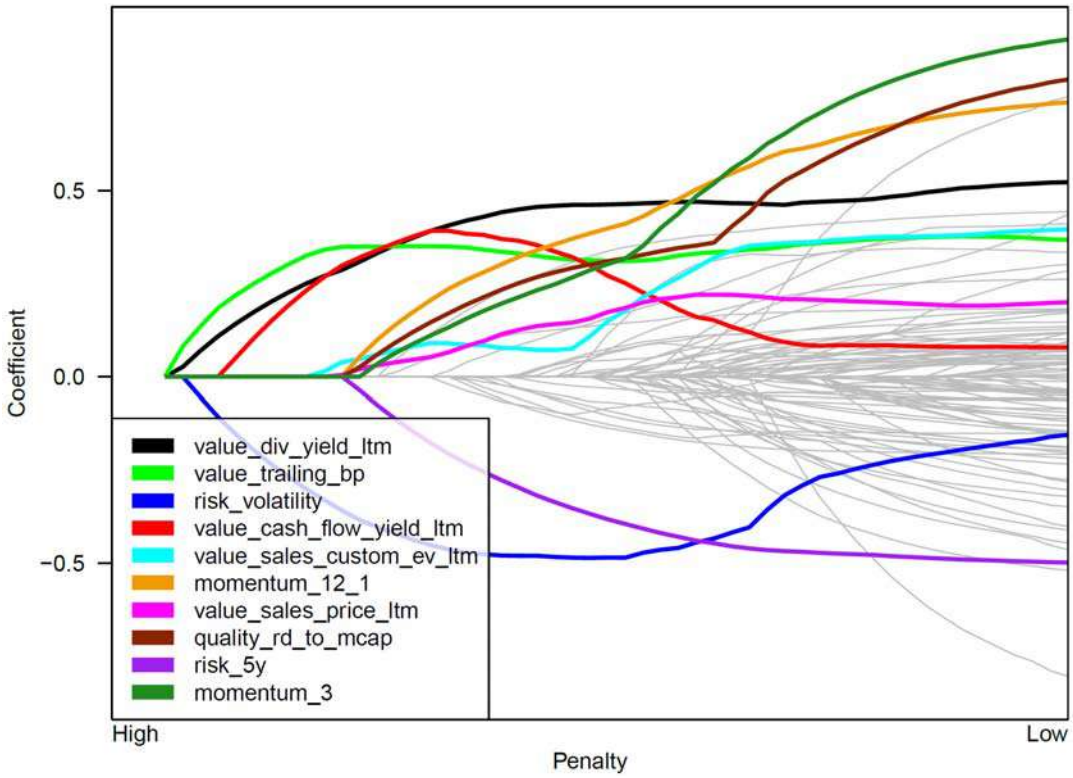
Source: Deutsche Bank Quantitative Strategy

There is one last detail to sort out: how do we choose the penalty coefficient λ? We can weasel out of this problem by not choosing it and letting it vary: we do not get a single solution, but a whole family of solutions, a **regularization path**, from the trivial, constant model (for $\lambda = \infty$) to the OLS model (for $\lambda = 0$). Eventually, however, we have to select one of those models: this can be done by cross-validation, or by looking at the out-of-sample performance, and/or by checking which models are interpretable.

Figure 14 shows a regularization path. The penalty decreases along the horizontal axis, from a model with no variables on the left, to the (unconstrained) ordinary least squares (OLS) model on the right. We notice that the variables enter the model one at a time, and that their coefficients progressively increase until they stabilize around the OLS values – but, in some cases, the values can decrease, as the weight of another variable, providing similar information, increases. They may also change sign.

One may be tempted by a poor man's sparse regression: start with the OLS model and discard the variables with the smallest coefficients. This "hard thresholding", and a variant, soft thresholding (truncate and shrink the remaining coefficients towards zero, to remove the discontinuity), can be justified if the predictors are orthogonal (see [20], table 2.3) and soft thresholding is often used as a step in the algorithms computing the regularization path.

Figure 14: Regularization path, from the most penalized, intercept-only model on the left, to the non-penalized, ordinary least squares model, on the right.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

In summary, the lasso has the following advantages:

- The model is simple (linear) and interpretable (we can limit the number of variables entering the model);
- The model can deal with a very large number of variables and is robust to collinearity;
- The model is unlikely to overfit the data;
- The algorithm is very fast, and the resulting model is small (just a linear relation).

However, it also has one limitation:

- It does not capture non-linear relations or interactions.

For more details about the lasso and its generalizations, please refer to [20].

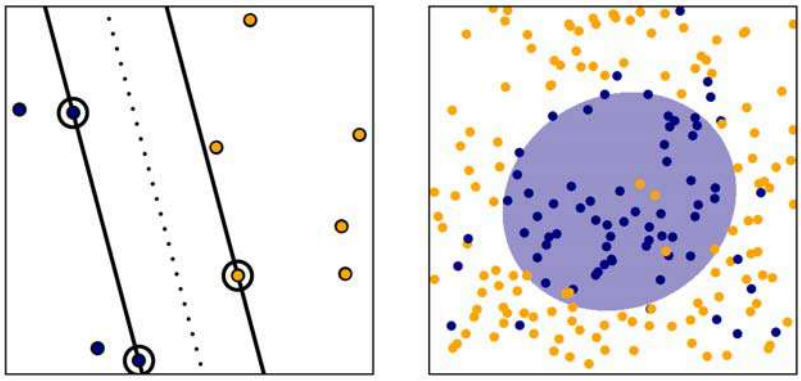
Support vector machines (SVM)

Support vector machines (SVM) are typically used for classification problems: given two clouds of points (e.g., companies that will go bankrupt in the near future and companies that will not) and we want to find a boundary that separates them.

Support vector machines combine two ideas.

- SVMs are **high-margin classifiers**: they do not look for a line that separates the two clouds of points, but a band, as large as possible (left figure). Notice that some of the data points touch the band: these are the **support vectors**. The other data points do not have any influence on the model and could be removed.
- SVMs transform the data by embedding the points into a higher-dimensional space, which tends to make the desired boundary linear – this is the **kernel trick** (right figure: the boundary is linear in x, y, x^2, y^2, xy). In practice, there is no need to give an explicit embedding: the Gram matrix ("kernel") suffices. In particular, this makes non-numeric data (text, graphs) or mixed data amenable to standard classification or regression methods. Radial basis function kernels are popular.

Figure 15: The two ideas behind support vector machines (SVM)



(a) Large margin classifier (b) Kernel trick

Source: Deutsche Bank Quantitative Strategy

The SVM optimization problem

$$\begin{array}{ll} \text{Find} & w, b \\ \text{To maximize} & 1/\|w\|_2^2 \quad \text{(width of the band)} \\ \text{Such that} & \forall i \in L^+ \; w'x_i + b \geqslant 1 \\ & \forall i \in L^- \; w'x_i + b \leqslant -1, \end{array}$$

can be written as a convex (quadratic) problem

$$\begin{array}{ll} \text{Find} & w, b \\ \text{To mimimize} & \|w\|_2^2 \\ \text{Such that} & \forall i \; y_i(w'x_i + b) \geqslant 1 \end{array}$$

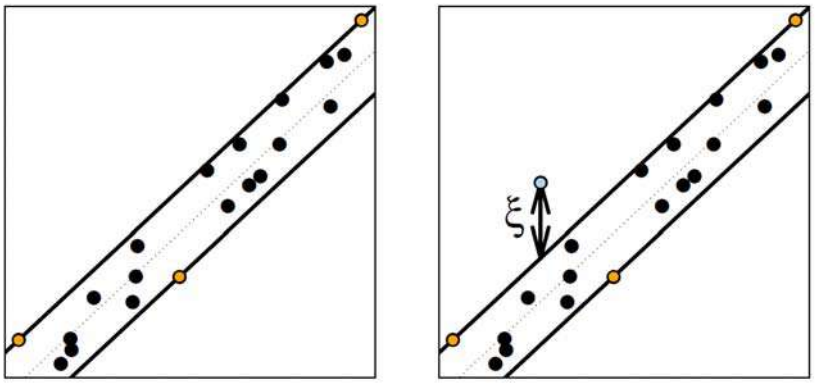
where $y_i = +1$ if $i \in L^+$ and -1 otherwise.

The **soft-margin SVM** allows misclassifications, but penalizes them.

$$\begin{array}{ll} \text{Find} & w, b, \xi \\ \text{To mimimize} & \|w\|_2^2 + c \sum \xi_i^2 \\ \text{Such that} & \forall i \; y_i(w'x_i + b) \geqslant 1 - \xi_i \\ & \forall i \; \xi_i \geqslant 0 \end{array}$$

Support vector machines can also be used for regression: the algorithm looks for a band, as narrow as possible, that contains the data; as for classification, one can add a penalty for points outside the band.

Figure 16: SVM for regression



(a) Narrow margin (b) Soft margin

Source: Deutsche Bank Quantitative Strategy

In summary, support vector machines have the following advantages:

- They are insensitive to the number of variables;
- They can capture non-linear relations and interactions.

However, they have the following limitations:

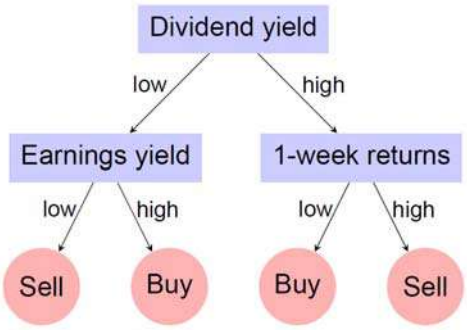
- They scale quadratically with the number of observations:[22] their use with large datasets is problematic;
- The model can be summarized by the kernel functions and the support vectors: it is not easily interpretable.

[22] Unless you use a linear SVM or resort to approximations, as in [29].

Decision trees and random forests

A **decision tree** is a list of questions (blue in the Figure 17), arranged in a hierarchical fashion, leading to decisions or forecasts (red).

Figure 17: A decision tree



Source: Deutsche Bank Quantitative Strategy

Decision trees are often built in a greedy,[23] top-down fashion:

- Select a predictor x_i;
- Select a breakpoint a_i;
- Cut the dataset, using the chosen predictor and breakpoint, into two new datasets, $\{ x : x_i \le a_i \}$ and $\{ x : x_i > a_i \}$ and proceed, recursively;
- Stop when the leaves are too small, the tree too deep, or there is no good predictor to break on;
- Prune the tree to avoid overfitting.

Various criteria are used to select the variable and its break point:

- Gini impurity[24] (equivalently, the Herfindahl index or Tsallis entropy),[25] $1 - \Sigma p_i^2$, used in CART;
- Information gain (also called Shannon entropy), $\Sigma\ -p_i \log p_i$, used in C5.0;
- Variance reduction;
- χ^2- or F-tests, used in CHAID;
- Permutation tests, used in conditional inference trees.

[23] Greedy algorithms tend to be suboptimal. There are non-greedy alternatives to decision trees, e.g., Bayesian rule lists [44, 27, 50].
[24] Unrelated to the Gini coefficient.
[25] Here are the definitions of all those quantities.

$$\text{Gini impurity} = 1 - \sum p_i^2$$
$$\text{Herfindahl index} = \sum p_i^2$$
$$\text{Shannon entropy} = -\sum p_i \log p_i$$
$$\text{Tsallis entropy} = \frac{1}{\alpha - 1}\left(1 - \sum p_i^\alpha\right)$$
$$\text{Rényi entropy} = \frac{1}{1-\alpha} \log \sum p_i^\alpha$$

Decision trees are very noisy: if you change the data a little bit (by adding or removing a few months, a few stocks, or a few variables), the results are completely different.

Decision trees have their uses, though. For instance, given a good but overly complex model, too complex for us to understand what it does, we can train a decision tree on its output, to see what it does. Since the decision tree is trained on the output of the black-box model and not on real data, we can generate much more training data, and therefore obtain a more stable result.

In summary, decision trees have the following advantages:

- They are eminently interpretable;
- They can capture interactions.

But they also have limitations:

- They are too noisy to be relied on;
- Though they can model interactions, the greedy algorithm used to build them can fail to capture those interactions.

Random forests

Decision trees are also useful as building blocks for more complicated models: for instance, a **random forest** is an ensemble of decision trees:

- Use a new random sample for each tree;
- For each node, select the variable to split on from a random subset of variables instead of all the variables;
- Do not prune the trees;
- Average the forecasts.

Random forests and, more generally, ensemble methods, look appealing, but they pose a few problems. First, they can be very unwieldy: since a forest contains hundreds or thousands of trees, it is cumbersome to store and time-consuming to fit and use. The situation gets worse when we need several forests, e.g., if we want to estimate a model on a moving or expanding window.

Interpretability is another problem: while a single tree is very intuitive, it is difficult to grasp what hundreds of them are saying.

Some suggest looking at the **importance** of each variable, i.e., how often each variable appears in the trees. Unfortunately, if several variables contain similar information (e.g., different measures of "value"), their importance will be diluted. Some suggest **partial dependency plots**, to show the influence of a variable when everything else remains constant: this makes the assumption that the predictors do not interact – but decision trees do model those interactions.

An often-touted advantage of random forests is their inability to overfit the data: as shown on Figure 18, in which we use a random forest (with no constraint on the tree depth) to forecast future returns and build quintile portfolios from those forecasts, the risk is still very real.

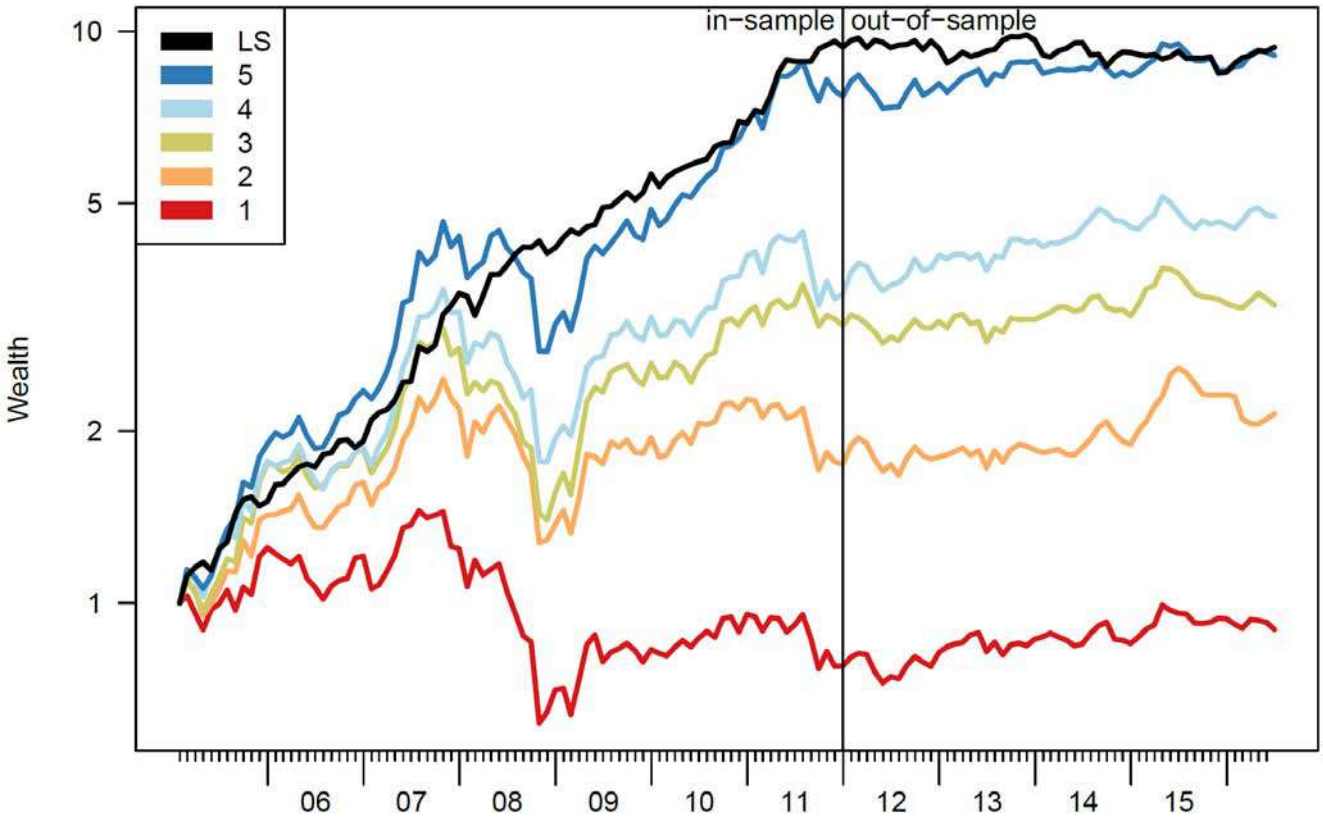
In summary, random forests have the following advantages:

- They can model interactions;
- They are less noisy than decision trees.

However, they also have limitations:

- They are no longer interpretable;
- They are often too large to be used.

Figure 18: Despite popular belief, random forests can overfit



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Boosting

Boosting is another ensemble algorithm, like random forests, which combines simple models ("base learners" or "weak learners"), e.g., decision trees. Contrary to random forests, those simple models are not trained on completely random subsets of the data: boosting puts more weight on observations not well accounted for by the previous models. This usually results in much smaller ensembles.

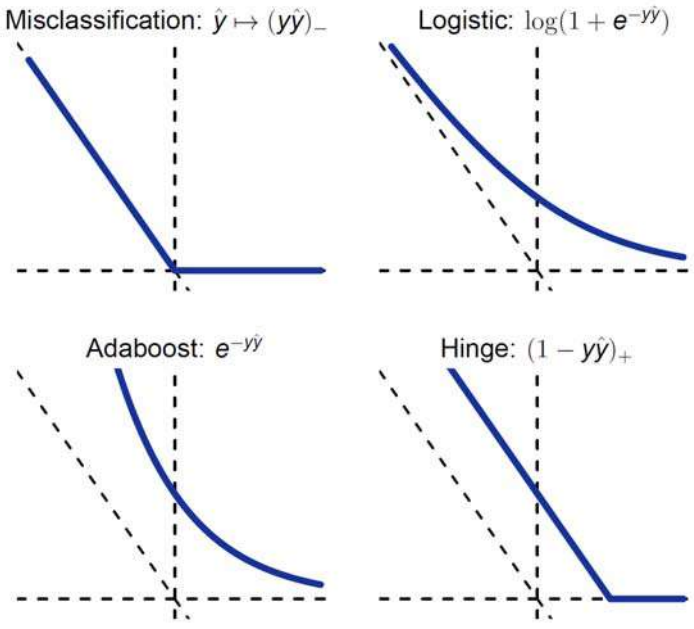
More precisely, the algorithm goes as follows:

- Fit a weak learner f to the data, giving more weight to currently misclassified observations;
- Move the model F towards this weak learner, $F \leftarrow F + \alpha f$;
- Iterate a few times.

We can choose the weak learners, how we compute the weights, the step size α, and the number of iterations. For instance, **Adaboost** uses the following:

- Weak learners: shallow decision trees;
- Exponential loss: $\text{Loss}(\hat{y}, y) = \exp(-y\hat{y})$, $y \in \{\pm 1\}$;
- Weight = $\exp(-y\hat{y})$;
- Stepsize α: such that $F + \alpha f$ minimizes the loss;
- Number of iterations: less than 10.

Figure 19: Loss functions, plotted for y = +1



Source: Deutsche Bank Quantitative Strategy

The **Viola-Jones algorithm** was one of the early applications of boosting: the camera in your mobile phone may be using it to detect faces. As weak learners, it uses simplistic feature detectors, vertical or horizontal edges, and vertical, horizontal or diagonal bars (or lines); they are combined with AdaBoost.

To detect faces, it scans the whole image to detect the first feature, e.g., a long horizontal dark bar, corresponding to the eyes.

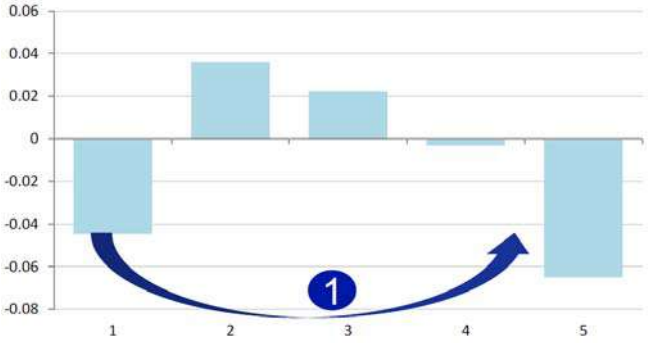
That gives a large set of potential faces, which has to be pruned. Among those candidate faces, it looks for the next feature, e.g., a bright area with darker areas on the left and the right, corresponding to a nose.

The algorithms iterates, each new feature further reducing the set of candidates. It is fine-tuned to reduce the number of false negatives.

Boosting can also be interpreted as *gradient descent* on a space of functions (indeed, the final, combined model is in the convex hull of the weak learners): this is *gradient boosting*.

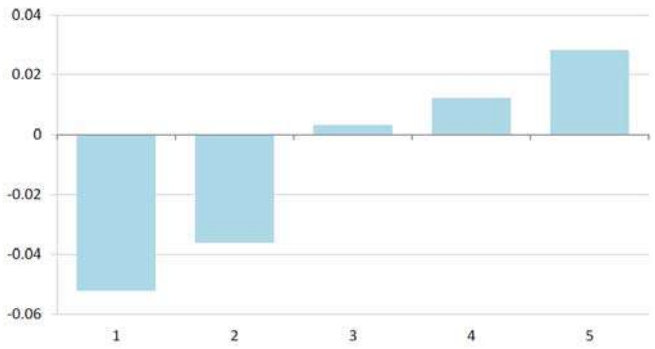
Deutsche Bank Quantitative Research has used boosting in the past, for instance in the N-LASR model [45, 47, 46]: the weak learners were continuous piecewise affine functions on the quintiles of one of the predictors.

Figure 20: N-LASR model, Sloan's accruals factor, figure 4 from [46]



Source: Bloomberg Finance LLP, Compustat, IBES, Russell, S&P, Thomson Reuters, Worldscope, Deutsche Bank Quantitative Strategy

Figure 21: N-LASR model, Sloan's accruals factor after transformation, figure 5 from [46]



Source: Bloomberg Finance LLP, Compustat, IBES, Russell, S&P, Thomson Reuters, Worldscope, Deutsche Bank Quantitative Strategy

Boosted trees have the following advantages:

- They can model interactions;
- They are much smaller than random forests;
- They are usually sparse, and less likely to overfit the data than random forests.

However, they still have a limitation:

- Although the individual trees are interpretable, their combination is not.

In this report, we will look at two examples of boosing, with decision trees (*xgboost*), and with 1-variable non-linear models (*mboost* or *gamboost*).

Since non-linear models are not as widely known as they should be, we review them in the next section.

Generalized additive models (GAM)

Let us examine another way, besides trees and kernels, to capture non-linear relations. To simplify the exposition, let us first assume there is only one predictor, x, i.e., we want to fit a model of the form $y = f(x) + \varepsilon$, where f is unknown – in other words, we want to smooth a cloud of points.

Local regression fits a linear model on a moving window: the envelope of the resulting lines gives a smooth approximation of the cloud of points (Figure 22). Instead of a moving window, one can use Gaussian weights (loess) or some other "kernel"[26]; instead of a linear model, one can use a constant model (Nadaraya–Watson kernel smoothing).

An alternative is to express f as a sum of "simpler" **basis functions**, $f = \Sigma f_i$, by minimizing

$$\text{Loss}(y, x, \beta) = \left(y - \sum \beta_i f_i(x)\right)^2.$$

Here are a few examples of basis functions (Figure 23):

- Locally constant functions;
- Locally affine functions;
- Locally polynomial functions (splines);
- Trigonometric functions (Fourier analysis);
- Wavelets;
- etc.

Yet another way of smoothing a cloud of points is to solve the optimization problem

$$\begin{array}{ll} \text{Find} & f \in \mathscr{C}^2 \\ \text{To minimize} & \sum (y_k - f(x_k))^2 + \lambda \int (f'')^2 \end{array}$$

The solution turns out to be piecewise polynomial – these are **splines**.

Notice that the parameter λ, or the number of basis functions, or the bandwidth of the kernel play a regularizing role, similar to what we saw with penalized regression.

Generalized additive models (GAM) generalize smoothing to several predictors by smoothing one variable at a time. The model is of the form

$$y = \sum_i f_i(x_i) + \varepsilon$$

where the f_i are unknown functions of one variable. Those models are usually estimated by backfitting:

[26] This is not the same notion of kernel as for SVMs: here, a kernel is a weight function.

- Assume that all the functions, except one, are known;
- Estimate that function, f_j, by smoothing the residuals,

$$f_j \leftarrow \text{Smooth}\left(y - \sum_{i \neq j} f_i\right)$$

- Iterate until convergence.

It is possible to add interactions, e.g., by considering models of the form

$$y = \sum_i f_i(x_i) + \sum_{i \neq j} g_{ij}(x_i, x_j) + \varepsilon.$$

Generalized additive models (GAM) have the following advantages:

- They can model non-linear relations, and even pairwise interactions;
- As linear models, they are easy to interpret.

However, they share the main problem of unpenalized models:

- They overfit the data when there are too many or collinear variables.

Generalized additive models can also be used as base learners for boosting.

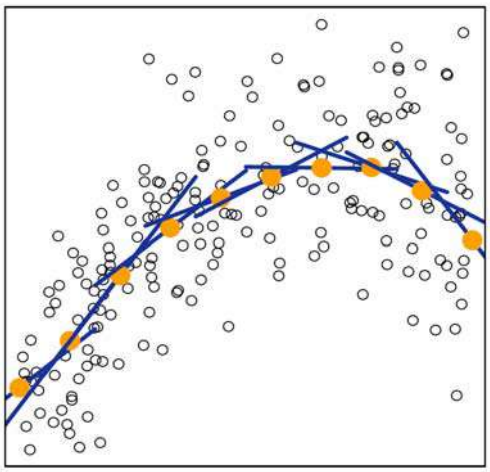
Boosted GAMs (sometimes called "mboost" in the rest of this report – that is the implementation we use) have the following advantages:

- They can model non-linear relations and even pairwise interactions;
- They remain linear: they are easy to interpret;
- They as sparse.

There is still one limitation, though:

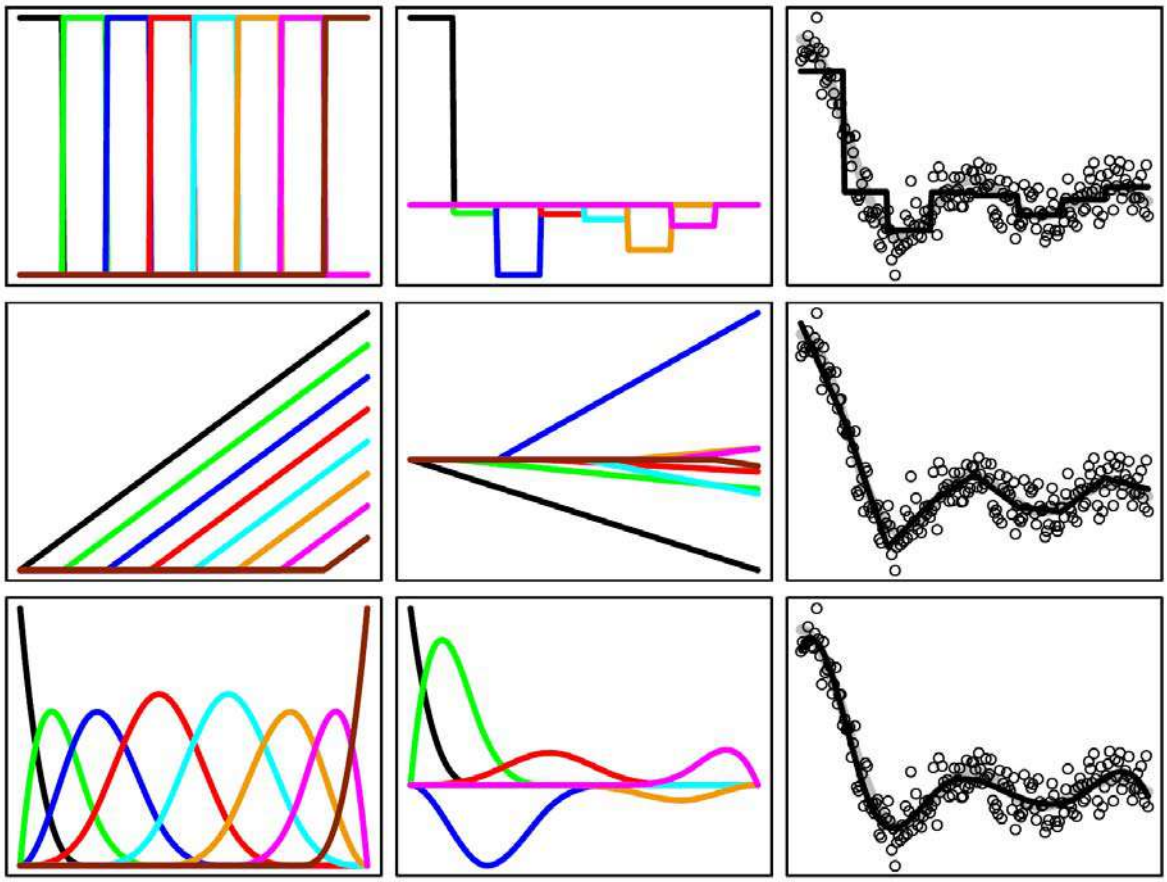
- As the number of boosting steps increases, the computations can become very time-consuming.

Figure 22: Local regression (kernel smoothing, loess): for each x, fit a weighted linear model, with more weight for observations close to x.



Source: Deutsche Bank Quantitative Strategy

Figure 23: Smoothing with basis functions. There is one family of basis functions per row (locally constant, locally linear, splines); the first column shows the basis functions; the second how they enter the model; the third one the data (points), the ground truth (grey) and the fitted model (black line).



Source: Deutsche Bank Quantitative Strategy

The dangers of machine learning in Finance

To avoid overfitting, machine learning suggests the following procedure.

- Split the data into a **training set** and a **validation set**;
- Fit the model on the training set and check its performance on the validation set;
- Fine-tune the parameters; choose the best model;
- Use a third **test set**, at the very end, just once, to estimate the actual performance of the model;
- Re-fit the model on the whole dataset.

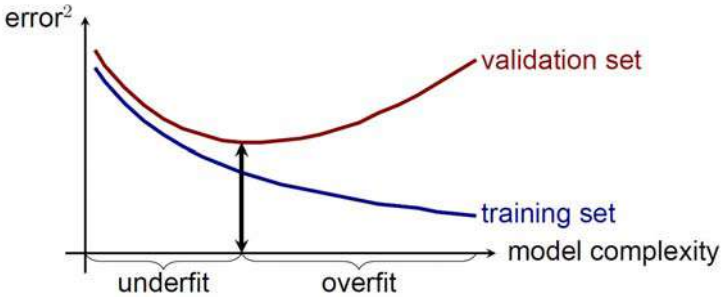
In practice, we use the first 70% to 90% of the data as training set, and the remaining, i.e., more recent observations, as validation set. The division is only based on time: since all stocks are subject to the same common, global influences, using countries or sectors would create dependencies between the training and validation sets.

The test set discipline is difficult to enforce: the test set should be used only once, at the very end, and the model should not be tweaked after that. Unfortunately, when presenting the final results, someone (internal or external client) often suggests some modification – following that suggestion would taint the test set and invalidate the conclusions. It may be easier to forego the test set during the study and, when we are sure it is finished, start to gather new data, for 3 to 6 months, to use as a test set.

If we fit models of varying complexity on the training set, we can choose that that gives the smallest error on the validation set, as suggested by Figure 24 and Figure 25.

With real data (Figure 26), the frontier between overfit and underfit is less clear.[27]

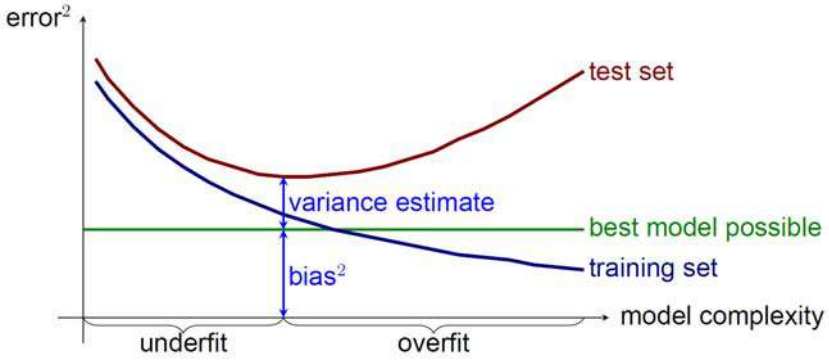
Figure 24: Model validation: choose the model complexity to minimize the error on the validation set



Source: Deutsche Bank Quantitative Strategy

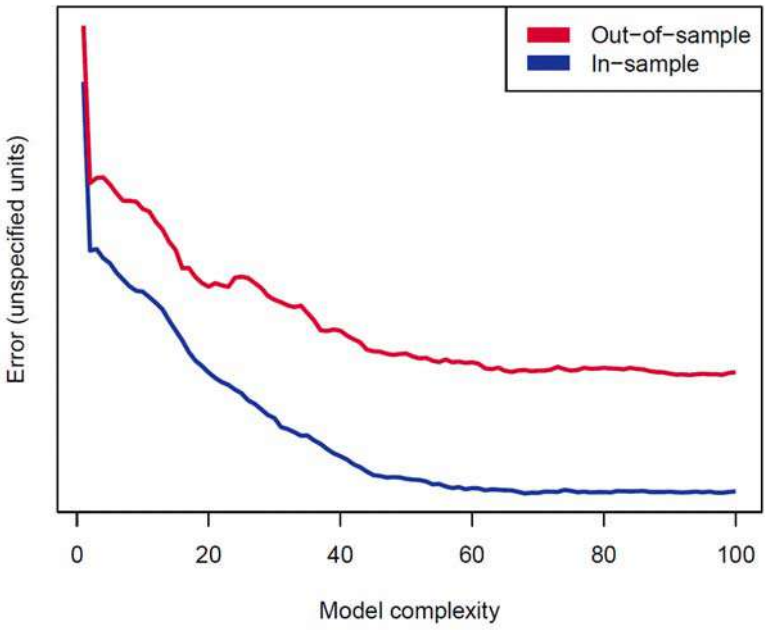
[27] For more examples on real data, check http://lossfunctions.tumblr.com/.

Figure 25: Bias and variance in model validation. As in Figure 12, we can decompose the (squared, estimated) error into (squared) bias and (estimated) variance. For more details, see [1].



Source: Deutsche Bank Quantitative Strategy

Figure 26: Error of a penalized regression as the complexity of the model is allowed to increase, with actual data. There is not always a clear frontier between under- and overfit: here, the out-of-sample (red) error has no clear minimum.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

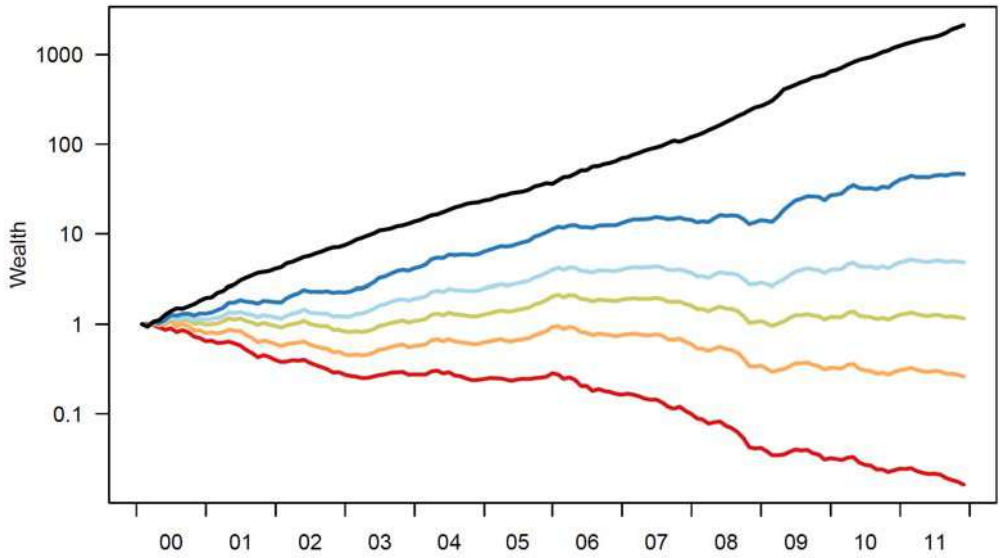
Cross-validation refers to variants of the fixed training/validation split: the training and validation sets are randomly selected several times, to provide a better estimate of the out-of-sample performance of the model. The most common is **k-fold cross-validation**:

- Partition the dataset into k parts;
- Use k−1 parts as a training set and the remaining part as a validation set: fit the model on the training set and estimate its performance on the validation set;
- Do this k times;
- Use the average performance as an estimate of the out-of-sample performance.

That approach, with financial data, usually leads to models that are excessively complex, difficult to interpret and whose in- and out-of-sample performance differs wildly. **It is a recipe to overfit the data.** Indeed, it assumes that the validation set and the training set are independent: but since many financial quantities change very slowly with time, and do not vary much for stocks in the same sector or country, this is an unreasonable and dangerous assumption (Figure 27). That lack of independence is also the reason why algorithms supposedly robust to overfitting, such as random forests, can overfit (as we saw a few pages ago).

In short: in finance, cross-validation does not work.

Figure 27: In-sample performance of the XGBoost model, with hyperparameters fine-tuned using cross-validation: annual returns reach 90%, and the information ratio is 8.5. While the out-of-sample performance remains decent (15%, 2.0, not shown on the plot), the difference casts doubts on the robustness of the resulting investment strategy.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

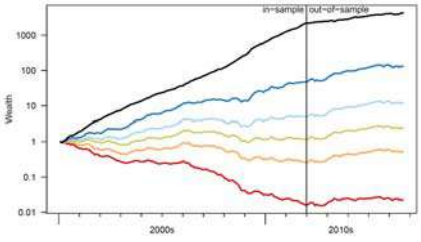
Since overfitting *will* occur, we need a way of identifying and correcting it.

Some people suggest comparing the distribution of the results of the strategy in- and out-of-sample. However, as shown in Figure 24, we do expect the in-sample returns to be higher, even if the model overfits the data as little as possible – the red curve is almost always above the blue one.

Another idea is to try to interpret the model: if we understand what it does, and if it makes sense from a financial point of view, the model is unlikely to be overfitting. Unfortunately, this is only possible for interpretable models (there are very few of them) and requires human input for each model.

One can also be more conservative and set the hyperparameters to values known (beforehand) to avoid overfitting: for instance, for penalized regression, one could stop on the regularization path once 10 variables have entered the model. Besides being purely empirical, that approach has a few drawbacks: not only is it almost guaranteed to underfit the data, it only works for hyperparameters with a regularizing effect – for instance, you cannot use it to choose the kernel in a support vector machine.

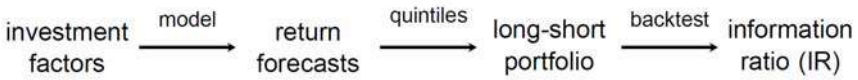
Figure 28: Performance of the XGBoost model, with hyperparameters fine-tuned using cross-validation.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Practice

In the second part of this report, we will use the algorithms introduced in the first part to combine the investment factors in our database (Figure 30 and Figure 33) and forecast future returns; we will evaluate the quality of these forecasts by using them to build a quintile long-short portfolio and by computing its out-of-sample information ratio.



While this practice is common, this goes against some of the recommendations we made earlier.

- We are forecasting returns, but we look at the information ratio of a strategy built from those returns, i.e., the model tries to optimize something that does not have a direct meaning, from a business point of view. This suboptimal approach has, however, a few advantages:
  - We can use off-the-shelf, efficient and reliable implementations of those algorithms;
  - Measuring the precision of the return forecasts is much faster than computing the performance of the resulting strategy: this significantly speeds up the optimization at the heart of those algorithms;
  - It may also have a regularizing effect, and reduce overfitting
- Since there is no good alternative, we are using separate training and test sets. We will see that the resulting models do overfit the data: they combine and/or transform the investment factors in unnatural and potentially dangerous ways, even if it seems beneficial out of sample.

Here is the structure of this second part.

- We first examine possible transformations of the input data, before estimating the model: should we winsorize, quantize, normalize, uniformize? Should we do this for both the predictors and the variable to predict?
- We then define a benchmark model, to check how much, if at all, machine learning improves on simplistic models.
- After those preliminaries, we examine the machine learning algorithms introduced in the first part, and focus on the choice of hyperparameters.

  We will see that the traditional machine learning approach, splitting the data into training and validation sets (or into in- and out-of-sample periods), to choose the best hyperparameters, gives very good out-of-sample performance, but very suspicious models – we will rein in the overfitting zeal of the algorithms by requiring that the models remain interpretable.

- Following this idea, we also add one more model: a linear regression, but with a constraint on the sign of the coefficients dictated by our investment knowledge;
- Finally, we compare all those models: the lasso and boosting have a comparable performance, but are outranked by the constrained regression; we also explain this difference.
- In a conclusion, we lay out the recipe for machine-learning-based model construction, repeat our warning about the blind use of machine learning tools in finance, but also, on a more positive note, highlight possible uses of those tools.

Data pre-processing

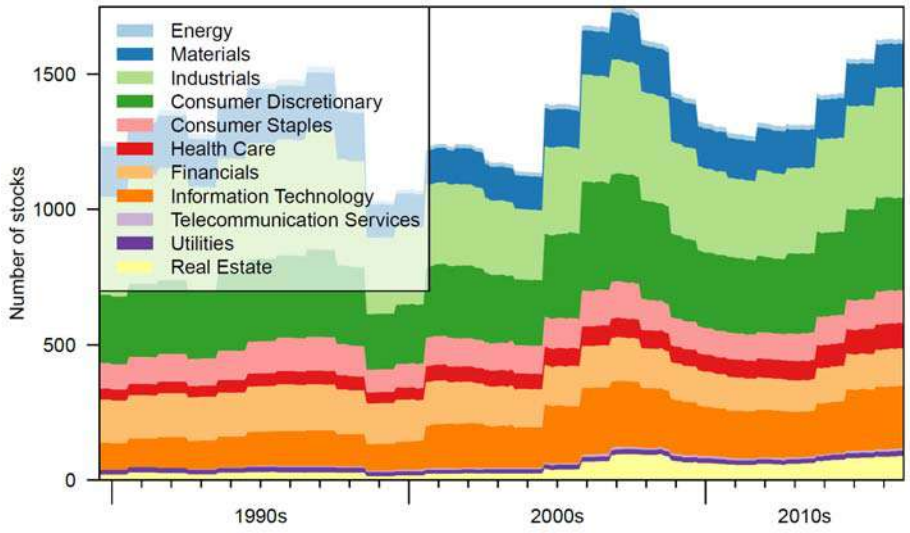
Data

In the following pages, we consider the Japanese equity market (proxied by the S&P BMI Japan constituents) as an example. Our usual investment factors (details on the next pages) are stock characteristics commonly used to select stocks. The data is monthly, and the portfolios rebalanced every month. We used 2000–2011 as in-sample period, and 2012–present as out-of-sample. We did not use the 1990s because some of our investment factors had too little coverage (Figure 31 and Figure 32).

Figure 34 to Figure 39 show the performance of quintile portfolios, built by ranking stocks according to those signals; the first quintile (low values) is in red, the fifth (high values) in blue, and the corresponding long-short portfolio (long the fifth quintile, short the first) in black.

Figure 40 shows the relation between each of those factors and forward returns – those relations need be neither linear nor even monotonic.

Figure 29: Universe size and composition



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 30: Investment factors

Growth
Asset Growth
EPS Growth
Sales Growth
EBITDA Growth
IBES 5Y EPS growth/stability
EPS 5Y growth
IBES FY1 mean CFPS growth
IBES FY1 mean DPS growth
IBES FY1 mean EPS growth
IBES Estimate Long Term Growth for EPS
IBES FY2 mean DPS growth

Momentum
12-1 month Total return
Total return, 252D
Total return, 63D
Total return, 126D
3-6 month Moving Average cross over
Price-to-52 week high
Price-to-52 week low

Quality
IBES FY1 EPS dispersion
IBES FY2 EPS dispersion
IBES NTM EPS dispersion
Accruals
Berry Ratio
Capex to Depreciation
Capex-to-Assets
Cash Flow-to-Assets
Change in Debt
Current Ratio
EBITDA Margin
Gross Margin
Payout on trailing operating EPS
Return on Assets
Return on Equity
Return on Invested Capital
CFROI
Sales-to-Total Assets
Altman's z-score
Merton's distance to default
Employee Growth
Debt-to-Equity
Debt-to-Market value of Equity
Quick ratio
Gross Profit-to-Assets
Piotroski's score
Piotrsoki's Profitability
Piotrsoki's Liquidity
Piotroski's efficiency
Piotroski's continuous score
Piotroski's score - 5 values
Graham
Grantham
G-score
M-score
5-year ROE Volatility
Quality Minus Junk
SGA-to-Sales
EBIT-to-Tangible Capital
Tier1 Capital Ratio (for Banks only)
Cash to Market cap
RnD Expenses to Sales
RnD Expenses to Market capitalization
Assets to Book value
Change in Asset Turnover
Change in Gross Margin
Change in Current Ratio
Change in ROA

Risk/Reversal
Total return, 21D
Total return, 1260D
Lottery Factor
Skewness, 1Y daily
Realized vol, 1Y daily
Idiosyncratic Volatility
Beta
Total return, 5D

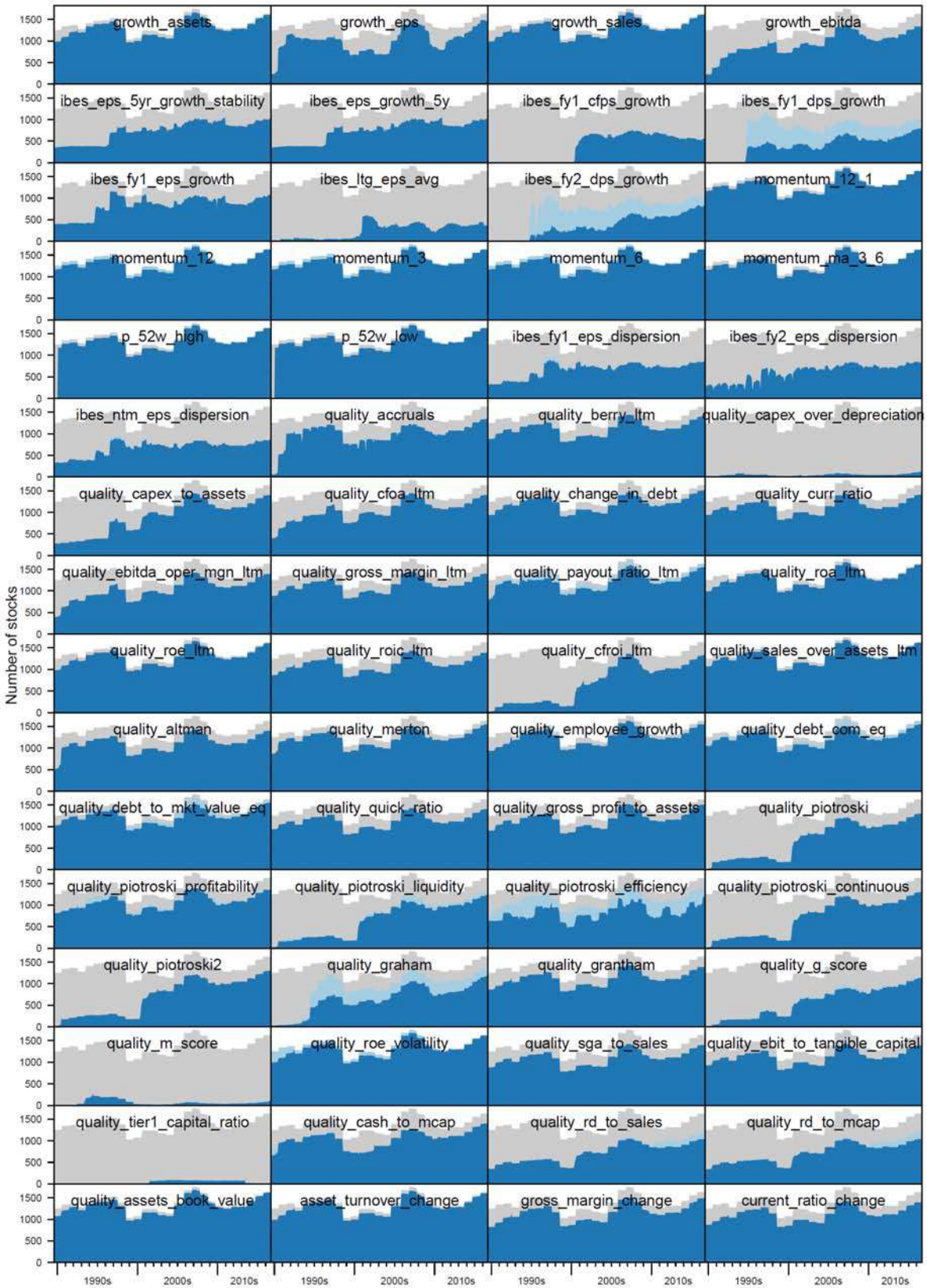
Sentiment
Broker Demand over Mcap
Broker Demand over Supply
Lender Demand over Mcap
Lender Demand over Supply
Supply over Mcap
IBES FY1 Mean CFPS Revision, 3M
IBES FY1 Mean DPS Revision, 1M
IBES FY1 Mean DPS Revision, 3M
IBES FY1 Mean EPS Revision, 1M
IBES FY1 Mean EPS Revision, 3M
IBES NTM EPS Revision, 1M
IBES NTM EPS Revision, 3M
IBES FY1 EPS up/down ratio, 1M
IBES FY1 EPS up/down ratio, 3M
IBES EPS up/down ratio, NTM
IBES FY1 Mean ROE Revision, 1M
IBES FY1 Mean ROE Revision, 3M
IBES FY1 Mean SAL Revision, 1M
IBES FY1 Mean SAL Revision, 3M
IBES NTM SAL Revision, 1M
IBES NTM SAL Revision, 3M
IBES Sales up/down ratio, NTM
IBES LTG Mean EPS Revision, 1M
IBES LTG Mean EPS Revision, 3M
Target price implied return
Recommendation average
Mean recommendation revision, 3M

Size
Market Cap (Small minus Big)
Abnormal Volume

Value
Cash Flow Yield, FY0
Cash Flow Yield, FY1
Dividend yield, FY1
Dividend yield, FY2
Dividend yield, NTM
Earnings yield, forecast FY1 mean
Earnings yield, forecast FY2 mean
NTM Earnings yield (IBES)
LTM Earnings yield (IBES)
Earnings Yield (Worldscope)
Buyback Yield
Dividend yield, trailing 12M
EV to EBITDA
FCF/EV
FCF Yield
Sales/EV
NTM Sales yield (IBES)
Sales to Price
Trailing Book yield
Est Book-to-price
Total Yield
Operating Cash Flow Yield

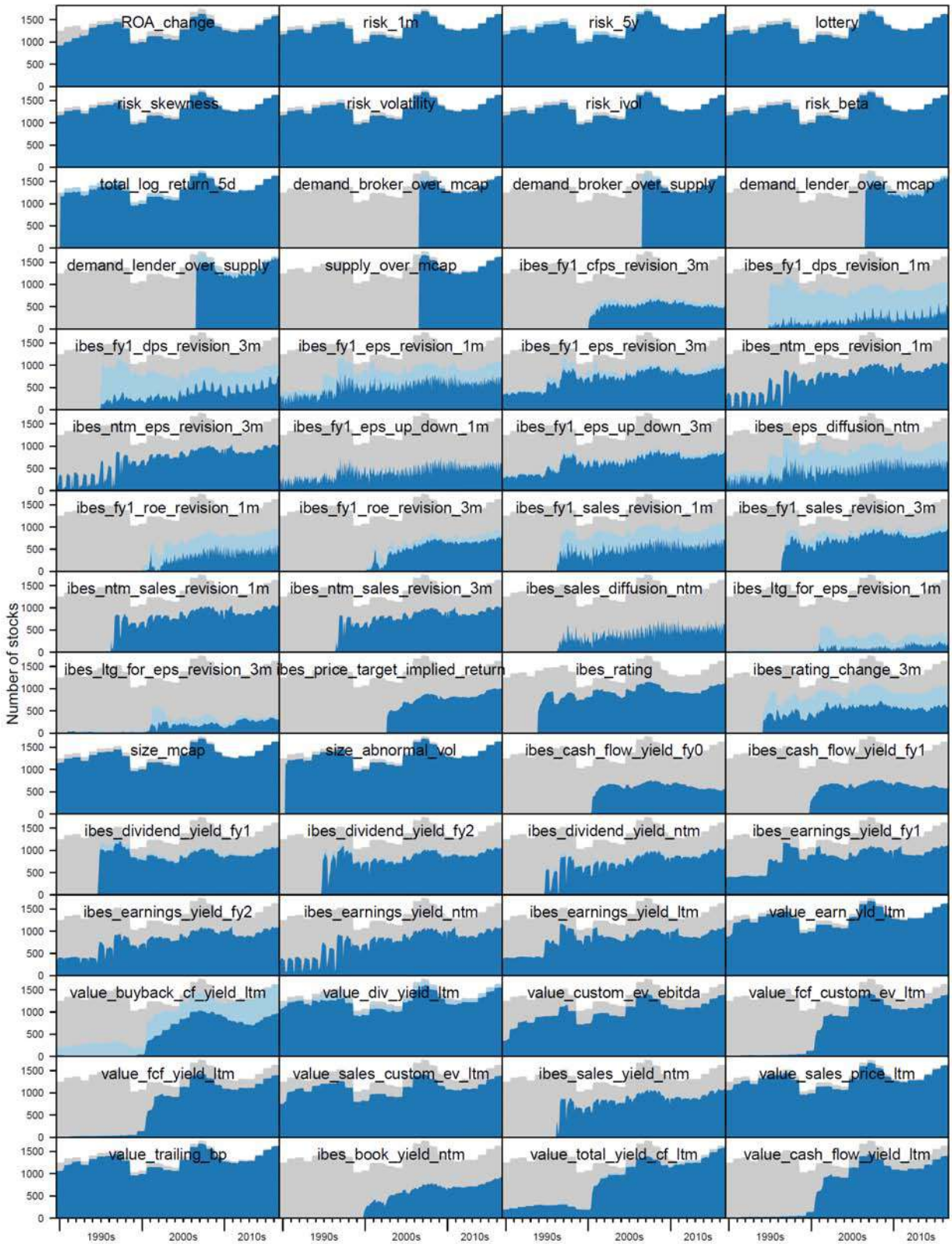
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 31: Coverage. Missing values are in grey, zeroes in light blue, non-zero values in dark blue.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 32: Coverage

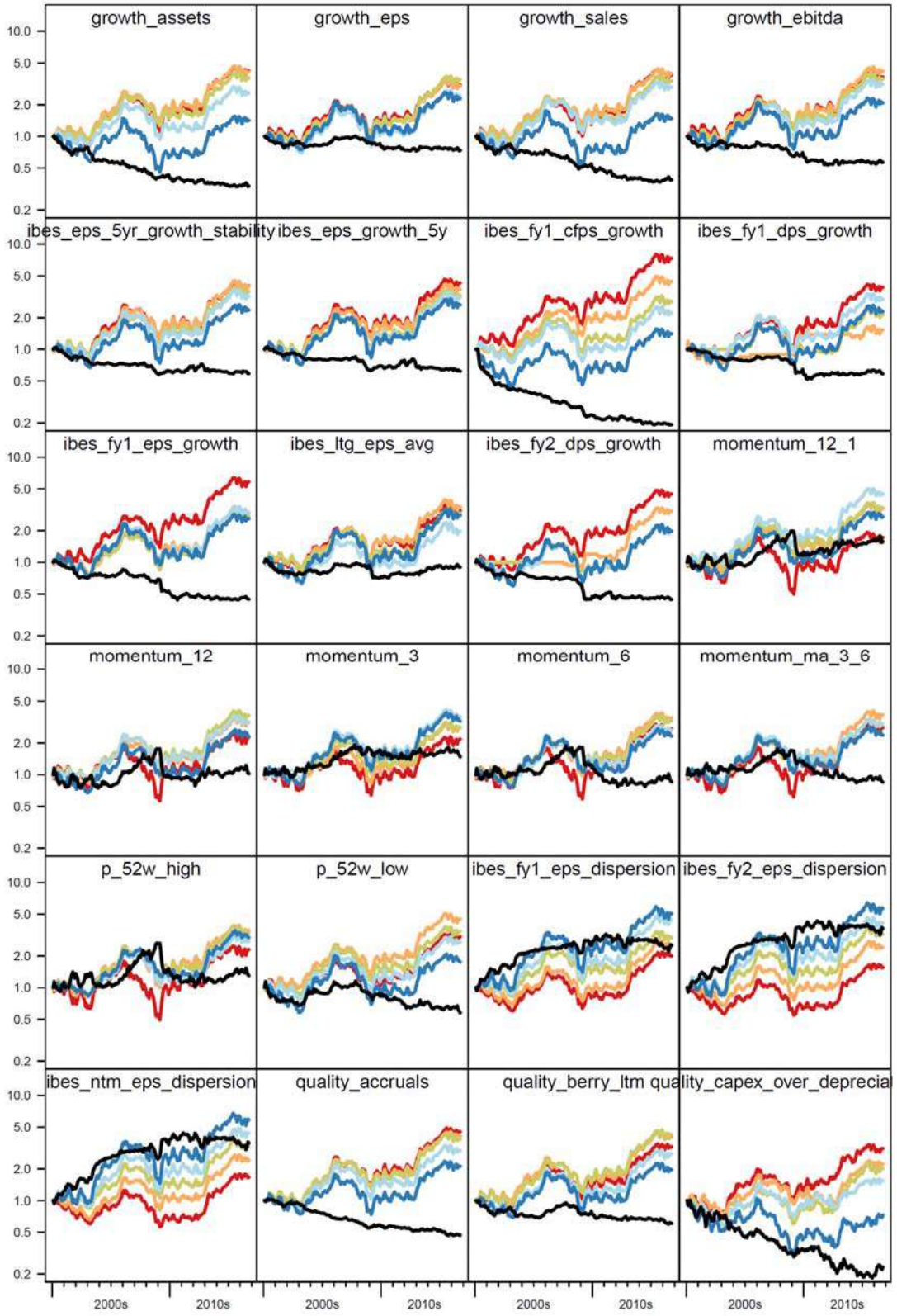


Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 33: Minimum spanning tree computed from the median cross-sectional correlation between our investment factors

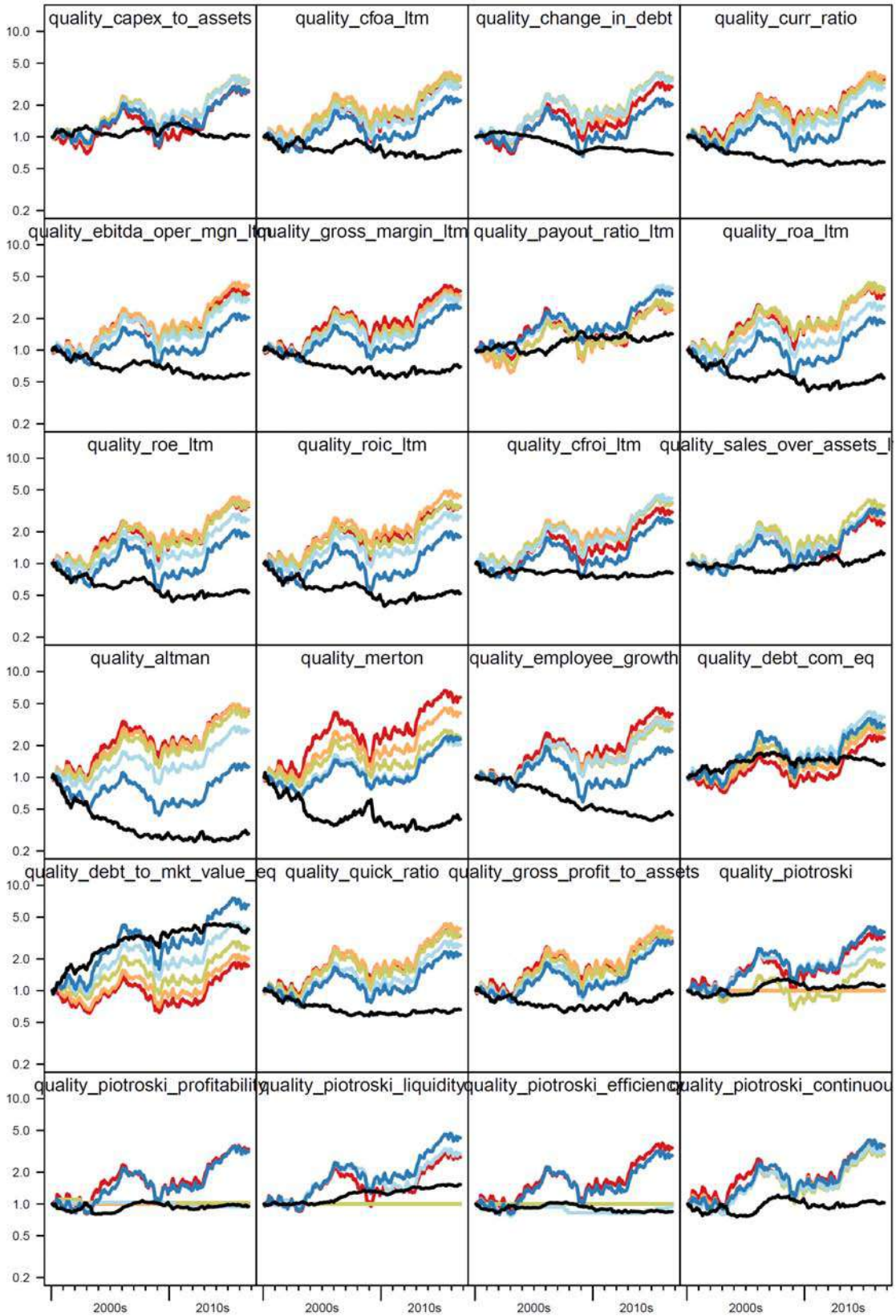
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 34: Performance of the raw investment signals



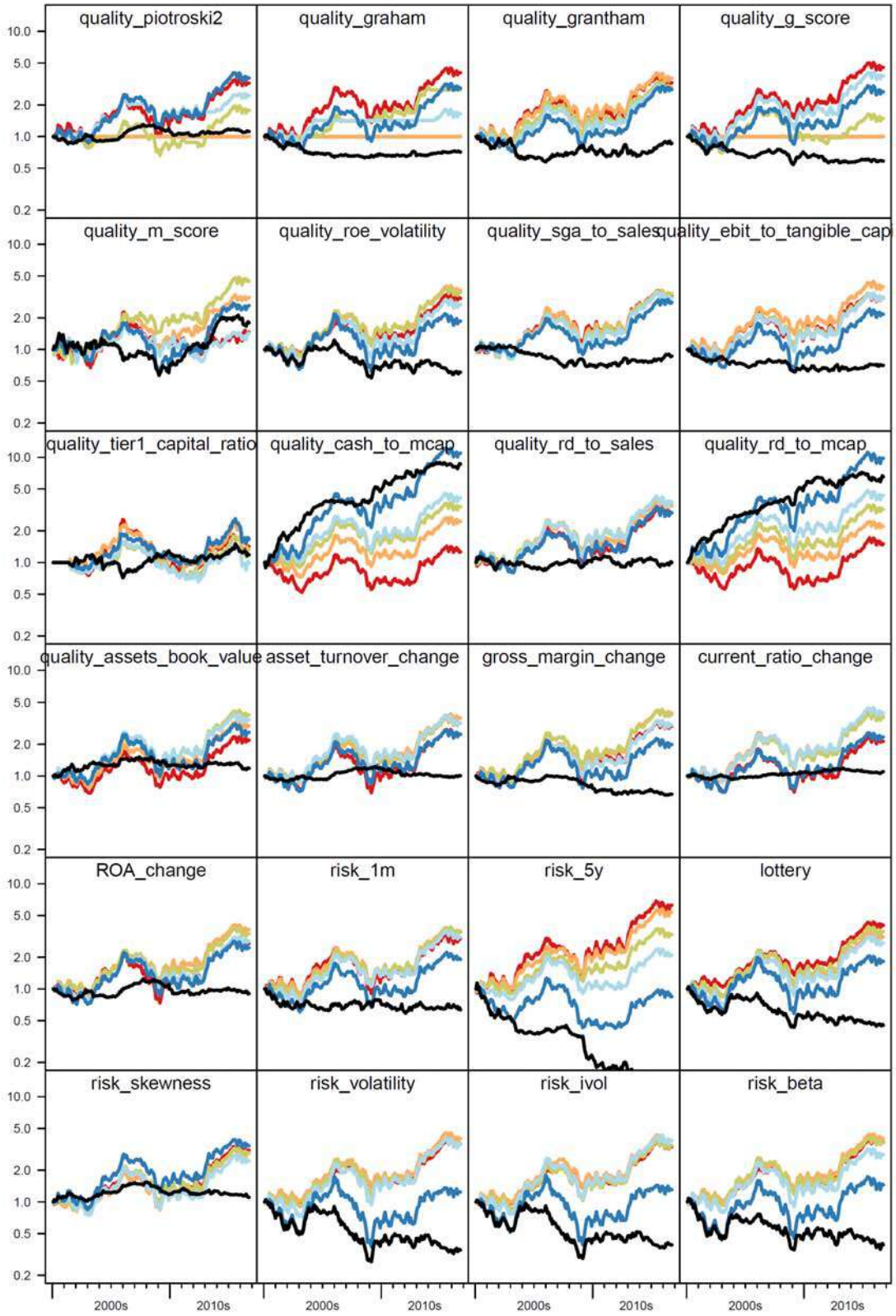
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 35: Performance of the raw investment signals



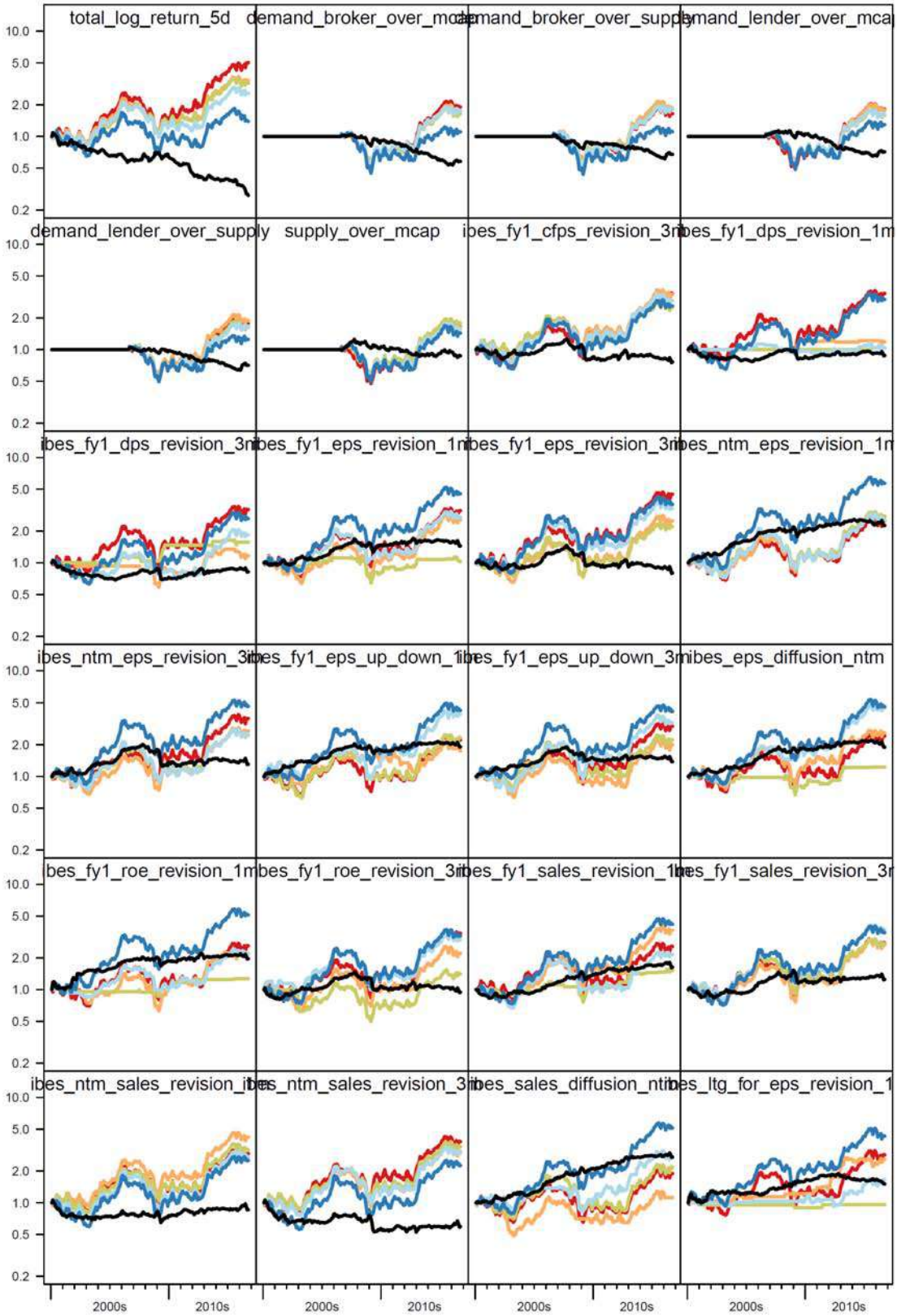
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 36: Performance of the raw investment signals



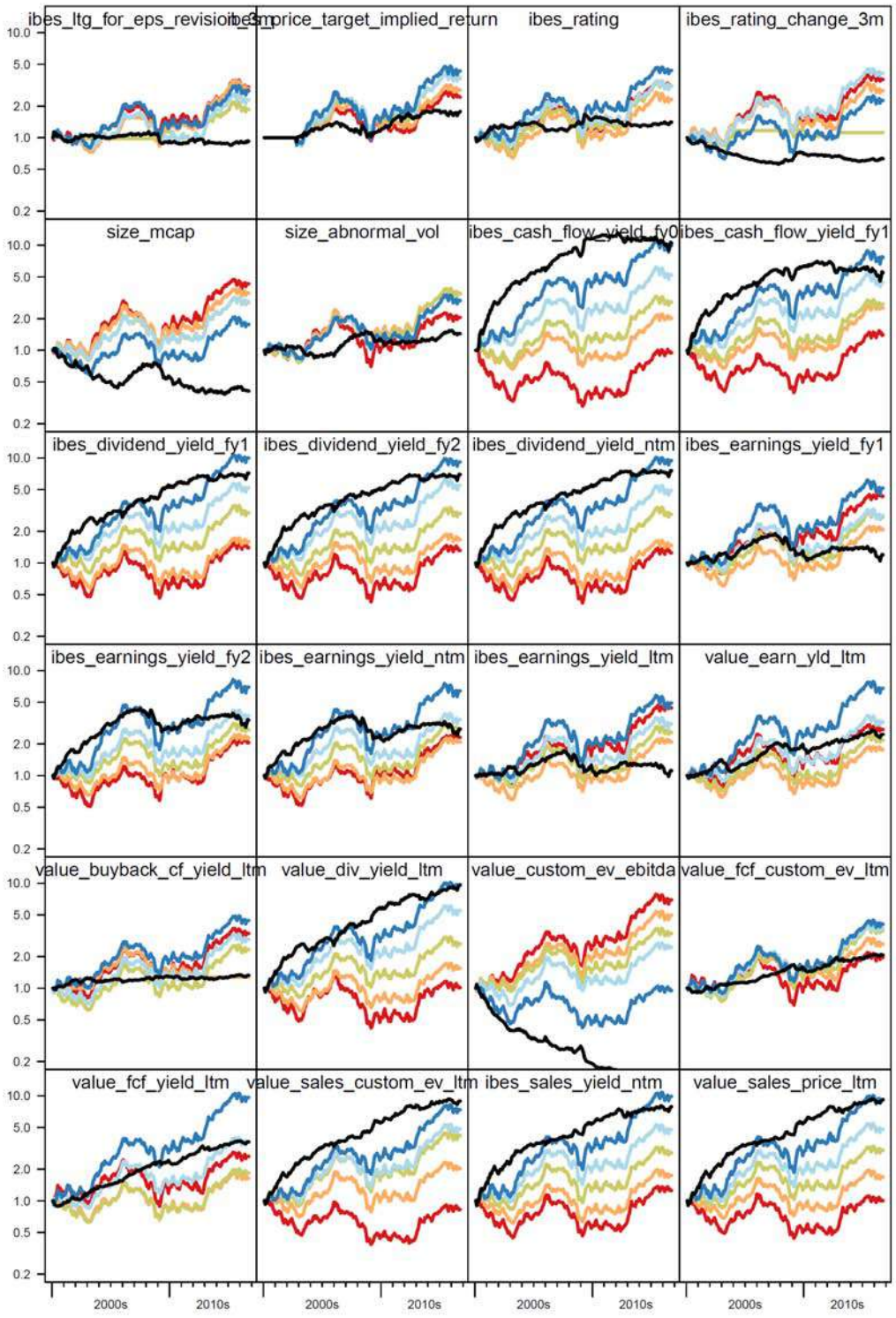
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 37: Performance of the raw investment signals



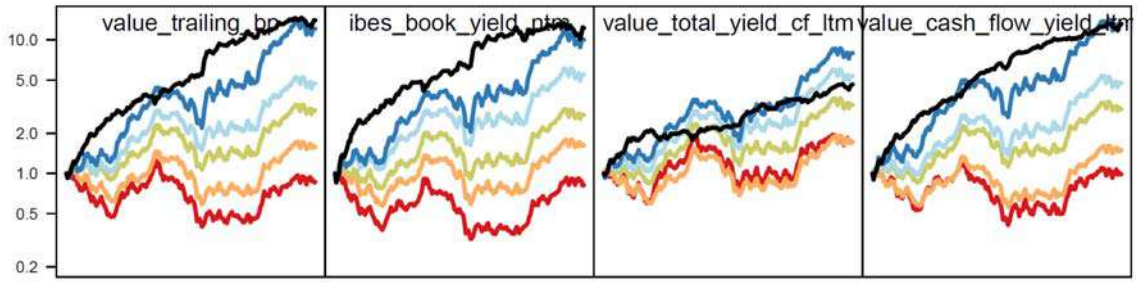
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 38: Performance of the raw investment signals



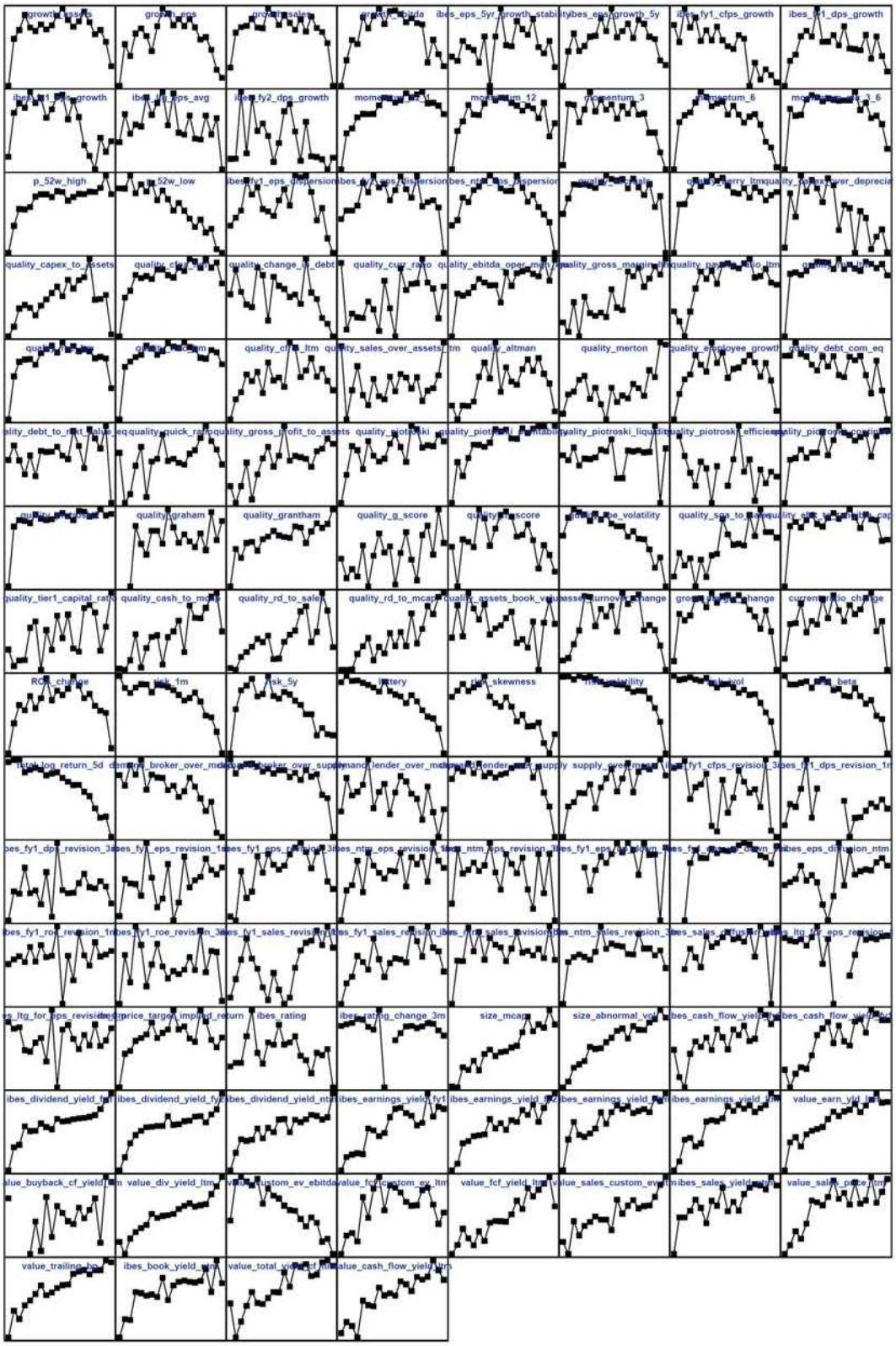
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 39: Performance of the raw investment signals



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 40: Relation between each variable (horizontal axis) and 1-month forward returns (vertical axis, average median in each vigintile)



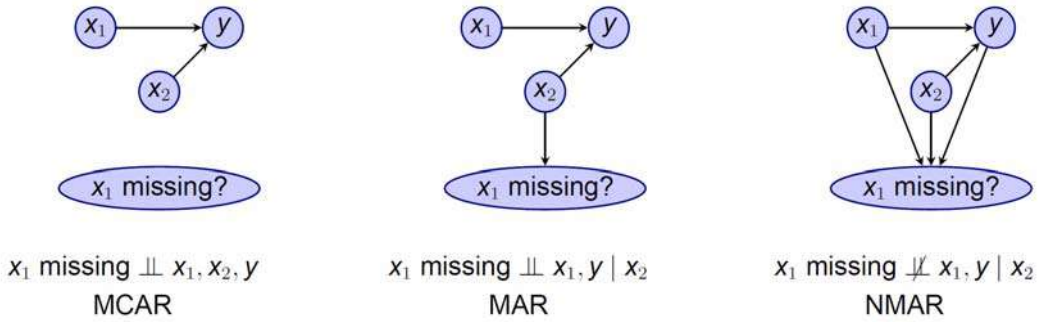
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Missing values

There are three types of missing values (Figure 41: The different types of missingness, as graphical models):

- **Missing completely at random** (MCAR): missingness is independent of everything else;
- **Missing at random** (MAR): missingness may depend on the other predictors but, conditioned on the other predictors, neither on the (unknown) missing value nor on the value of the variable to predict;
- **Not missing at random** (NMAR): missingness is informative; it may depend on the (unknown) missing value or on the variable to predict, even if we condition on the other predictors.

Figure 41: The different types of missingness, as graphical models



Source: Deutsche Bank Quantitative Strategy

There are many ways of dealing with missing values.

There are few of them, but one can choose algorithms that deal with missing values. Decision trees and most tree-based algorithms readily accept missing values: if the variable used in a node is missing, the algorithm simply takes both branches at the same time – in the end, it does not reach a single leaf but a group of leaves, whose forecasts can be combined. However, even for algorithms that can deal with missing values, many implementations cannot.

If there are few missing values, one can simply discard all observations with missing values. How much data can be thrown away depends a lot on the data: while discarding 5% of the data can already be worse than naive imputation, in some situations, the difference only becomes visible after removing 90% of the data...

A popular approach is to "impute" the missing values. For many people, this means replacing the missing values with the most likely value, in some sense: this could be the overall mean or median of the variable, or some forecast of that value from the other, non-missing variables.

To see the problem with this "naive imputation" (imputation with the most likely value), imagine you are trying to compute the volatility of a time series of returns, half of which are missing. Replacing the missing values with the most likely value, i.e., with a constant, would significantly lower the volatility. Even when the problem is not that obvious, replacing missing values with a constant value usually leads to biased estimates.

Instead of imputing the missing values with a single value, we can replace the missing values with a distribution – in practice, a very small sample from that distribution (1 to 5 observations) is sufficient. This is justified if the data is MCAR or MAR (in the latter case, that distribution should be conditional on the values of the other variables).

Since we can rarely assume that the data is missing at random, we can also add a binary variable indicating whether the value was missing. This ensures that we do not discard any piece of information.

With the dataset used in this report, we cannot discard observations with missing values: some of the variables are not available before 2006. For the sake of simplicity, we will use naive imputation with the median of each variable, computed separately for each date.

Data preprocessing: predictors – should we transform the data?

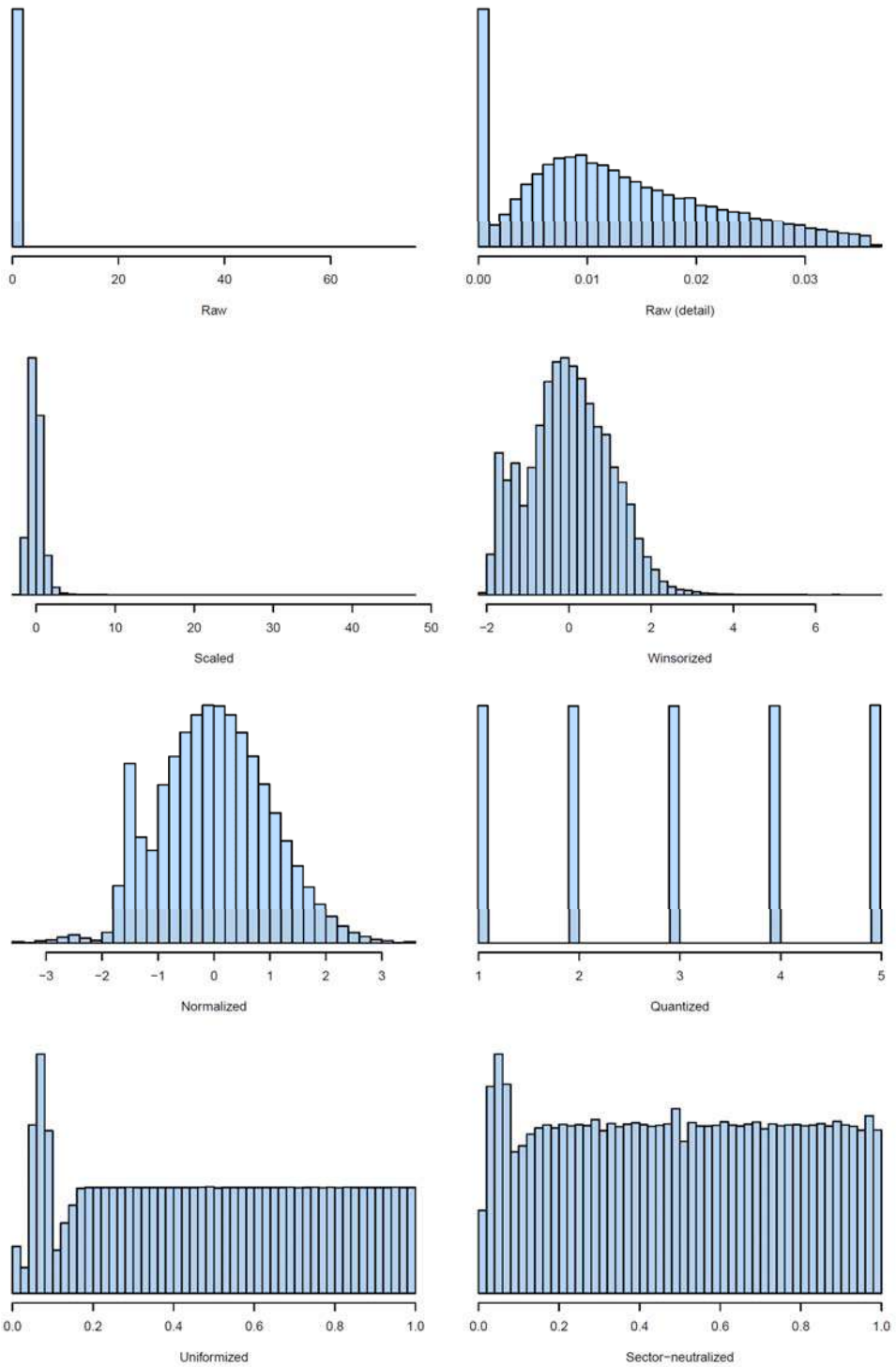
Figure 42 shows possible transformations of the data, taking one of the variables, a dividend yield, as example. We can see that the raw data presents outliers, is skewed, and zero-inflated. Since we transform the data independently at each date, the peak at zero becomes a bit blurry. If we were willing to examine the variables one by one, we could manually identify those that are zero-inflated and transform them accordingly, but, given the large number of variables, we prefer to treat them all in the same way.

Figure 43 shows the effect of transforming the predictors on the performance of the quintile portfolios built using a penalized linear model (elastic net with 10 variables).

- Not pre-processing the data, i.e., mixing variables on different scales, leads to sub-optimal performance.
- Simply rescaling the predictors, with an affine transformation to ensure they all have zero mean and unit variance, is also suboptimal: even if the first two moments match, the variables still have very different distributions – some are skewed, some are zero-inflated, many have fat tails, most have outliers.
- Forcing the data to have a Gaussian distribution or quantizing it helps, but a uniform distribution seems to perform best.

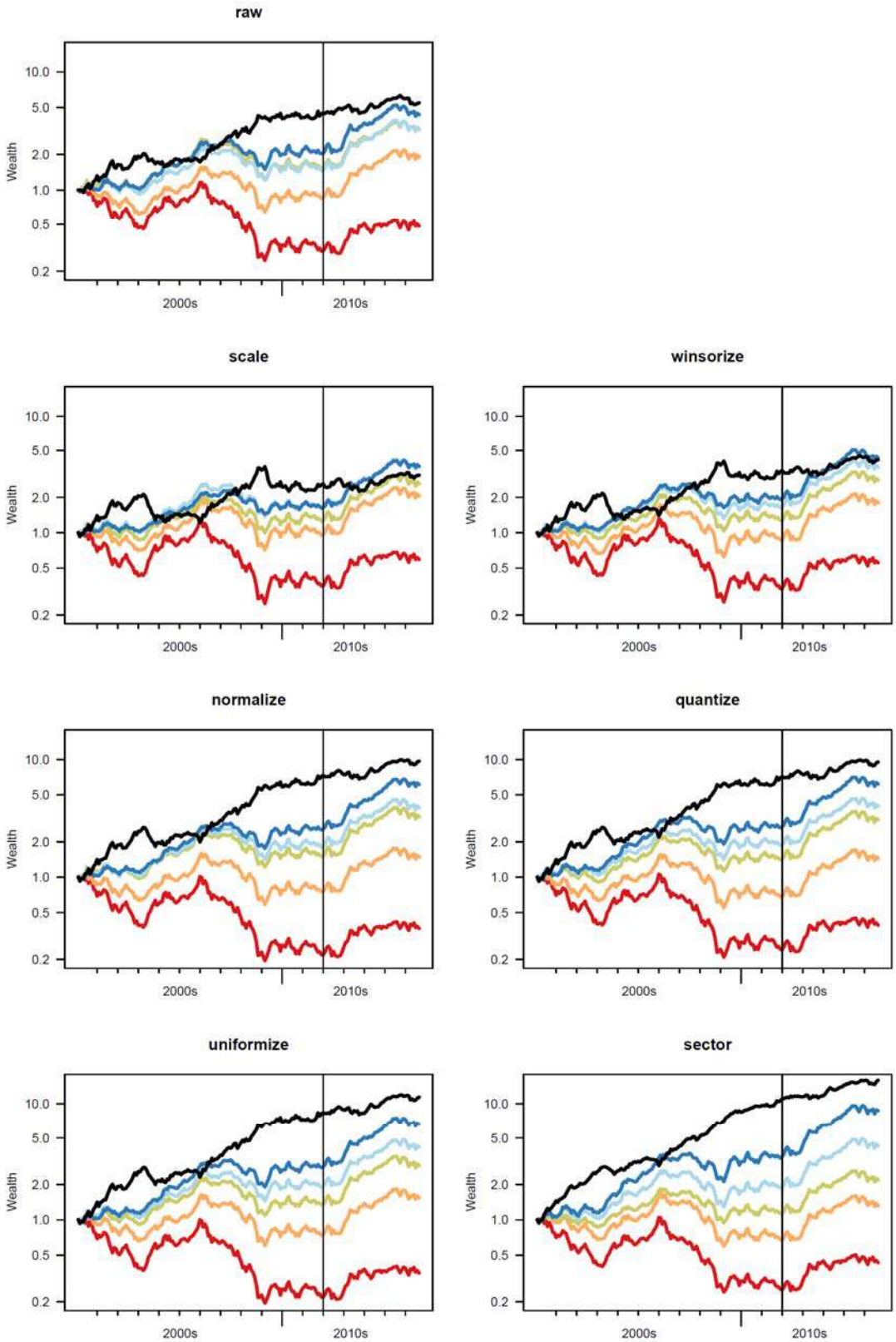
One reason **uniformizing the predictors** works best is that, by cutting the tails of the distribution, it prevents any observation from having an undue leverage on the resulting model.

Figure 42: Transformations one can apply to the predictors



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 43: Effect of pre-processing the predictors



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

More predictor transformations

Many more transformations are possible. Here are a few.

- One can make the predictors sector-neutral, by uniformizing them separately within each sector – Figure 43 suggests this significantly lowers the risk of the resulting strategy;
- One can try to neutralize some undesirable risk factor f, by modeling a predictor x as a function of the risk factor, e.g., $x = \alpha + \beta f + \varepsilon$, and replacing the predictor x with the corresponding residual ε.
- Instead of (or in addition to) the residuals of those regressions, one could also use their coefficients, β.
- Instead of normalizing the data cross-sectionally, one can normalize them in a time series fashion, e.g., by replacing x_{it} with $(x_{it} - \mu_{it})/\sigma_{it}$, where μ_{it} is the average of x_{is}, $s \le t$, or with $\Phi^{-1}(x_{it})$, where Φ^{-1} is an estimator of the cumulated distribution function of x_{is}, $s \le t$.

We will not investigate those transformations further in this report: see [3] for more details.

Data preprocessing: variable to predict – what should we predict?

Figure 44 and Figure 45 show the effect of transforming the variable to predict, i.e., the 1-month ahead returns, (the model is still a penalized regression, selecting at most 10 variables).

Most of the transformations perform slightly better than the raw returns, with no striking differences between them.

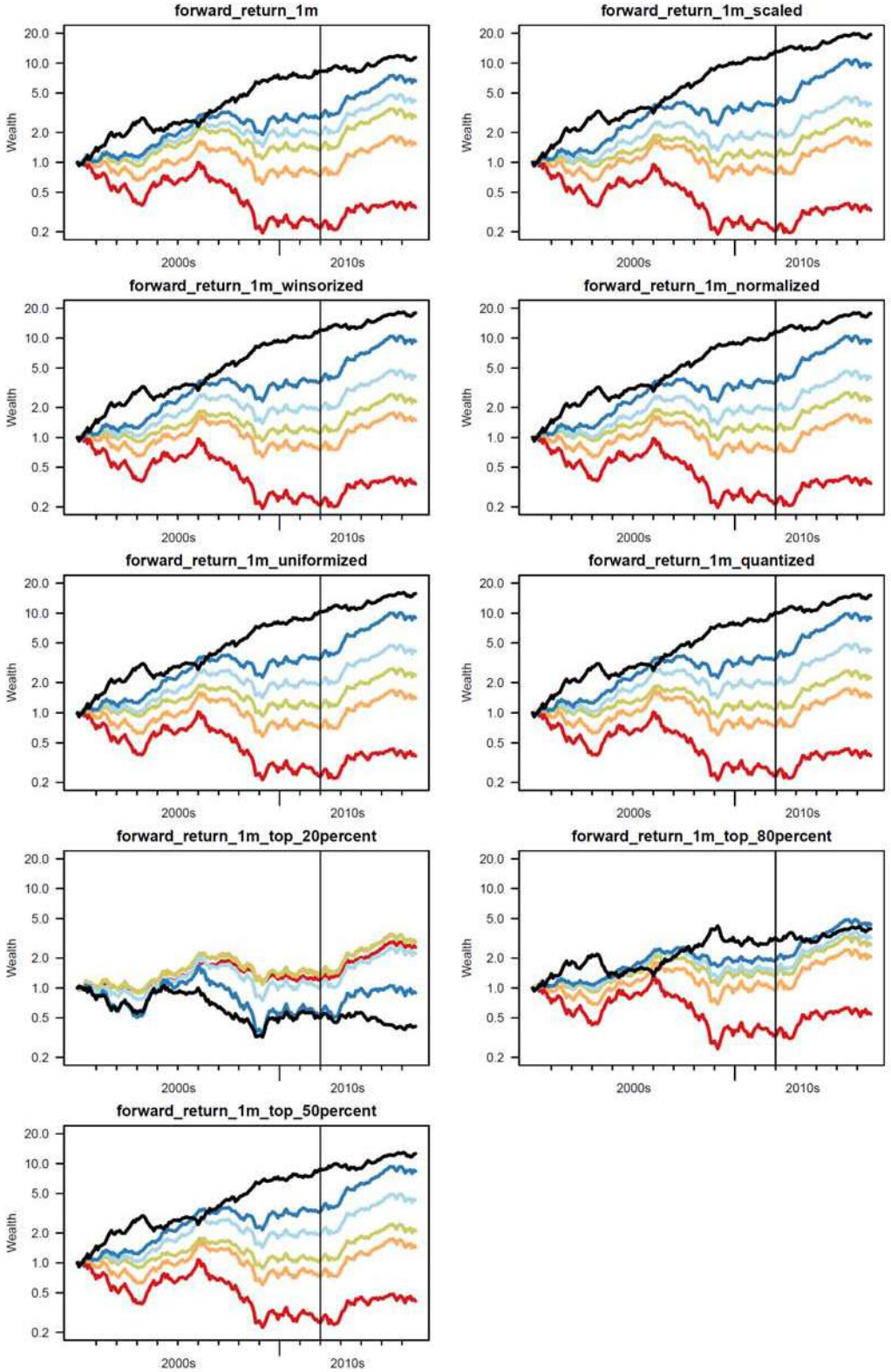
The only exceptions are the binary variables,[28] attempting to predict out-performers (top 20%) or avoid under-performers (i.e., select stock in the top 80%), for which the performance is significantly worse.

While replacing continuous variables with binary variables discards potentially relevant information, that is not the main cause of that difference – indeed, predicting whether a stock will be in the top or bottom 50% does not lead to a big drop in performance. Here are two possible explanations.

- Focusing on the top or bottom 20% creates **unbalanced classes**, which can make classification problems harder [8].
- To predict whether a stock will be in the top quintile, most algorithms will start to notice that stocks with high volatility are likely to remain in the top or bottom quintile and try to slightly improve, if possible, this risky strategy. This is a problem we have mentioned earlier: we should try to optimize something that makes sense from a business point of view. What makes sense, in portfolio construction, is an investment strategy with good returns and low risk: we should be maximizing the strategy returns and minimizing its risk. Predicting whether a stock will be in the top (or bottom) quintile is too far away from that goal.

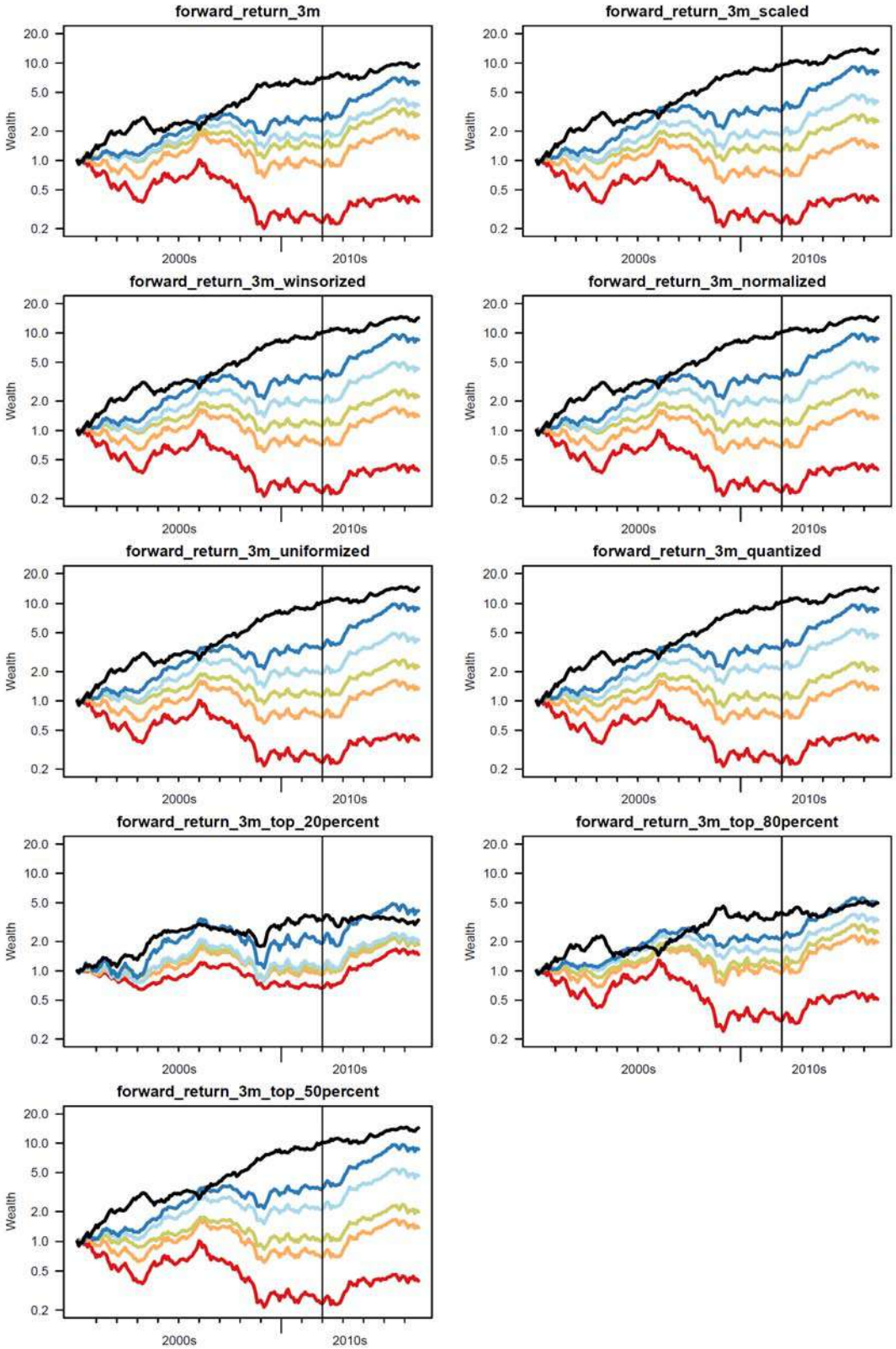
[28] We used a penalized logistic regression for binary variables.

Figure 44: Effect of pre-processing the variable to predict: transforming the variable to predict seems beneficial, except for discretization; in particular, creating unbalanced classes leads to bad performance.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 45: Effect of pre-processing the variable to predict: the results are similar if we try to predict 3-month-ahead returns instead of 1-month-ahead returns.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

The next table summarizes the effect of those transformations, on either the predictors or the variable to predict or both. In the rest of this report, we will uniformize everything – but, if you wanted to invest in those strategies, neutralizing the effect of the sector may lead to an even better performance.

Figure 46: Out-of-sample performance of a penalized regression

input	target	CAGR	AnnVol	IR	tstat	Skew	Kurtosis	HitRatio	MaxDD	VaR95	ES95
raw	raw	0.05	0.11	0.42	1.0	−1.13	1.3	0.65	−0.16	−0.05	−0.08
scale	raw	0.05	0.13	0.39	1.0	−0.57	0.3	0.62	−0.22	−0.07	−0.08
winsorize	raw	0.06	0.11	0.50	1.2	−0.50	−0.1	0.58	−0.15	−0.05	−0.07
normalize	raw	0.07	0.11	0.62	1.4	−0.68	0.5	0.62	−0.15	−0.05	−0.08
quantize	raw	0.07	0.11	0.63	1.4	−0.51	0.2	0.64	−0.15	−0.04	−0.07
uniformize	raw	0.07	0.11	0.68	1.5	−0.71	0.5	0.67	−0.14	−0.04	−0.08
sector	raw	0.08	0.08	1.02	2.2	−0.44	0.5	0.62	−0.08	−0.03	−0.05
raw	uniformize	0.07	0.12	0.56	1.3	−0.36	0.3	0.58	−0.18	−0.05	−0.07
scale	uniformize	0.08	0.13	0.61	1.4	−0.45	0.0	0.65	−0.19	−0.07	−0.08
winsorize	uniformize	0.08	0.11	0.70	1.6	−0.39	0.7	0.60	−0.13	−0.04	−0.07
normalize	uniformize	0.08	0.10	0.79	1.7	−0.53	0.4	0.67	−0.14	−0.04	−0.06
quantize	uniformize	0.09	0.10	0.90	2.0	−0.50	0.4	0.65	−0.11	−0.03	−0.06
uniformize	uniformize	0.10	0.10	0.97	2.1	−0.48	0.2	0.64	−0.11	−0.04	−0.06
sector	uniformize	0.10	0.08	1.17	2.5	−0.48	0.5	0.64	−0.09	−0.03	−0.05
uniformize	raw	0.07	0.11	0.68	1.5	−0.71	0.5	0.67	−0.14	−0.04	−0.08
uniformize	scale	0.09	0.09	1.00	2.2	−0.36	0.0	0.65	−0.09	−0.03	−0.05
uniformize	winsorize	0.09	0.09	0.97	2.1	−0.42	0.2	0.67	−0.10	−0.03	−0.05
uniformize	normalize	0.10	0.10	1.01	2.2	−0.45	0.2	0.65	−0.10	−0.03	−0.06
uniformize	quantize	0.09	0.10	0.91	2.0	−0.47	0.2	0.65	−0.12	−0.04	−0.06
uniformize	uniformize	0.10	0.10	0.97	2.1	−0.48	0.2	0.64	−0.11	−0.04	−0.06
uniformize	top 50%	0.09	0.10	0.82	1.8	−0.49	0.4	0.65	−0.13	−0.04	−0.06
uniformize	top 20%	−0.05	0.14	−0.39	−0.7	0.40	−0.1	0.36	−0.32	−0.07	−0.08
uniformize	top 80%	0.06	0.13	0.43	1.0	−0.44	0.1	0.58	−0.21	−0.07	−0.08

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Ties

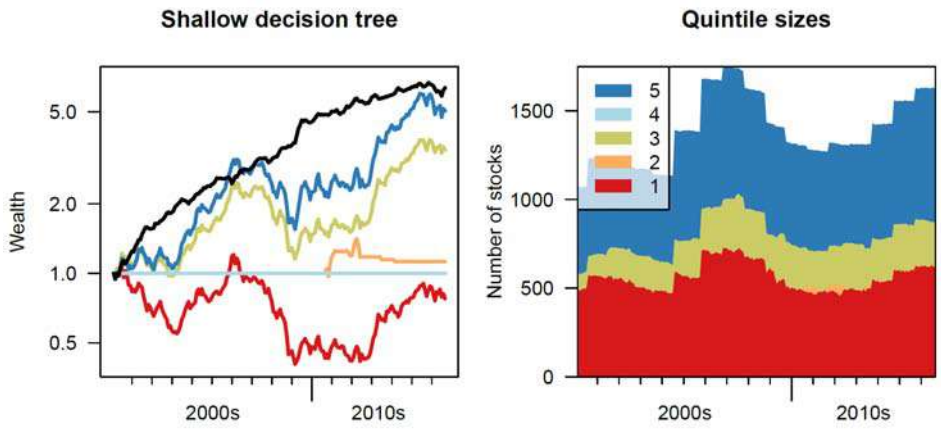
Ties can appear in the input or the output of a model and, in both cases, they pose problems. We have already seen that ties, and in particular unbalanced discrete variables, pose problems in the variable to predict.

Figure 48 shows that some of our predictors are discrete: for instance, the Piotroski signal is a sum of binary variables and can only take 9 possible values. Attempting to build quintile portfolios from those 9 values (top plots) gives unbalanced and possibly empty portfolios. Using those 9 values to build 9 portfolios still gives unbalanced portfolios: the extreme values are quite rare. Those discrete values also pose problems when fed to machine learning algorithms: most algorithms prefer continuous variables.

Ties can appear in the output, as well: some machine learning algorithms, in particular those based on trees, can produce a lot of ties, which can pose problems when building quintile portfolios, as shown on Figure 47.

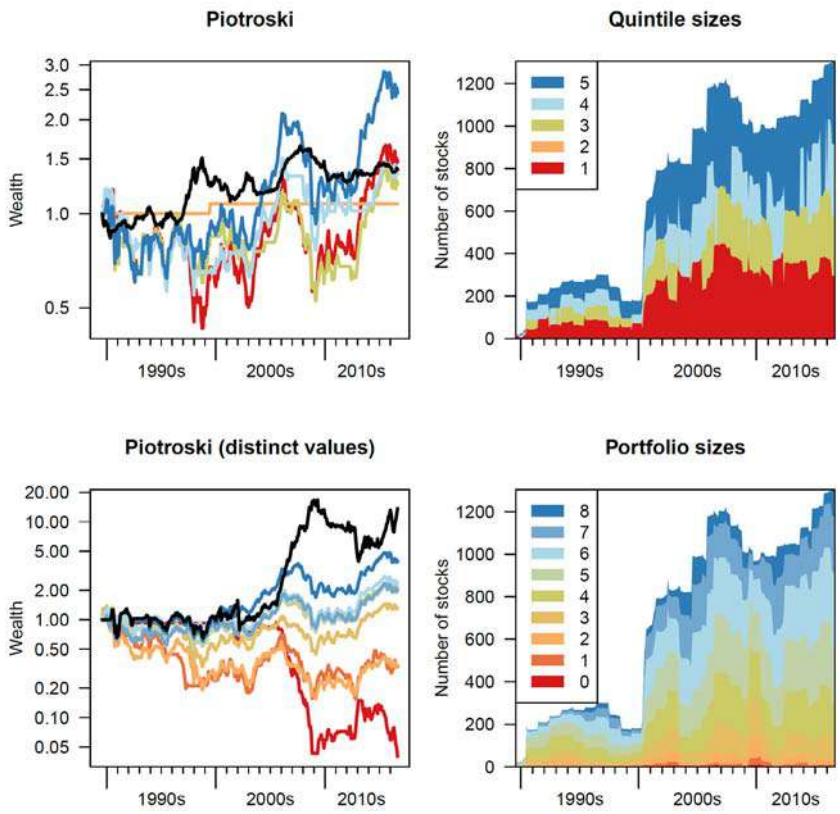
To avoid those problems, the best is to check where those ties come from – often, they are caused by excessive rounding, and can easily be avoided. If this is not possible, one can add a small amount of noise to discrete signals, just enough to avoid duplicated values – this does not fix the problem, but may be sufficient to hide it.

Figure 47: Shallow or unbalanced trees generate a lot of ties, and unbalanced or empty quintile portfolios.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 48: Some of our investment signals are discrete



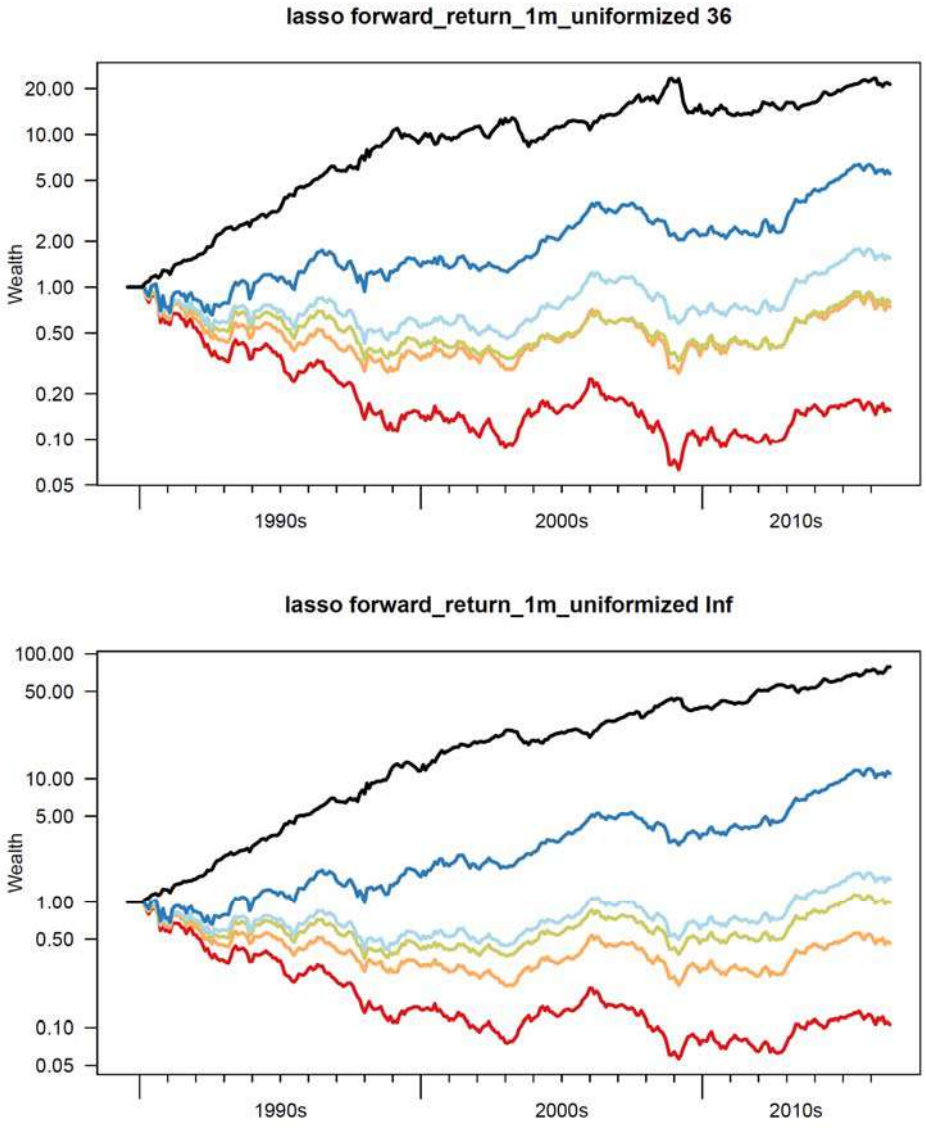
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Estimation period

In finance, one often chooses an "in-sample" period to estimate the model and an "out-of-sample" period to assess its performance. This clearly highlights if the model overfits the data (figure 18).

Alternatively, one can estimate the model on a moving window, of various sizes, or even use an expanding window. Figure 49 shows the performance if the model (here, a penalized regression, selecting at most 10 variables) is estimated on a 3-year moving window, or on an expanding window;[29] Figure 50 shows the risk-return profile as we change the window size: larger windows lead to higher returns and lower risk.

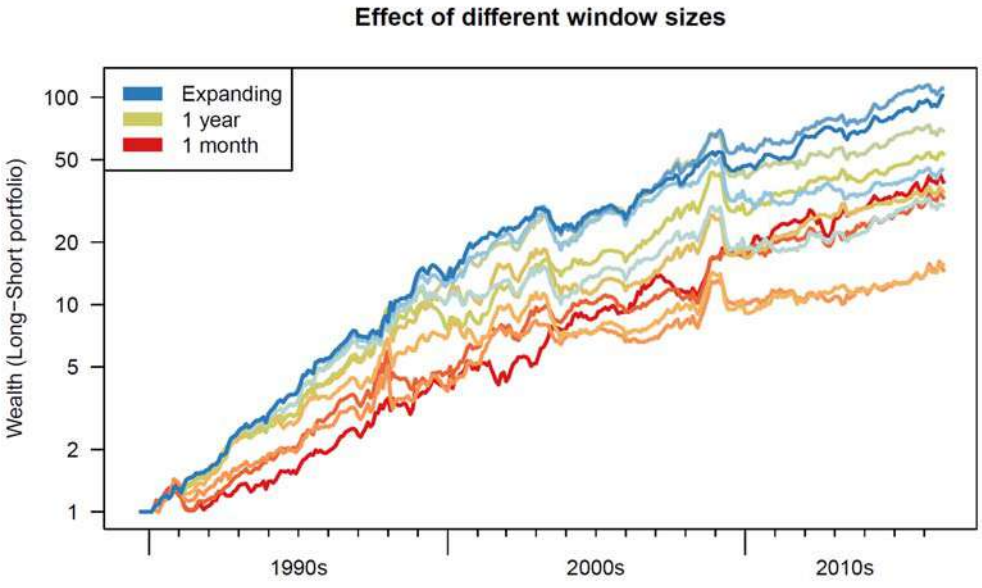
Figure 49: Penalized regression, since 1990, with a 3-year or expanding window.

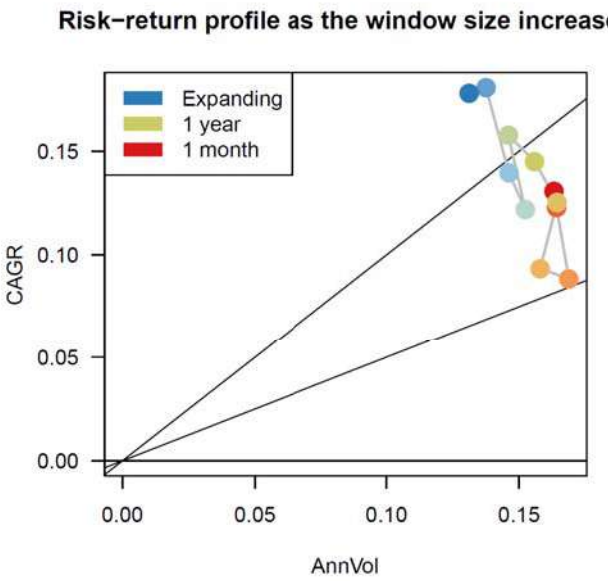


Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

[29] For most models, it is also possible to use exponentially decaying weights.

Figure 50: How the window size affects the performance of the strategy





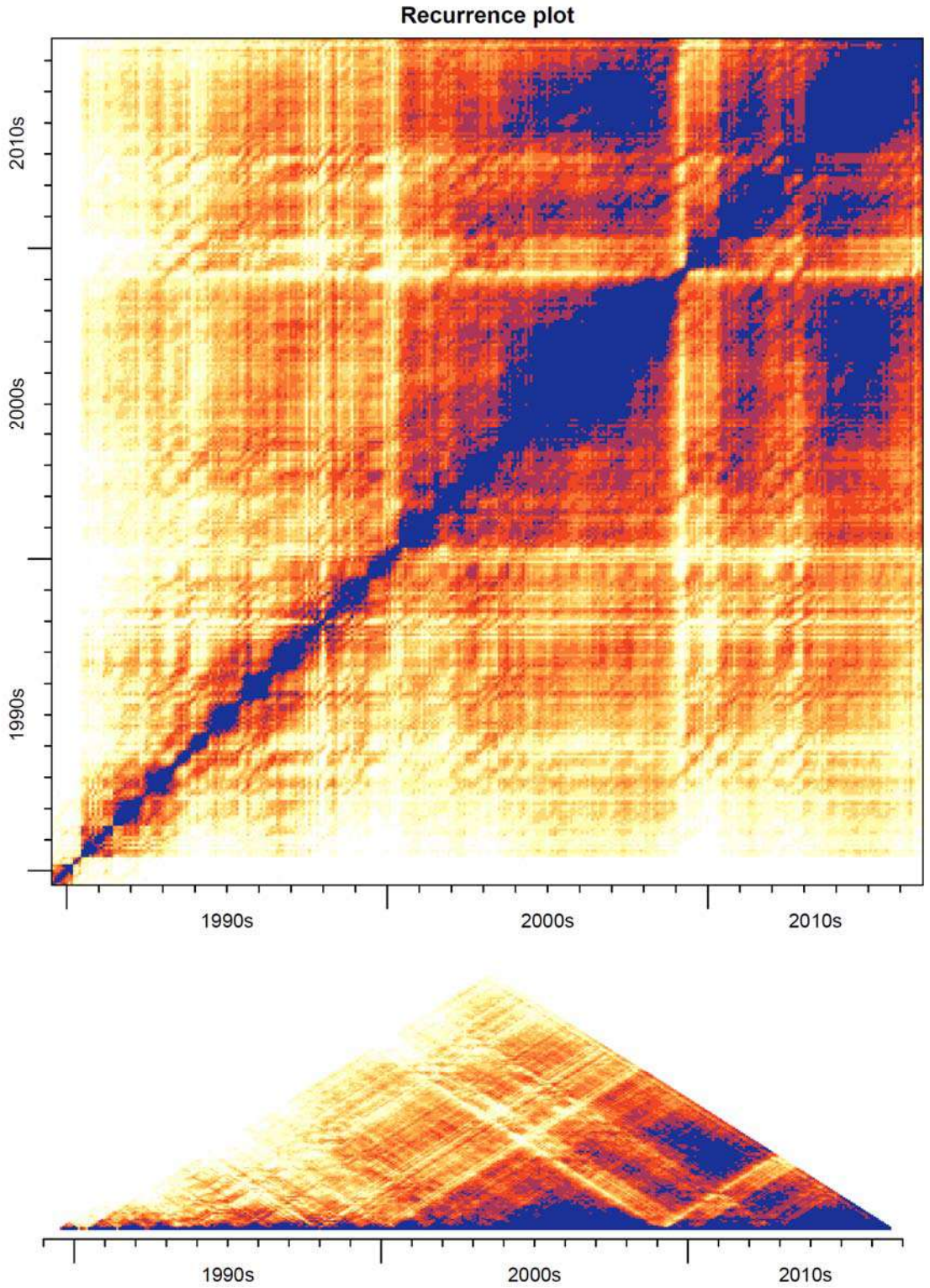
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Regimes

To see if there are different "regimes" in the data, to measure how similar the in- and out-of-sample periods are, and to assess how diverse the in-sample period was, we can look at a recurrence plot [30, 38] of the cross-sectional rank correlation matrices between the factors. Figure 51 plots the distances between those correlation matrices, with black corresponding to short distances, i.e., similar periods. We can see that:

- The period from 2000 to 2009 was relatively uniform, and similar to 2011–present;
- The two years 2009–2010 stand out;
- The previous period, 1990–2000 was significantly different and heterogeneous.
- We can also notice annual patterns, especially in the 1990s, reflecting the fact that many of the variables only change once a year.

Figure 51: Recurrence plot. Dark areas correspond to similar periods. The second plot is the upper left half of the first one, rotated.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

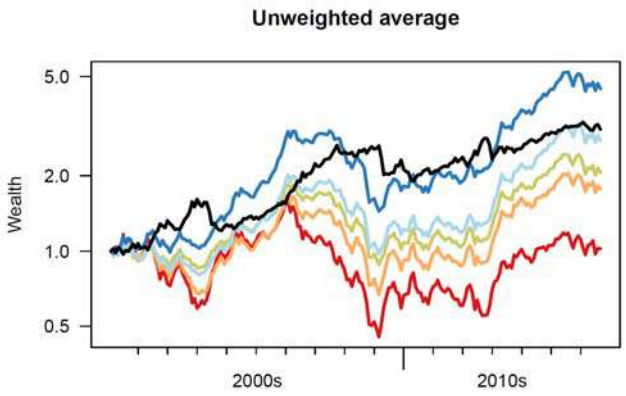
Looking for the best model

Baseline model

Before trying all the models we can think of and the latest machine learning fads, we need some baseline model, to make sure that those fancy algorithms indeed add something to simplistic approaches. We will use an unweighted average of the uniformized investment signals (taking into account the direction in which they are supposed to be used: for instance, measures of risk enter this average with a negative sign). Figure 52 shows the performance of this model.

This is just a baseline model: the performance is supposed to be neither great nor consistent – since the factors came from the literature and are known to have some predictive power on future returns, we expect it to be positive, on average, but not much more. Indeed, the information ratio is a mediocre 0.5 and the strategy presents worrying drawdowns.

Figure 52: Baseline model: unweighted average



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

False start

Since there are dozens or hundreds of machine learning models [14, 43], we were first tempted to try them all. We have selected the models from the caret R package [23], but limited ourselves to models that could be fitted in a reasonable amount of time. To limit computation time further, we have used the default parameters for all those models.

We ended up with the following models (there are many duplicates, i.e., different implementations of the same algorithm, usually with different default hyperparameters):

- **Linear models**: lm, glm, bayesglm, nnls (non-negative least squares);
- **Penalized linear models** (lasso, elastic net): enet, glmnet, lars, lars2;
- Linear models with **variable selection**: leapBackward, leapForward, leapSeq;
- **Principal component regression**, partial least squares: pcr, simpls, kernelpls;
- **Decision trees**: ctree, ctree2, rpart, rpart1SE, rpart2;
- **Regression trees**: earth, gcvEarth;
- **Boosting**: blackboost, BstLm, bstTree, gamboost, gbm, glmboost, LogitBoost;
- **Neural networks**: nnet, dnn;
- **Self-orgamizing maps** (SOM): bdk (bidirectional Kohonen maps map x and y to the same space, alternating x and y during training), xyf (X-Y-fused SOM concatenates x and y);
- **Linear discriminant analysis**: lda, Mlda, sda (shrinkage LDA, high-dimensional LDA), sparseLDA.

With the exception of Kohonen maps, these are all supervised learning algorithms, used to predict forward returns (or, for the classification algorithms, whether the forward returns will be above the median).

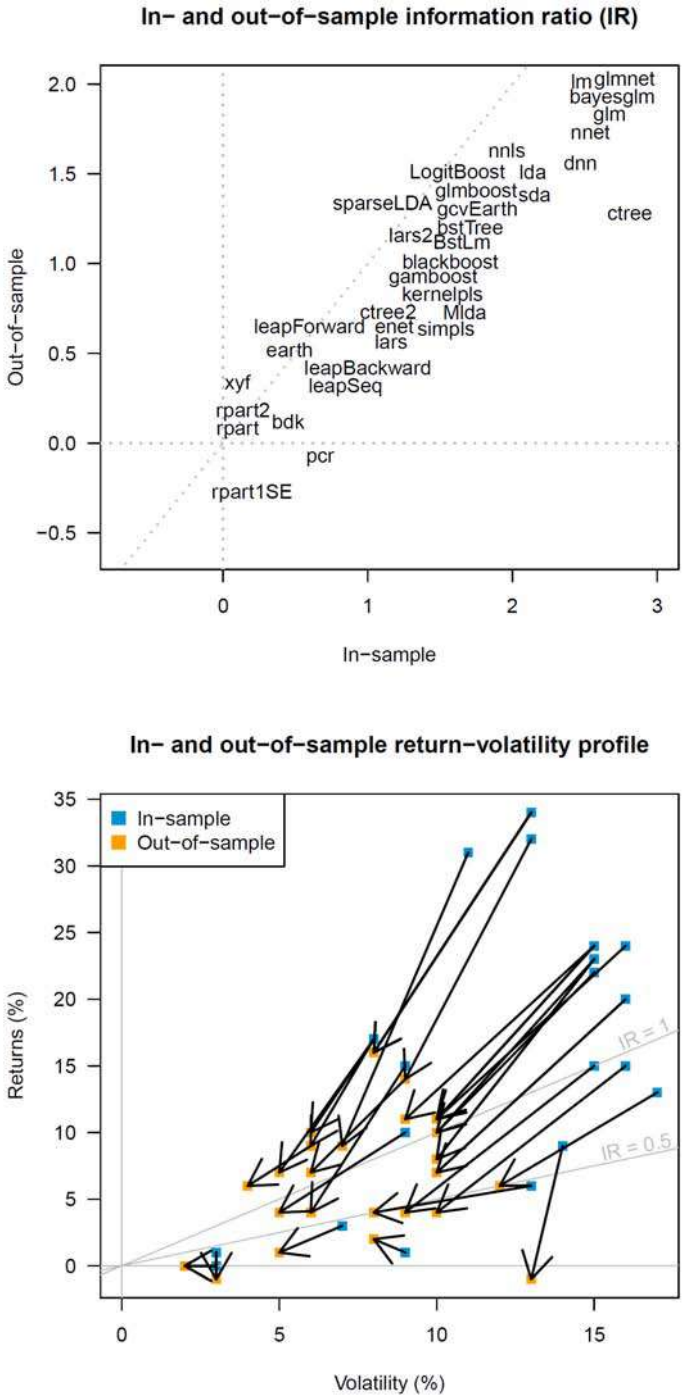
Figure 53 shows that to a similar but non-worrying extent, all those algorithms overfit the data (the in-sample performance is better than the out-of-sample one) and that the out-of-sample volatility is much lower than the in-sample one (this may be due to the fact that the in-sample period contains the global financial crisis, though, as suggested by Figure 54).

Figure 55 and Figure 56 suggest that, without any hyperparameter tuning, the best models are the linear ones – in particular, ordinary least squares, despite overfitting the data, performs well. This is doubly worrying: we do not want to overfit the data, and we want to improve on linear models.

Figure 57 and Figure 58 show the performance in the volatility×return space. We can see that the quintile portfolios are ordered as expected, that the long-short portfolios have a similar information ratio but a lower volatility, and that linear models, neural nets, penalized models and boosting perform best. Figure 59 and Figure 60 provide the actual numbers.

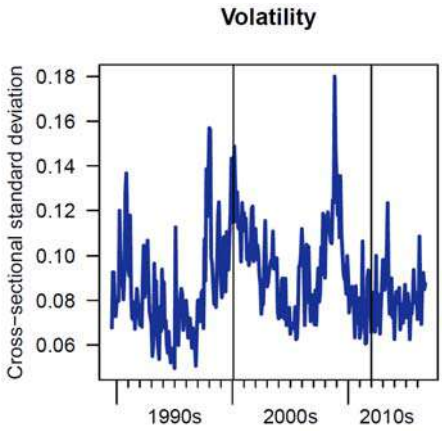
In the rest of this report, we will focus on the hyperparameters of a handful of those models: penalized regression, decision trees, boosted trees and boosted splines.

Figure 53: In- and out-of-sample performance: the in-sample performance is better (top), but the volatility is lower out-of-sample (bottom)



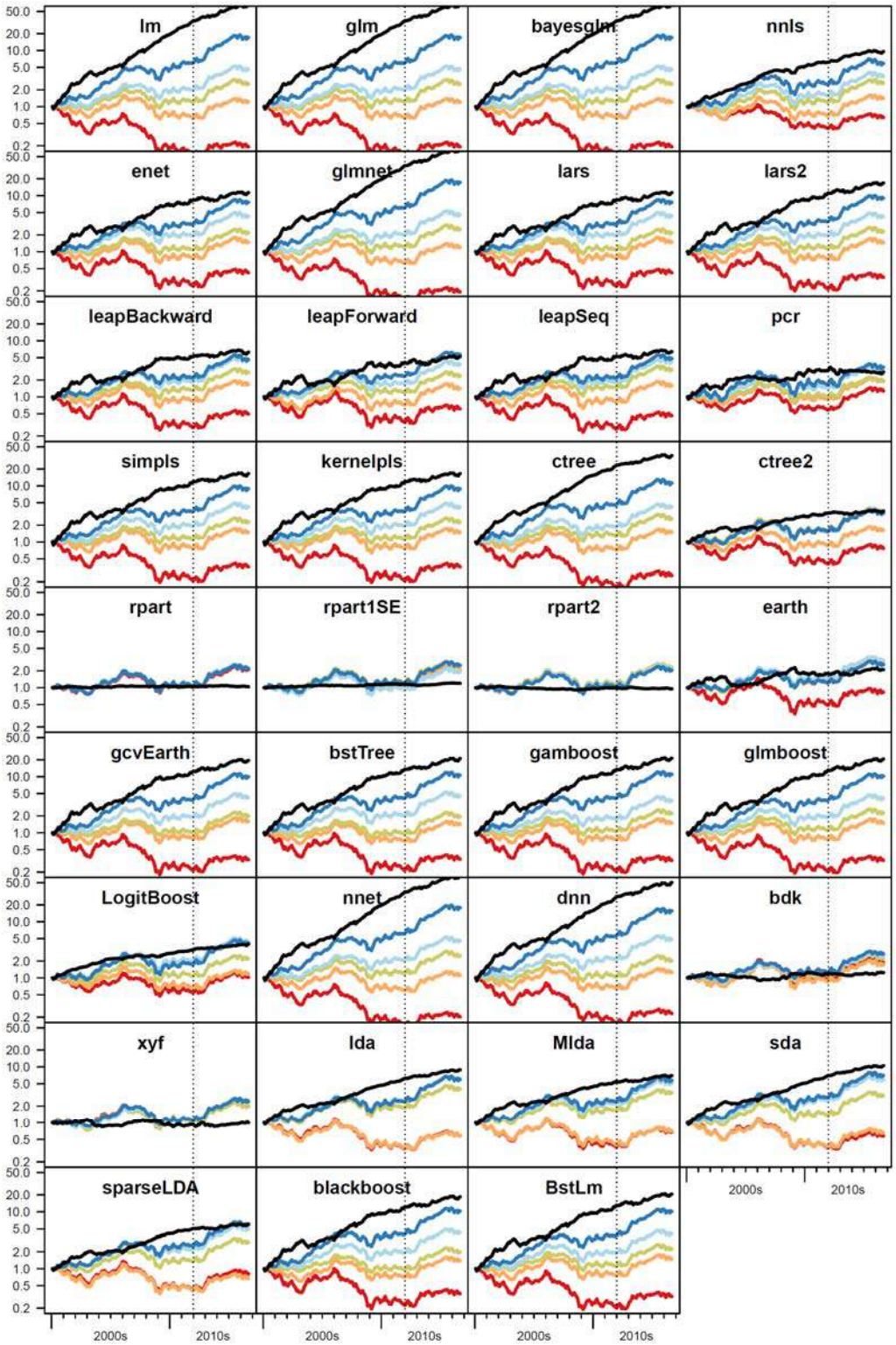
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 54: The in-sample period was more volatile



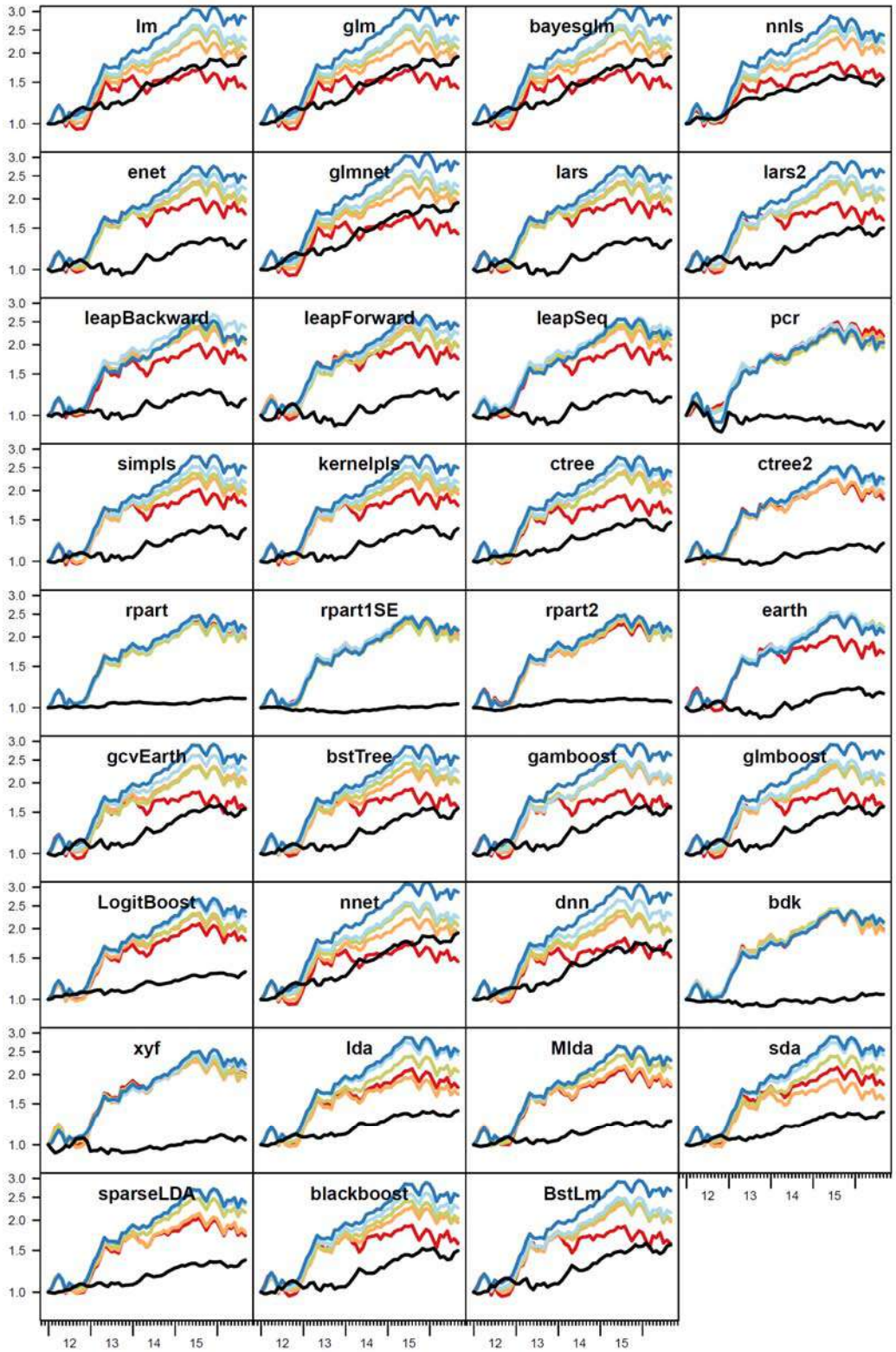
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 55: Performance of a large number of algorithms



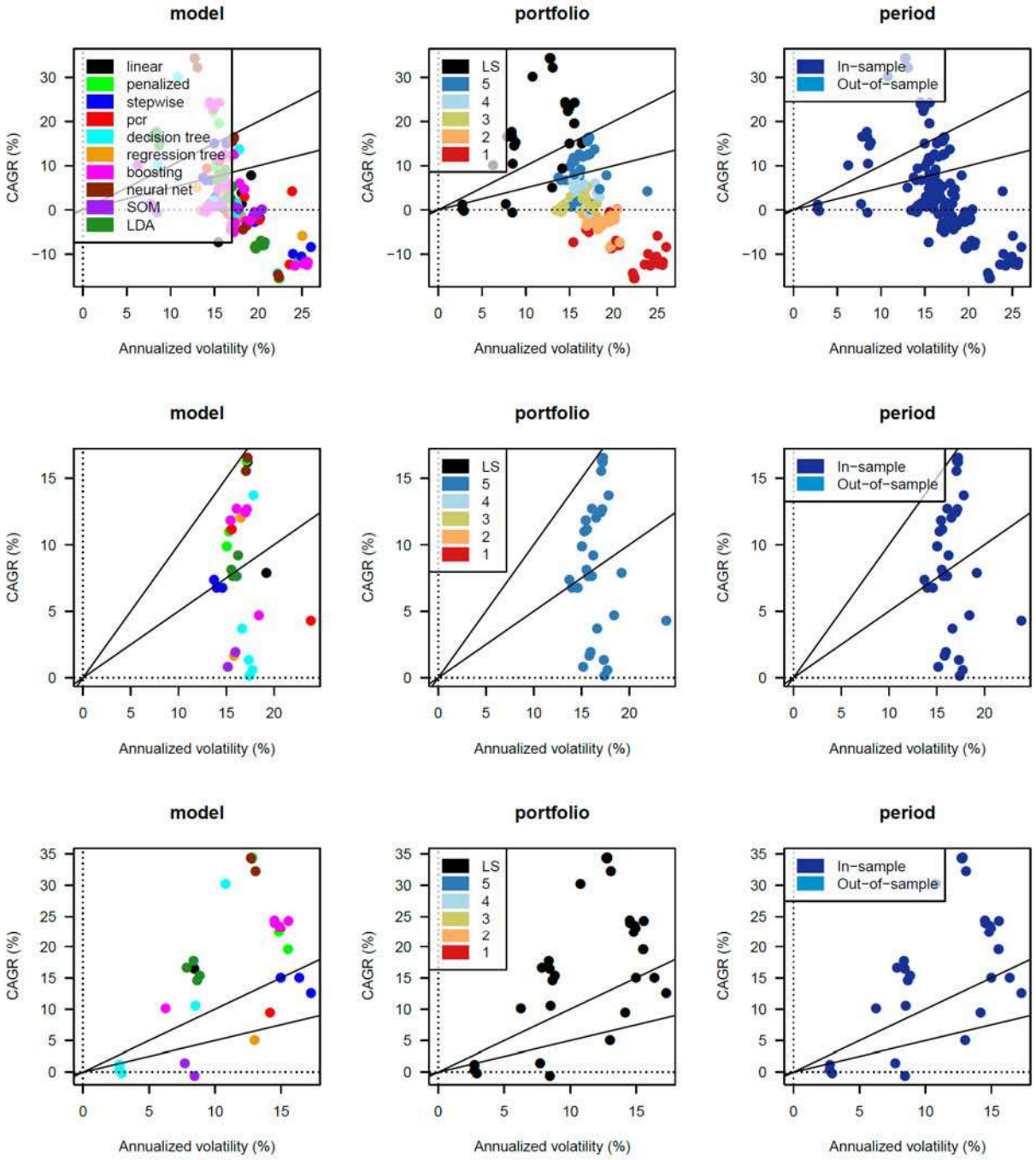
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 56: Out-of-sample performance of a large number of algorithms



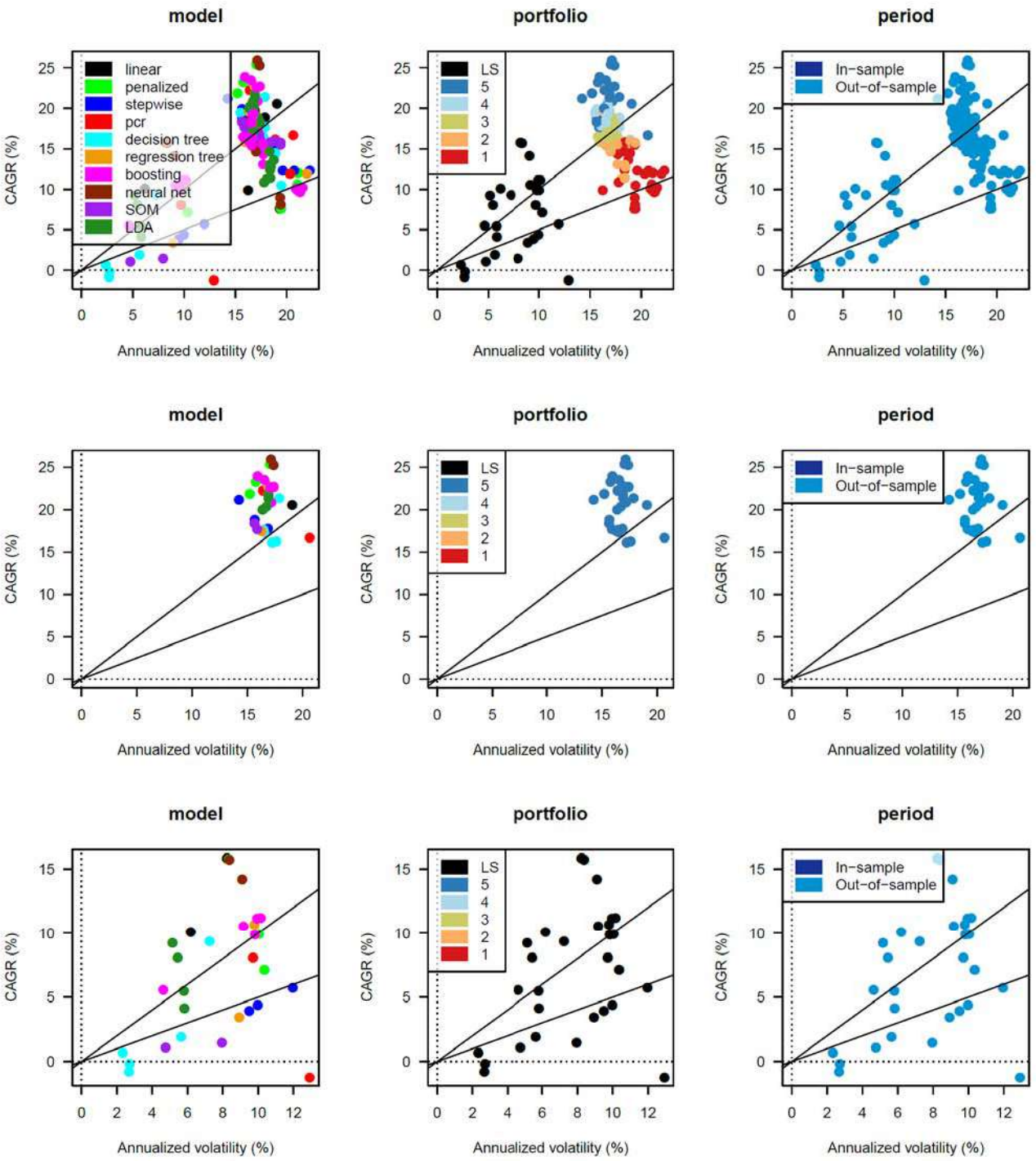
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 57: In-sample performance of a larger number of algorithms. All the plots are identical, but they show or highlight different aspects of the data. Each dot corresponds to a portfolio. The first row shows all the portfolios; the second the fifth quintile portfolios, the third the long-short portfolios.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 58: Out-of-sample performance of a larger number of algorithms



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 59: Performance (out-of-sample, long-short portfolio, sorted by decreasing information ratio)

Model	Portfolio	CAGR	AnnVol	IR	tstat	Skew	Kurtosis	HitRatio	MaxDD	VaR95	ES95
bayesglm	LS	0.16	0.08	1.92	4.0	−0.29	0.0	0.75	−0.07	−0.03	−0.04
glm	LS	0.16	0.08	1.92	4.0	−0.29	0.0	0.75	−0.07	−0.03	−0.04
lm	LS	0.16	0.08	1.92	4.0	−0.29	0.0	0.75	−0.07	−0.03	−0.04
glmnet	LS	0.16	0.08	1.89	3.9	−0.32	0.1	0.74	−0.07	−0.03	−0.04
nnet	LS	0.16	0.08	1.88	3.9	−0.19	0.0	0.74	−0.07	−0.03	−0.04
sda	LS	0.09	0.05	1.66	3.5	−0.27	−0.2	0.75	−0.04	−0.02	−0.03
nnls	LS	0.10	0.06	1.63	3.4	−0.02	−0.6	0.63	−0.10	−0.02	−0.03
dnn	LS	0.14	0.09	1.56	3.3	0.02	0.5	0.70	−0.08	−0.04	−0.05
lda	LS	0.08	0.05	1.54	3.3	−0.17	−0.6	0.72	−0.04	−0.02	−0.02
LogitBoost	LS	0.06	0.04	1.30	2.8	−0.05	−0.8	0.68	−0.05	−0.01	−0.02
ctree	LS	0.09	0.07	1.29	2.8	−0.24	−0.2	0.68	−0.10	−0.03	−0.04
bstTree	LS	0.11	0.09	1.15	2.5	−0.18	−0.3	0.63	−0.10	−0.04	−0.05
BstLm	LS	0.11	0.10	1.12	2.4	−0.42	0.0	0.67	−0.11	−0.03	−0.05
glmboost	LS	0.11	0.10	1.12	2.4	−0.42	0.0	0.67	−0.11	−0.03	−0.05
gamboost	LS	0.11	0.10	1.10	2.4	−0.35	−0.5	0.63	−0.10	−0.04	−0.05
gcvEarth	LS	0.11	0.10	1.09	2.4	−0.44	−0.5	0.67	−0.11	−0.04	−0.05
blackboost	LS	0.10	0.10	1.01	2.2	−0.20	−0.5	0.63	−0.10	−0.04	−0.05
Mlda	LS	0.06	0.06	1.00	2.2	−0.29	−0.9	0.60	−0.07	−0.03	−0.03
lars2	LS	0.10	0.10	0.99	2.2	−0.42	0.0	0.63	−0.11	−0.03	−0.06
sparseLDA	LS	0.06	0.06	0.93	2.0	−0.31	−0.3	0.61	−0.06	−0.03	−0.03
kernelpls	LS	0.08	0.10	0.83	1.8	−0.16	0.2	0.60	−0.12	−0.04	−0.05
simpls	LS	0.08	0.10	0.83	1.8	−0.16	0.2	0.60	−0.12	−0.04	−0.05
enet	LS	0.07	0.10	0.69	1.6	−0.50	0.5	0.61	−0.14	−0.04	−0.07
lars	LS	0.07	0.10	0.69	1.6	−0.50	0.5	0.61	−0.14	−0.04	−0.07
ctree2	LS	0.03	0.06	0.56	1.3	−0.19	−0.5	0.56	−0.09	−0.02	−0.03
rpart1SE	LS	0.01	0.02	0.54	1.2	0.29	−0.3	0.58	−0.03	−0.01	−0.01
leapForward	LS	0.06	0.12	0.48	1.1	−0.48	−0.1	0.63	−0.20	−0.07	−0.07
leapSeq	LS	0.04	0.10	0.44	1.0	−0.25	0.1	0.53	−0.14	−0.04	−0.06
leapBackward	LS	0.04	0.09	0.41	1.0	−0.26	0.1	0.53	−0.16	−0.05	−0.06
rpart2	LS	0.01	0.02	0.36	0.8	−0.74	1.4	0.61	−0.04	−0.01	−0.02
xyf	LS	0.03	0.08	0.35	0.8	−0.54	1.4	0.58	−0.14	−0.03	−0.05
earth	LS	0.03	0.08	0.34	0.8	−0.53	−0.2	0.63	−0.17	−0.04	−0.05
rpart	LS	0.01	0.02	0.26	0.6	−0.01	0.4	0.54	−0.03	−0.01	−0.01
bdk	LS	0.00	0.05	−0.01	0.0	0.48	0.3	0.53	−0.09	−0.02	−0.02
pcr	LS	−0.01	0.13	−0.09	−0.1	0.77	1.9	0.44	−0.25	−0.06	−0.07

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 60: Performance (out-of-sample, long portfolio, sorted by decreasing information ratio)

Model	Portfolio	CAGR	AnnVol	IR	tstat	Skew	Kurtosis	HitRatio	MaxDD	VaR95	ES95
nnet	5	0.26	0.17	1.51	3.1	−0.30	−0.4	0.68	−0.16	−0.07	−0.09
BstLm	5	0.24	0.16	1.50	3.1	−0.20	−0.4	0.65	−0.14	−0.06	−0.08
glmboost	5	0.24	0.16	1.50	3.1	−0.20	−0.4	0.65	−0.14	−0.06	−0.08
bayesglm	5	0.25	0.17	1.49	3.1	−0.31	−0.3	0.67	−0.16	−0.07	−0.09
glm	5	0.25	0.17	1.49	3.1	−0.31	−0.3	0.67	−0.16	−0.07	−0.09
glmnet	5	0.25	0.17	1.49	3.1	−0.31	−0.3	0.67	−0.16	−0.07	−0.09
leapForward	5	0.21	0.14	1.49	3.1	−0.13	−0.4	0.65	−0.13	−0.05	−0.07
lm	5	0.25	0.17	1.49	3.1	−0.31	−0.3	0.67	−0.16	−0.07	−0.09
lars2	5	0.23	0.16	1.48	3.1	−0.17	−0.4	0.65	−0.14	−0.06	−0.08
dnn	5	0.25	0.17	1.45	3.0	−0.31	−0.3	0.65	−0.17	−0.07	−0.09
enet	5	0.22	0.15	1.44	3.0	−0.18	−0.5	0.63	−0.15	−0.06	−0.07
lars	5	0.22	0.15	1.44	3.0	−0.18	−0.5	0.63	−0.15	−0.06	−0.07
gamboost	5	0.24	0.17	1.42	3.0	−0.24	−0.4	0.67	−0.15	−0.06	−0.08
kernelpls	5	0.22	0.16	1.36	2.8	−0.32	−0.3	0.65	−0.17	−0.06	−0.09
simpls	5	0.22	0.16	1.36	2.8	−0.32	−0.3	0.65	−0.17	−0.06	−0.09
gcvEarth	5	0.23	0.17	1.35	2.8	−0.26	−0.3	0.67	−0.17	−0.07	−0.09
blackboost	5	0.23	0.17	1.33	2.8	−0.30	−0.4	0.63	−0.18	−0.07	−0.09
bstTree	5	0.23	0.17	1.31	2.8	−0.29	−0.4	0.63	−0.18	−0.07	−0.09
lda	5	0.22	0.17	1.30	2.8	−0.35	−0.4	0.63	−0.17	−0.07	−0.09
sda	5	0.22	0.17	1.30	2.7	−0.31	−0.4	0.68	−0.16	−0.07	−0.09
sparseLDA	5	0.20	0.16	1.23	2.6	−0.26	−0.5	0.65	−0.17	−0.07	−0.09
xyf	5	0.20	0.16	1.22	2.6	−0.20	−0.4	0.68	−0.15	−0.06	−0.08
ctree	5	0.21	0.18	1.20	2.6	−0.28	−0.3	0.63	−0.19	−0.07	−0.10
leapSeq	5	0.19	0.16	1.20	2.6	−0.37	−0.4	0.63	−0.17	−0.07	−0.08
Mlda	5	0.20	0.17	1.20	2.6	−0.26	−0.4	0.65	−0.17	−0.07	−0.09
LogitBoost	5	0.20	0.17	1.17	2.5	−0.25	−0.3	0.63	−0.18	−0.07	−0.09
ctree2	5	0.19	0.17	1.11	2.4	−0.34	−0.3	0.65	−0.18	−0.07	−0.09
earth	5	0.18	0.16	1.11	2.4	−0.24	−0.5	0.65	−0.14	−0.07	−0.08
bdk	5	0.18	0.16	1.09	2.4	−0.25	−0.3	0.63	−0.15	−0.06	−0.08
nnls	5	0.21	0.19	1.08	2.3	−0.14	−0.5	0.60	−0.22	−0.07	−0.10
rpart2	5	0.18	0.17	1.06	2.3	−0.29	−0.4	0.65	−0.15	−0.07	−0.09
leapBackward	5	0.18	0.17	1.05	2.3	−0.48	−0.2	0.67	−0.22	−0.08	−0.10
rpart	5	0.17	0.17	1.00	2.2	−0.24	−0.4	0.67	−0.18	−0.07	−0.09
rpart1SE	5	0.17	0.17	0.98	2.2	−0.27	−0.4	0.63	−0.17	−0.07	−0.09
pcr	5	0.17	0.21	0.81	1.9	−0.12	0.1	0.63	−0.25	−0.08	−0.12

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Hyperparameter tuning: grid search

Most of the algorithms we have described in the first part depend on a few parameters: the penalty scale(s) for the lasso, the penalty for misclassified observations for support vector machines, the number and depth of the trees for random forests, the step size and the number of steps for boosting – and some implementations provide even more parameters to fine-tune.

While those "**hyperparameters**" may seem innocuous, some of them turn out to have a huge impact on the performance of the models – and it is rarely obvious which ones do. It is therefore important to fine-tune machine learning algorithms by carefully choosing those parameters.

Let us take the lasso or, more precisely, the elastic net, as an example. There are two parameters:

- $\lambda \geq 0$ is the amplitude of the penalty;
- $\alpha \in [0,1]$ is the balance between the L^2 and L^1 penalties, corresponding to $\alpha = 0$ and $\alpha = 1$ respectively.

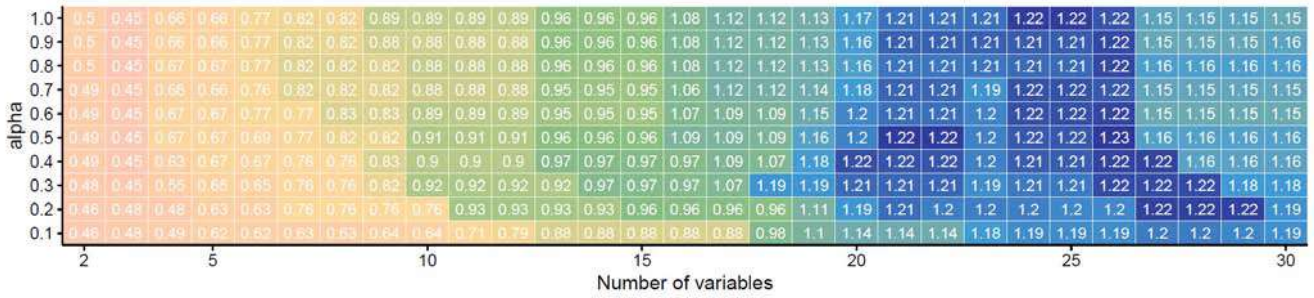
Since λ is not easy to interpret, we can try to use the number n of variables selected by the model instead.

In Figure 61, we use a **grid search**[30] to optimize the out-of-sample information ratio, as α and the number of variables selected varies. The maximum is obtained for $\alpha = 0.5$ and 26 variables. We also notice that α does not seem to play a big role. In this plot, we have limited ourselves to 30 variables, to ensure that the model remains easy to interpret.

Figure 62 shows the whole regularization path: if we regularize even less, i.e., if we allow even more variables, the information ratio jumps from 1.2 to 1.85. From Figure 63, we see that we are basically fitting a linear model with 90% of our variables – that is a lot for a supposedly sparse model...

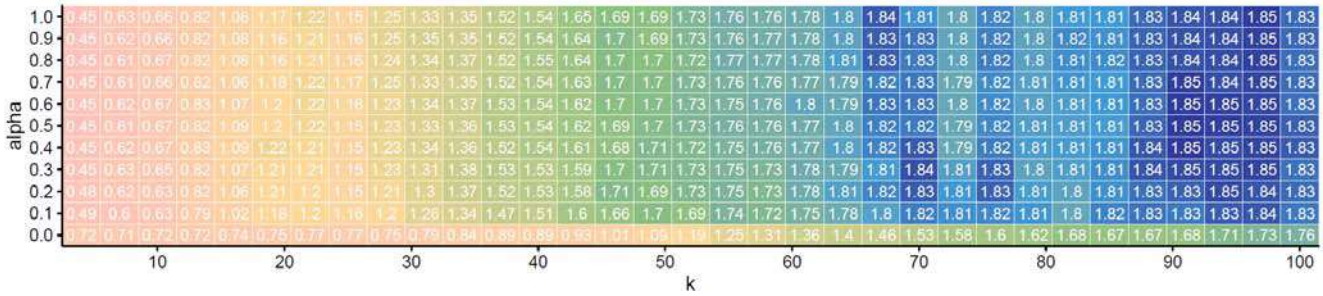
[30] To be rigorous, we should not use two datasets, a training (in-sample) set and a validation (out-of-sample) set, but three: we also need a validation set. The results we present are therefore not truly "out-of-sample". See [43] for more details.

Figure 61: Out-of-sample information ratio (IR) for the elastic net models, as a function of α and the number of variables selected. Note that $\alpha = 0$ is missing from the plot: it corresponds to the ridge regression, which has no sparsifying effect.



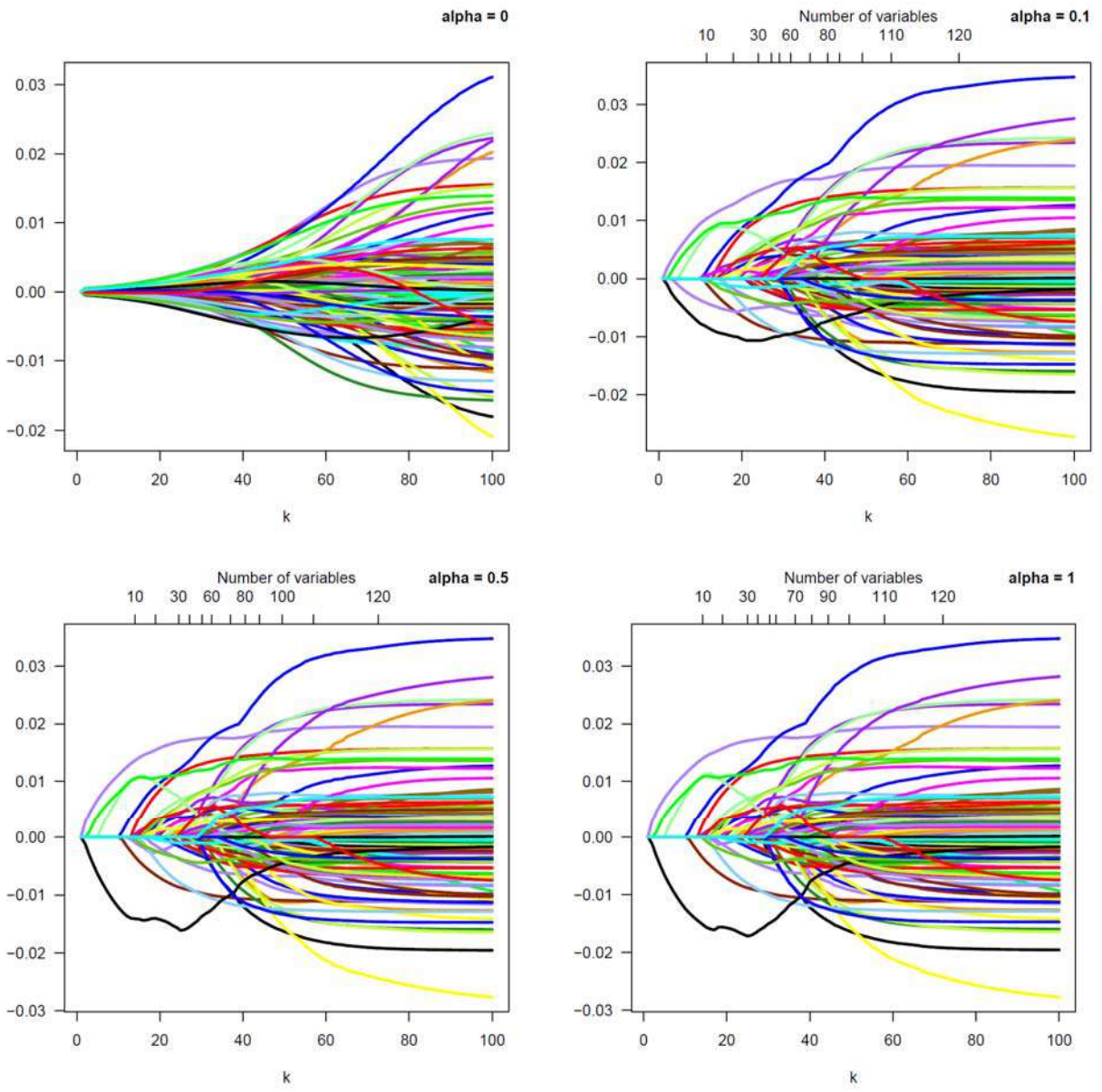
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 62: Out-of-sample information ratio (IR) for the elastic net models for the whole regularization path. Here, k = $-\log \lambda$, up to an additive constant; λ varies approximately from 1 to 10^{-6}.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

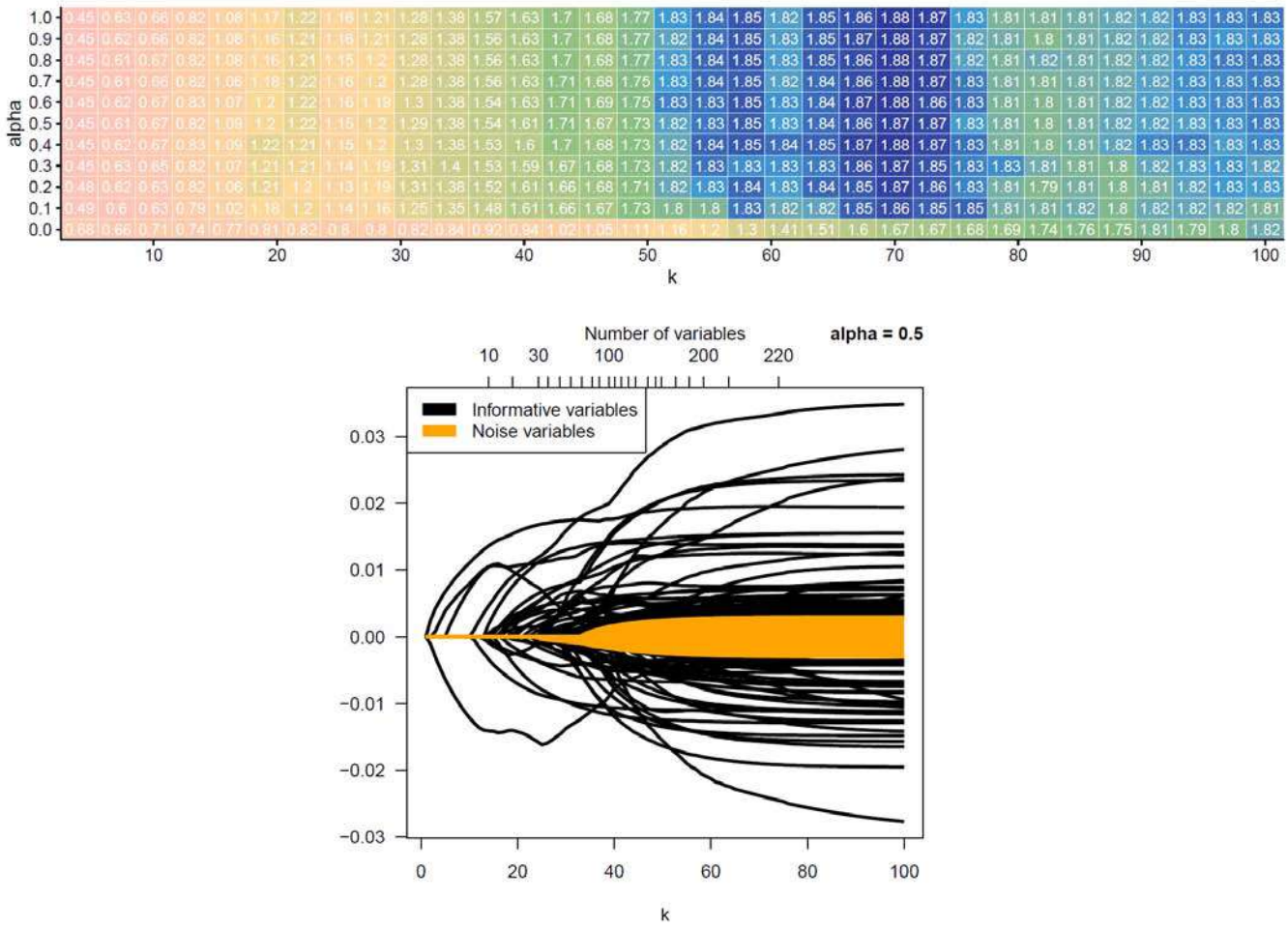
Figure 63: Elastic net regularization paths, from ridge regression (α=0) to the lasso (α=1).



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 64 shows the same results, with extra predictors containing nothing but noise (124 meaningful variables, uniformized, and 124 random uniform variables). The information ratio remains comparable, and almost all the noise variables are included in the model.

Figure 64: Lasso with noise variables



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

How worrying is this?

Having potentially irrelevant variables is actually not that worrying: if you want a lot of relevant variables in your model, it is inevitable that a few irrelevant ones sneak in.[31] Since those extra variables contain mostly noise, they almost cancel each other out and have a small effect on the forecasts.

While not worrying, those results are however disappointing: in this situation, i.e., when most of the predictors are relevant, when the variables are clean (outliers, etc. disappeared when we uniformized the data), when they are sufficiently different, when there is enough training data for the number of variables available, when there is no underlying sparse model, the lasso does not improve on linear regression.

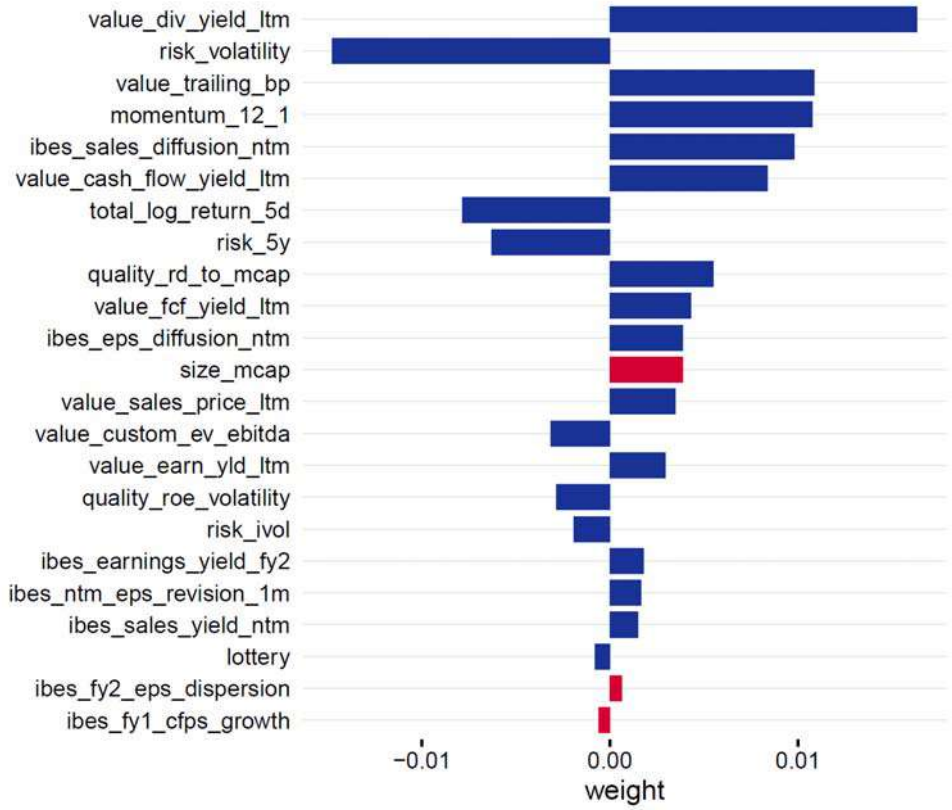
But this means we have a problem: we saw in the first part (figures 10 and 11) that unpenalized regression had a tendency to include similar variables with large coefficients and opposite signs. The apparent good out-of-sample performance could be misleading...

[31] There is a similar problem in biology: when testing tens of thousands of genes, we have to accept false positives – but we can control the false discovery rate (FDR).

One way of ensuring that the model does not overfit the data is to check if we can interpret it. In particular, we know the direction in which each factor should enter the model: for instance, earnings yield should have a positive influence on future returns while volatility should have a negative one.

We find that unexpected signs start to appear after around 20 variables (Figure 65): we can therefore consider the 20-variable model "safe" and the less penalized ones "suspicious".

Figure 65: Weights for a penalized regression, after some of the coefficients have appeared with an unexpected sign (red)



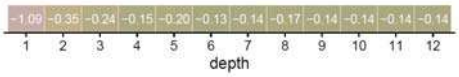
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Decision trees

For decision trees, we only let one parameter vary, the maximum tree depth (Figure 66): the performance increases as the tree is allowed to grow larger, and then stabilizes (the default is not to limit the tree size). However, it remains... negative.

While decision trees are valuable when you want interpretable models, or an interpretable approximation of a model, we cannot recommend them for general use, especially for regression problems.

Figure 66: Out-of-sample information ratio (IR) for the decision tree models, as the depth of the trees increase.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

We did not do any hyper-parameter fine-tuning for random forests and support vector machines [29] because fitting those models takes a very long time.

XGBoost

Let us now examine XGBoost. Figure 67 shows the out-of-sample information ratio (IR) for various combinations of the hyperparameters: step size, number of iterations and tree depth.

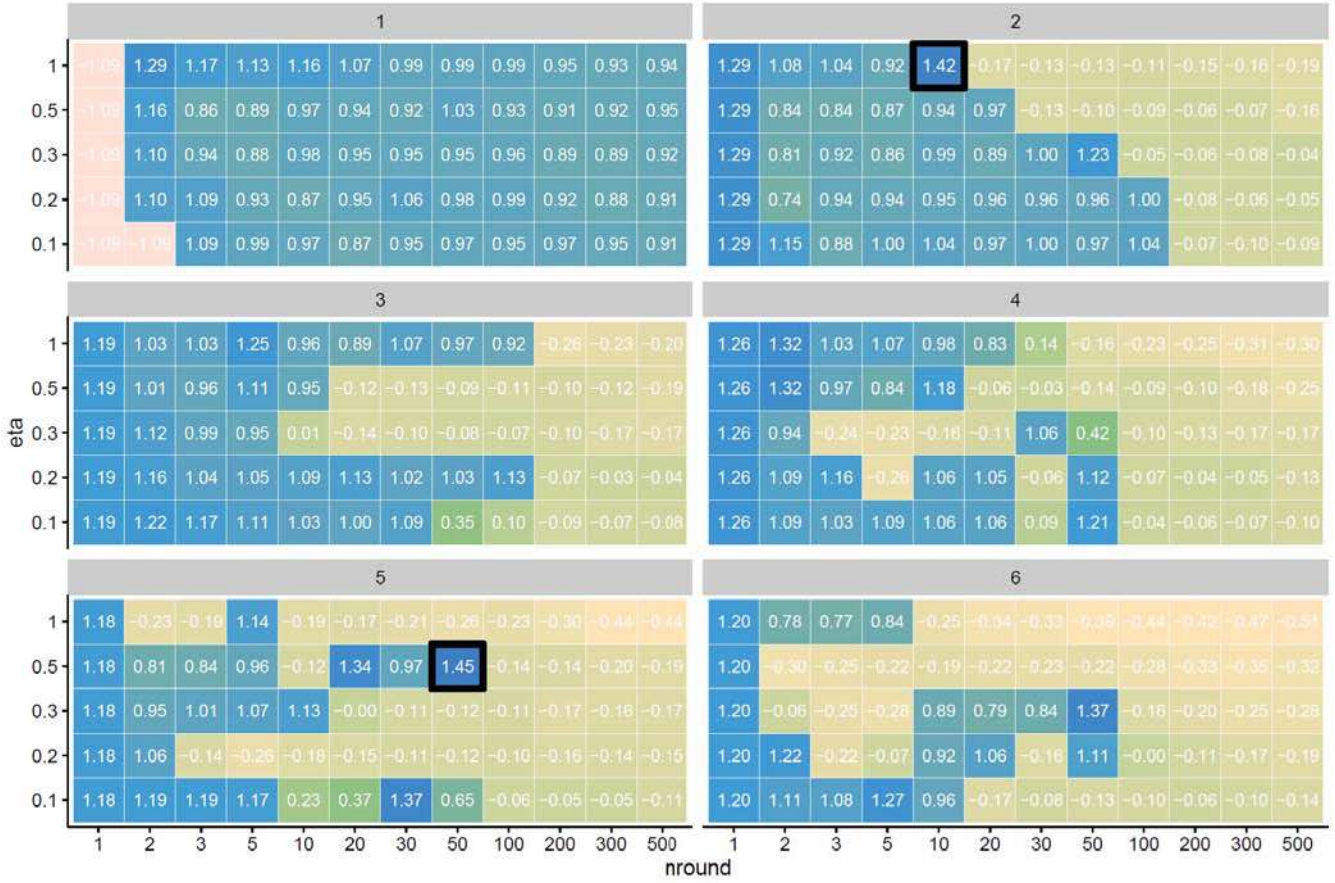
- The information ratio (IR) culminates at 1.45;[32]
- The optimal tree depth seems to be 2 or 5 (2 may be safer).
- The results are very noisy. In particular, we expect the performance to be approximately the same on the diagonals corresponding to a constant effective number of steps (product of the step size and the actual number of steps), and increase as the step size decreases: this effect is not clearly visible.

Figure 68 shows the trees entering the best depth-2 model. While a single tree is interpretable, it is unclear whether 10 of them remain so. This lack of interpretability, combined with the noise in the out-of-sample performance, casts doubts on the reliability of the model.

While XGBoost has a good reputation for classification problems, it may not be a good choice for regression problems.

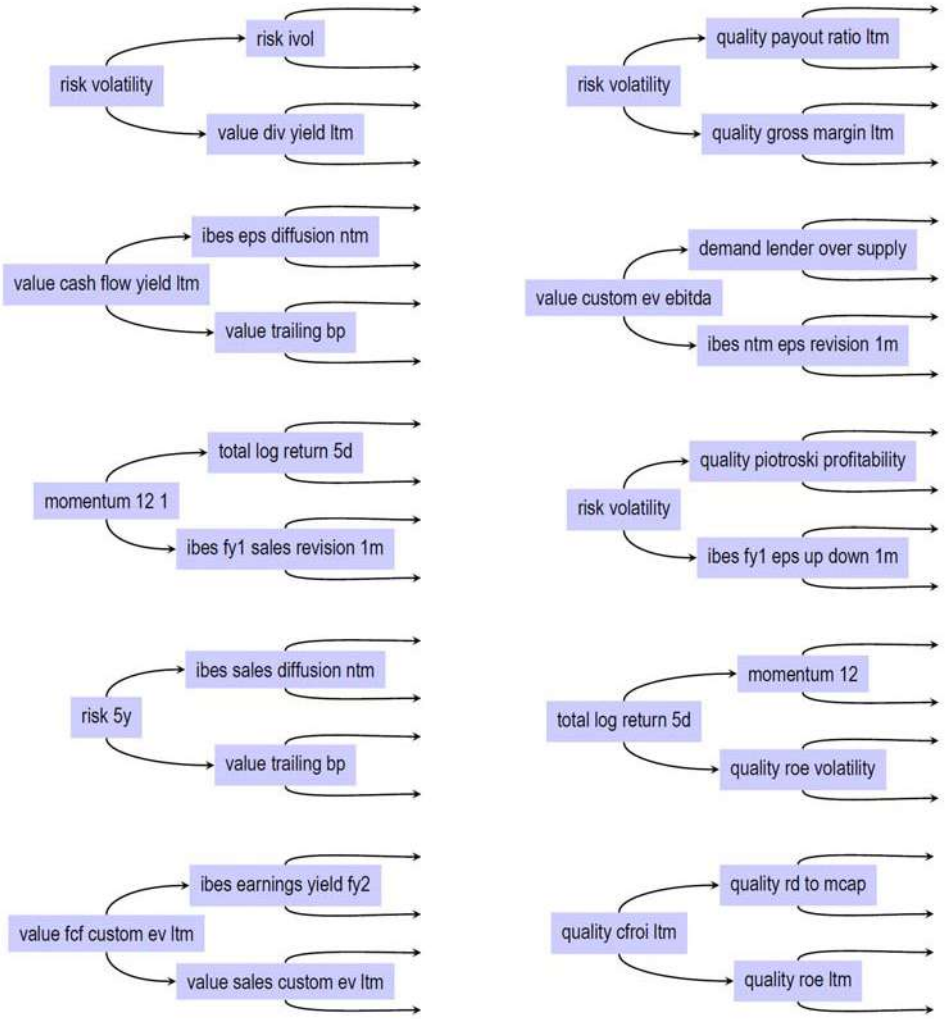
[32] The attentive reader will notice that the best information ratio is different from that announced in figure 25: the grid was different (the maximum was obtained for 150 steps) and *subsampling* was used.

Figure 67: Out-of-sample information ratio (IR) for the xgboost models, as a function of the hyperparameters, step size (η), number of iterations (nround), and tree depth (from 1 to 6). The best value is highlighted.



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 68: Trees in the XGBoost model



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

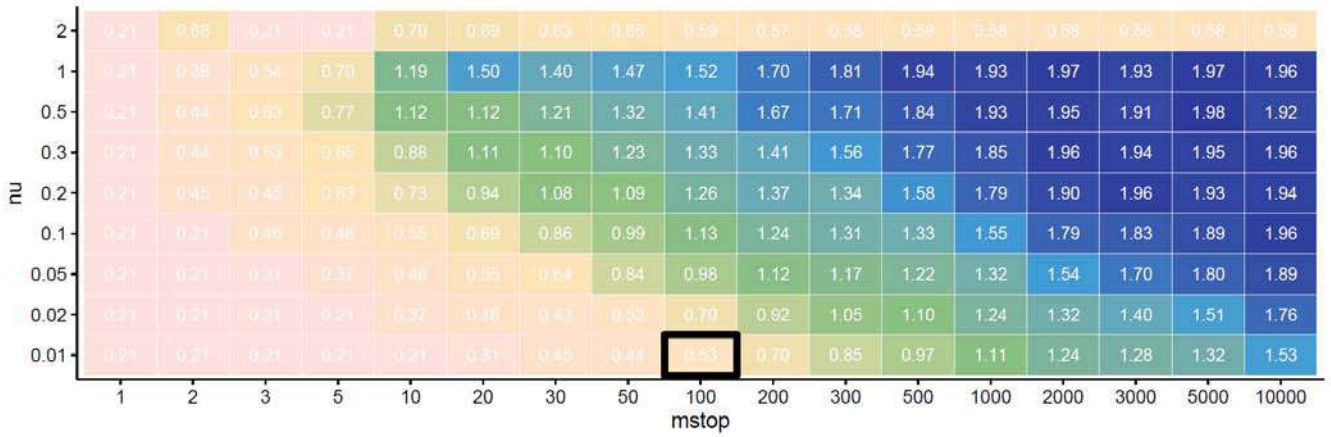
MBoost

Model boosting looks more promising (Figure 69).

- As expected, the performance depends mostly on the effective number of steps (the product of step size and actual number of steps).
- We were expecting the performance to increase along the diagonals as the step size decreases: this effect is not present, and a step size between 0.1 and 1 seems to be sufficient – the default is 0.01, and the usual advice is to decrease it further.
- The best performance is obtained for a very large number of steps (5,000) – the default (framed in black, on the plot) is 100 steps.

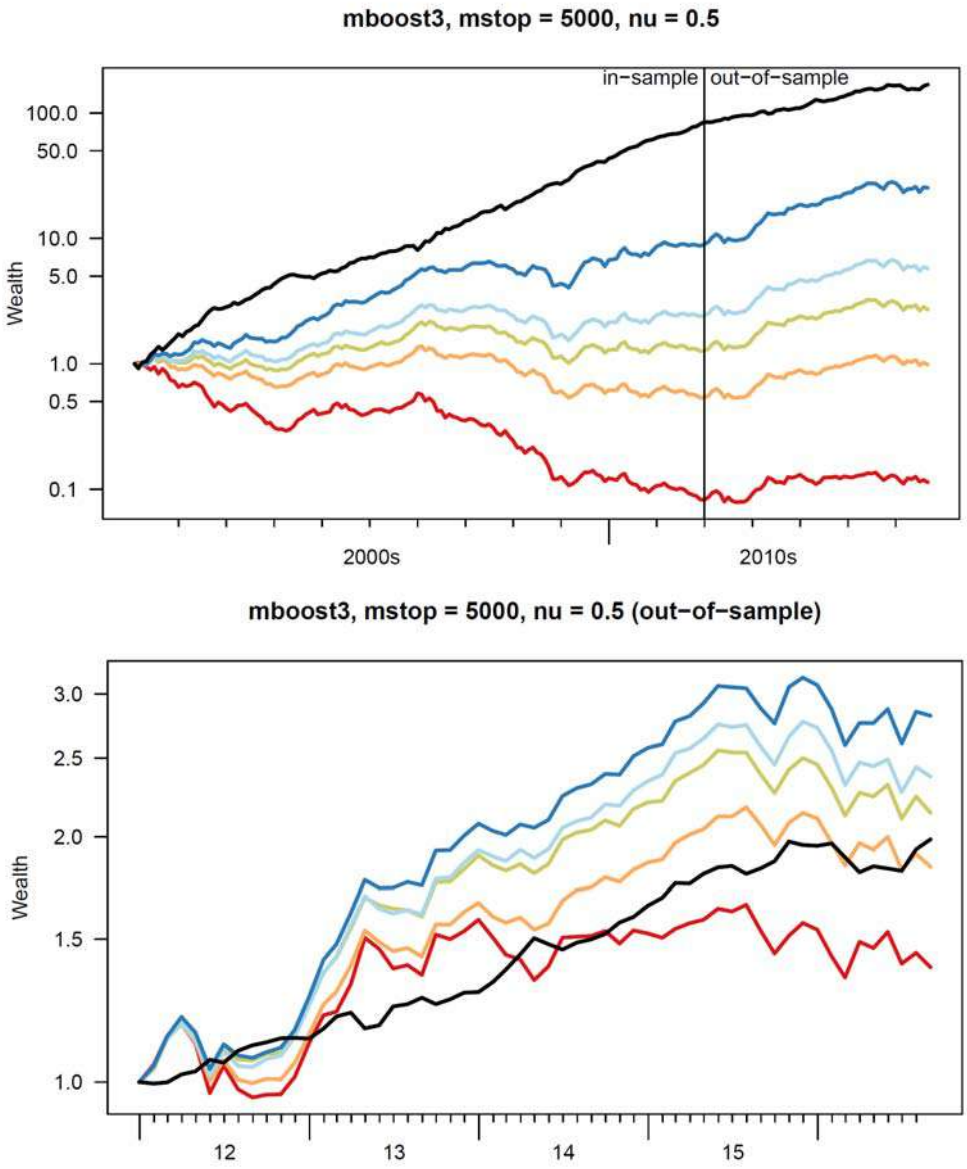
The grid search suggests that the best out-of-sample this model gives is 1.98 (Figure 70).

Figure 69: Out-of-sample information ratio (IR) for the mboost models, as a function of the hyperparameters



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 70: Performance of the best mboost model, in- and out-of-sample

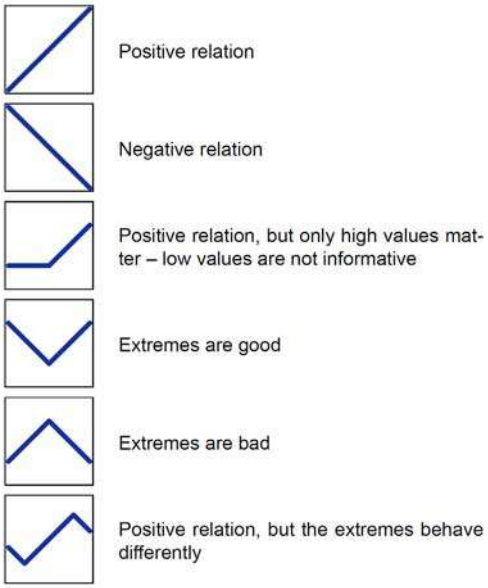


Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Since our boosting model uses 1-dimensional, non-linear models as base learners, we can check how variables enter the model and, as with the lasso, ensure that the model remains interpretable – we suspect that the best "out-of-sample" numbers are excessive.

Figure 71 shows some of the patterns we expect to see.

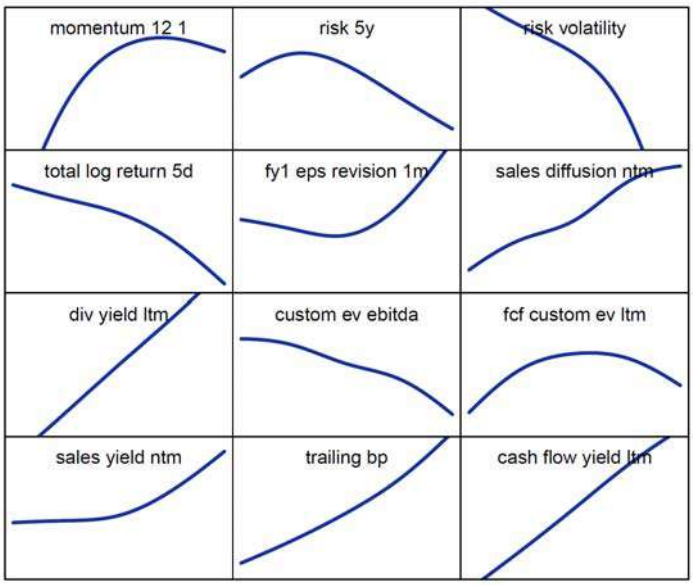
Figure 71: Some of the variable transformations we expect to see in the mboost model



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 72 shows how the first 12 variables enter the model: so far, so good.

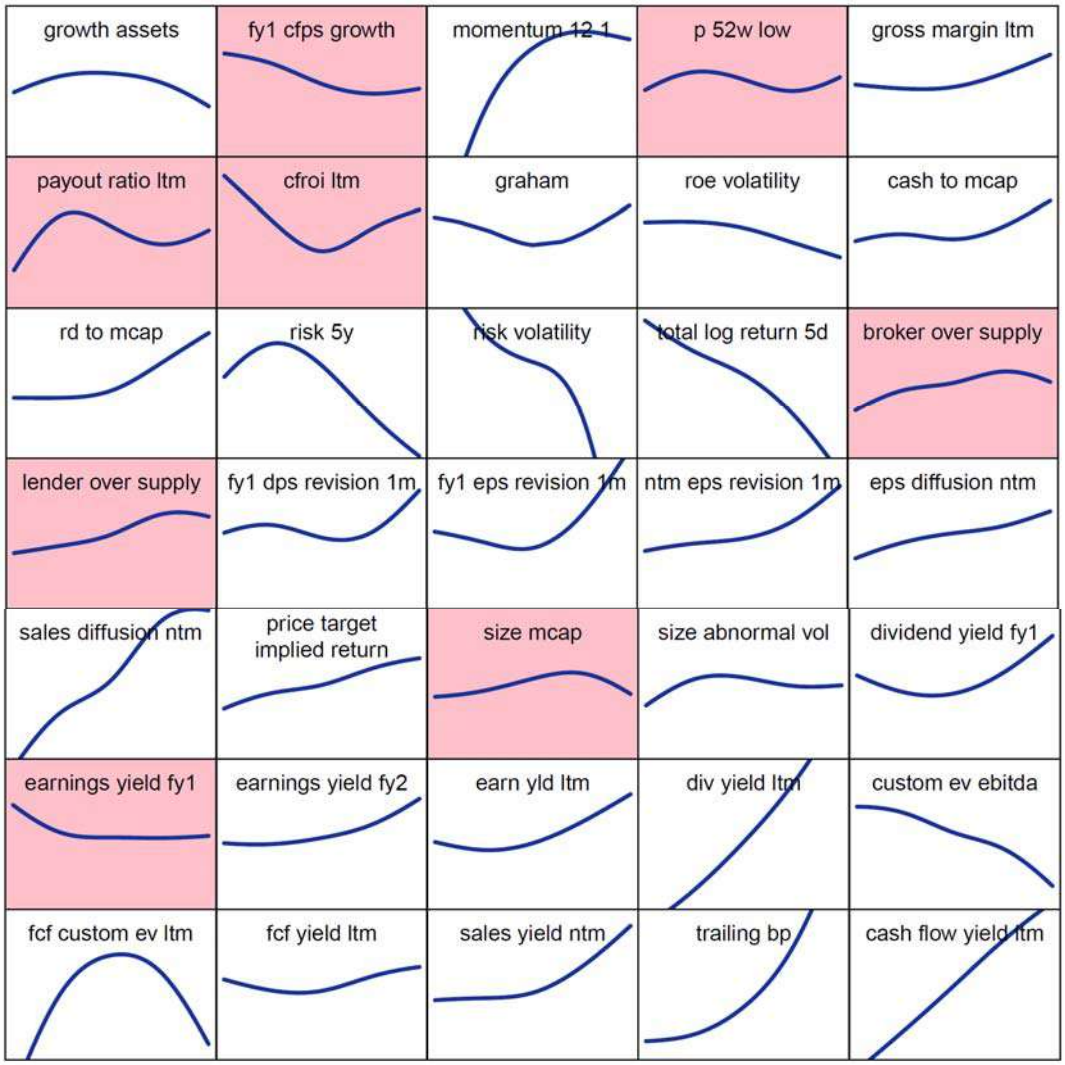
Figure 72: The first 12 variables to enter the mboost model (13 steps of size 0.5)



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 73 shows the variables in the model after 50 steps, with those entering the model in an unnatural way highlighted. The imaginative reader may be able to explain why those variables are transformed in these ways, but that task will require a lot of convincing and will become increasingly harder as the number of steps increases.

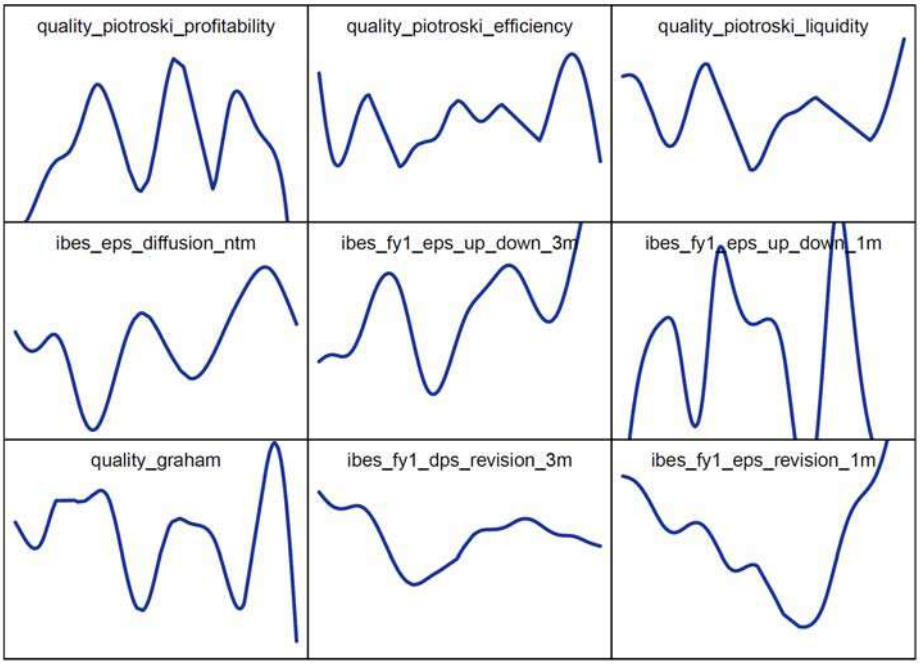
Figure 73: Components of the mboost model, after 50 steps of size 0.5; variables transformed in a suspicious way are highlighted



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

In case we have not frightened you enough, Figure 74 shows some of the transformations selected by another boosting model (using splines with 20 knots, the default for the tools we used, instead of 5, as in the rest of this document), with apparent good performance – the out-of-sample information ratio was 4.5.

Figure 74: Some of the suspicious variable transformations suggested by a model with an even better out-of-sample performance than that presented in the text (50,000 steps of size 1, with 20-knot splines): the out-of-sample information ratio was 4.5...



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

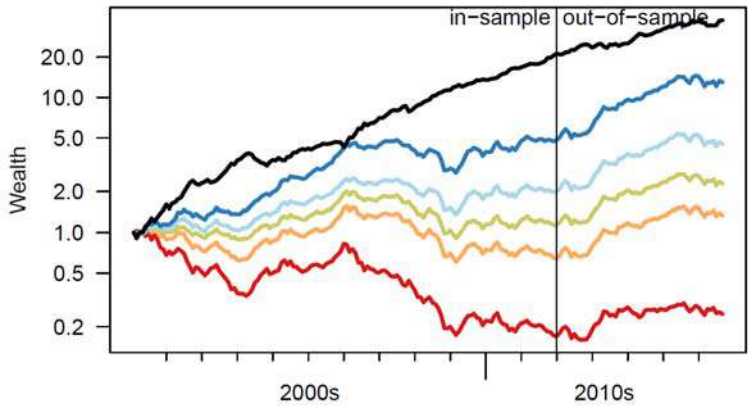
Constrained regression

Since we are concerned with the interpretability of the models, let us go back to basics and consider linear models. Our baseline model imposed not only the sign of the coefficients, but also their values (+1 or –1). Instead, we can only set the signs and let the values vary: this is a **constrained regression**.

$$\begin{aligned} &\text{Find} && \beta_0, \ldots, \beta_k \\ &\text{To minimize} && \sum_i \left(y_i - \beta_0 - \sum_j \beta_j x_{ij} \right)^2 \\ &\text{Such that} && \beta_1, \ldots, \beta_k \geqslant 0 \end{aligned}$$

Figure 75 and Figure 76 show that the performance is decent – better than what we have seen so far.

Figure 75: Constrained regression



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 76: Out-of-sample performance of the models examined

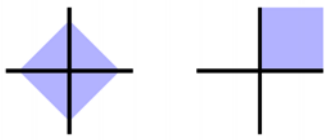
Strategy	CAGR (%)	AnnVol (%)	IR	tstat	Skew	Kurtosis	Hit Ratio	MaxDD (%)	VaR95 (%)	ES95 (%)	Turnover (%)
mboost	12.1	9.2	1.32	2.8	-0.22	-0.19	0.70	-9.8	-3.5	-4.8	138
lasso	11.6	9.5	1.22	2.6	-0.32	-0.15	0.70	-8.1	-3.5	-5.0	110
xgboost	5.1	6.5	0.78	1.7	0.19	-0.14	0.61	-9.8	-2.0	-3.2	141
baseline	5.7	11.7	0.49	1.2	-0.78	1.77	0.54	-17.6	-4.3	-7.5	76
constrained	13.0	9.3	1.39	2.9	-0.09	-0.21	0.70	-8.5	-3.6	-4.3	150

Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

This good performance can be explained as follows.

Sign-constrained regression and lasso are both constrained regressions: the shape of the constraints differ (Figure 77), but they have a similar regularizing effect. However, while the lasso constraints are not informative (they shrink the coefficients towards zero), the sign constraints are informative: they bring extra information into the model. In addition, the constrained regression is allowed to use as many variables as it wants (but the constraints still have a sparsifying effect: it uses less than half).

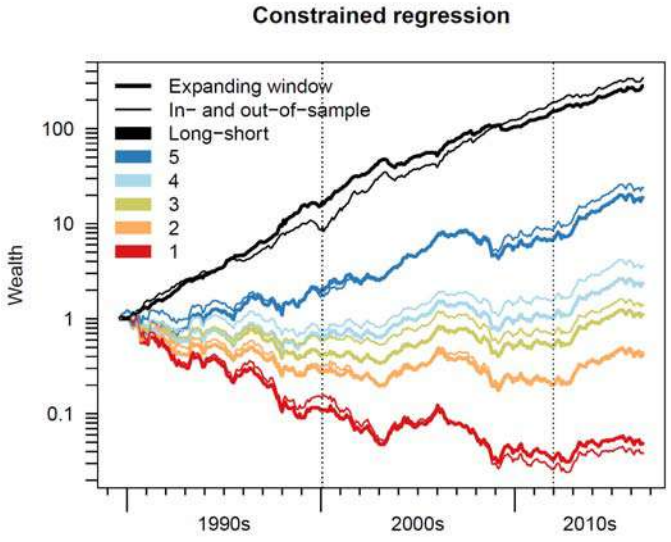
Figure 77: Constraints used in the lasso (left) and in the sign-constrained regression (in dimension 2)



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 78 and Figure 79 show the performance of models fitted in different training sets: our in-sample period (2000–2011 – only investible from 2012 onwards), a 3-year moving window (investible, reactive to market changes, but too noisy) and an expanding window (investible and stable).

Figure 78: Difference between the in-sample performance and that of an investible strategy



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 79: Performance, in different periods, with different training samples: 2000–2011 (the in-sample period used in the rest of the report), a 3-year moving window (the model is re-evaluated once a year), an expanding window.

Strategy	Period	CAGR (%)	AnnVol (%)	IR	tstat	Skew	Kurtosis	Hit Ratio	MaxDD (%)	VaR95 (%)	ES 95 (%)	Turnover (%)
expanding	1990-1999	32.7	11.7	2.80	7.9	0.61	2.36	0.81	-8.8	-2.4	-4.3	268
3y	1990-1999	31.9	11.4	2.80	8.0	0.37	2.24	0.82	-8.9	-2.7	-4.7	253
2000-2011	1990-1999	23.3	11.5	2.03	6.2	-0.24	0.66	0.74	-20.7	-4.0	-5.8	153
expanding	2000-2011	20.9	10.2	2.05	6.6	-0.75	1.96	0.76	-18.2	-2.3	-5.6	218
3y	2000-2011	15.7	13.7	1.15	3.9	-1.37	4.21	0.70	-34.4	-5.5	-9.6	150
2000-2011	2000-2011	30.4	12.9	2.35	7.4	-0.17	0.35	0.77	-19.1	-3.8	-6.0	139
expanding	2012-present	14.2	8.6	1.65	3.4	0.05	0.47	0.76	-7.2	-3.4	-4.0	211
3y	2012-present	12.5	9.0	1.39	2.9	-0.39	0.27	0.69	-9.3	-3.5	-5.2	178
2000-2011	2012-present	13.6	9.3	1.45	3.0	-0.13	-0.18	0.71	-8.5	-3.6	-4.3	150

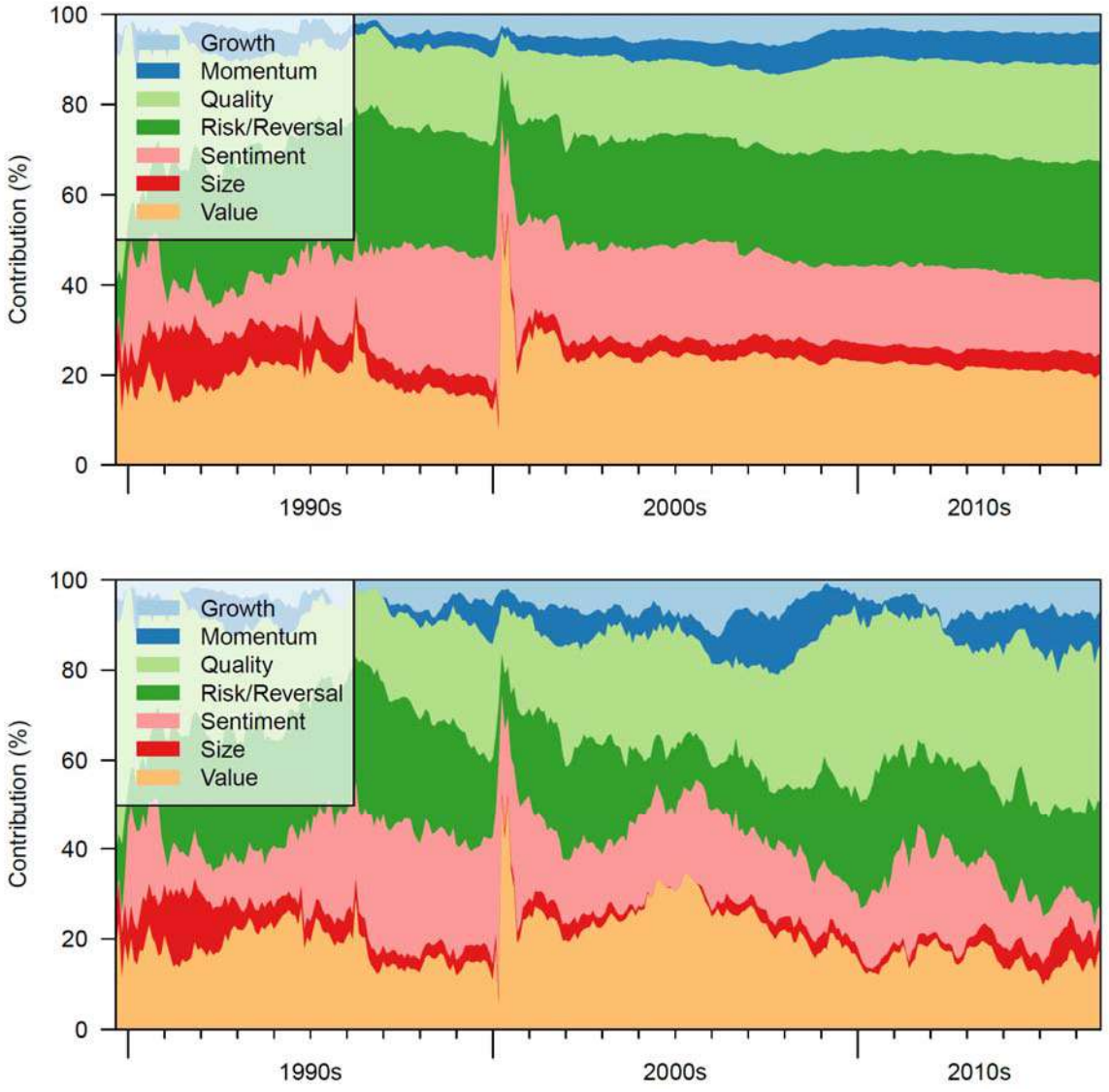
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 80 shows that the largest contributions are stable over time and come from value, risk, quality and sentiment factors.

Figure 81 shows the variables with the largest weights.

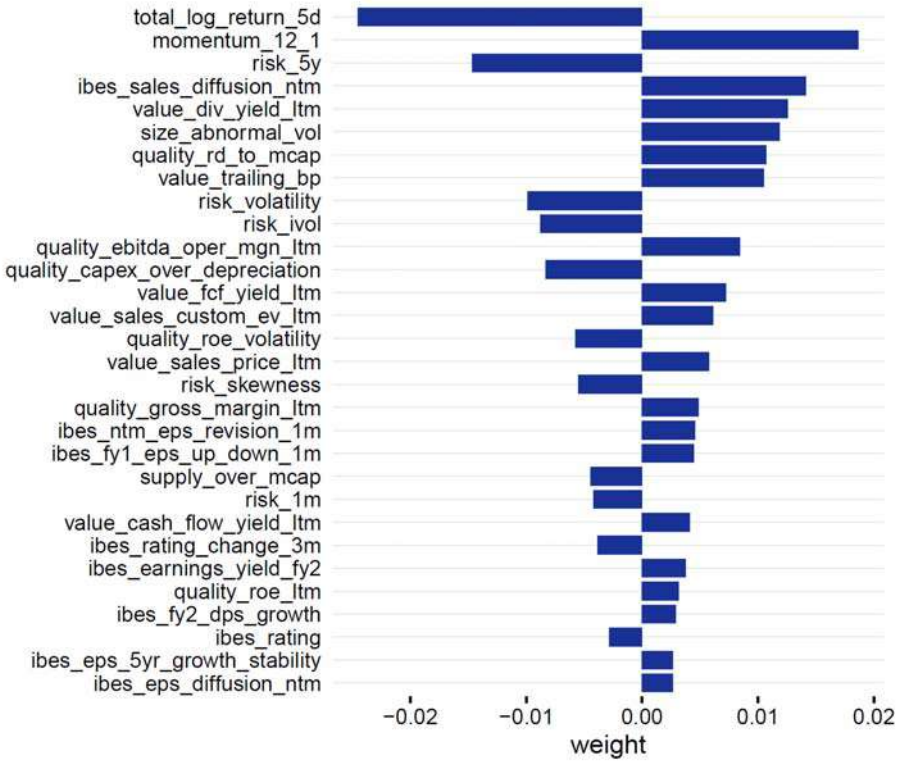
Figure 82 to Figure 84 show how the individual weights change with time.

Figure 80: Contributions to the constrained linear model, over time: the weights are stable, as one would expect from a model fitted on an expanding window. The glitch around 2000 is due to the absence of many of the variables before that: for the first few months after they appear, there is not enough history to have stable weights. The top plot uses an expanding window, the bottom a 3-year moving window: the contributions are similar, but the 3-year moving window model is more volatile.



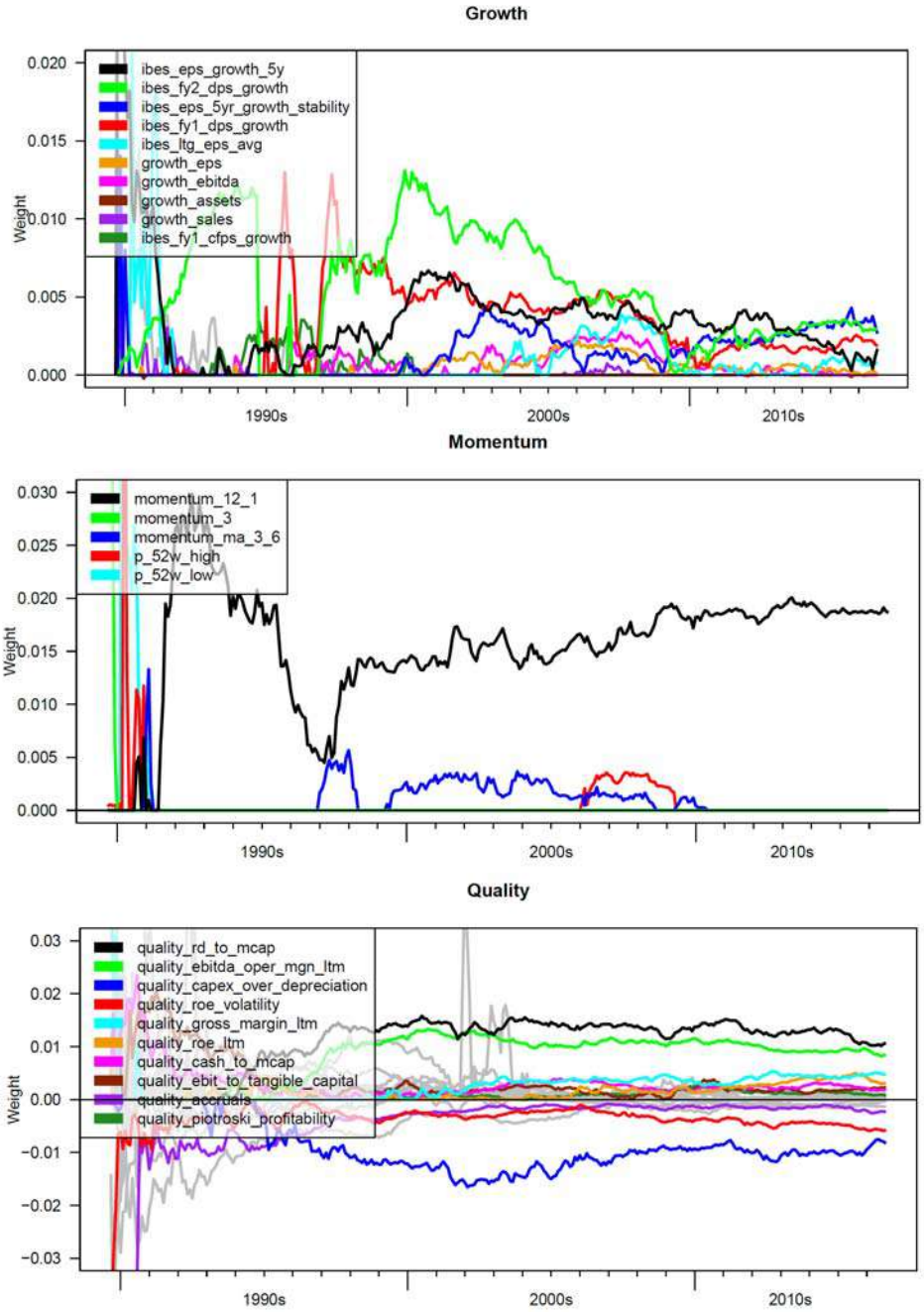
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 81: Largest weights in the sign-constrained model (expanding window)



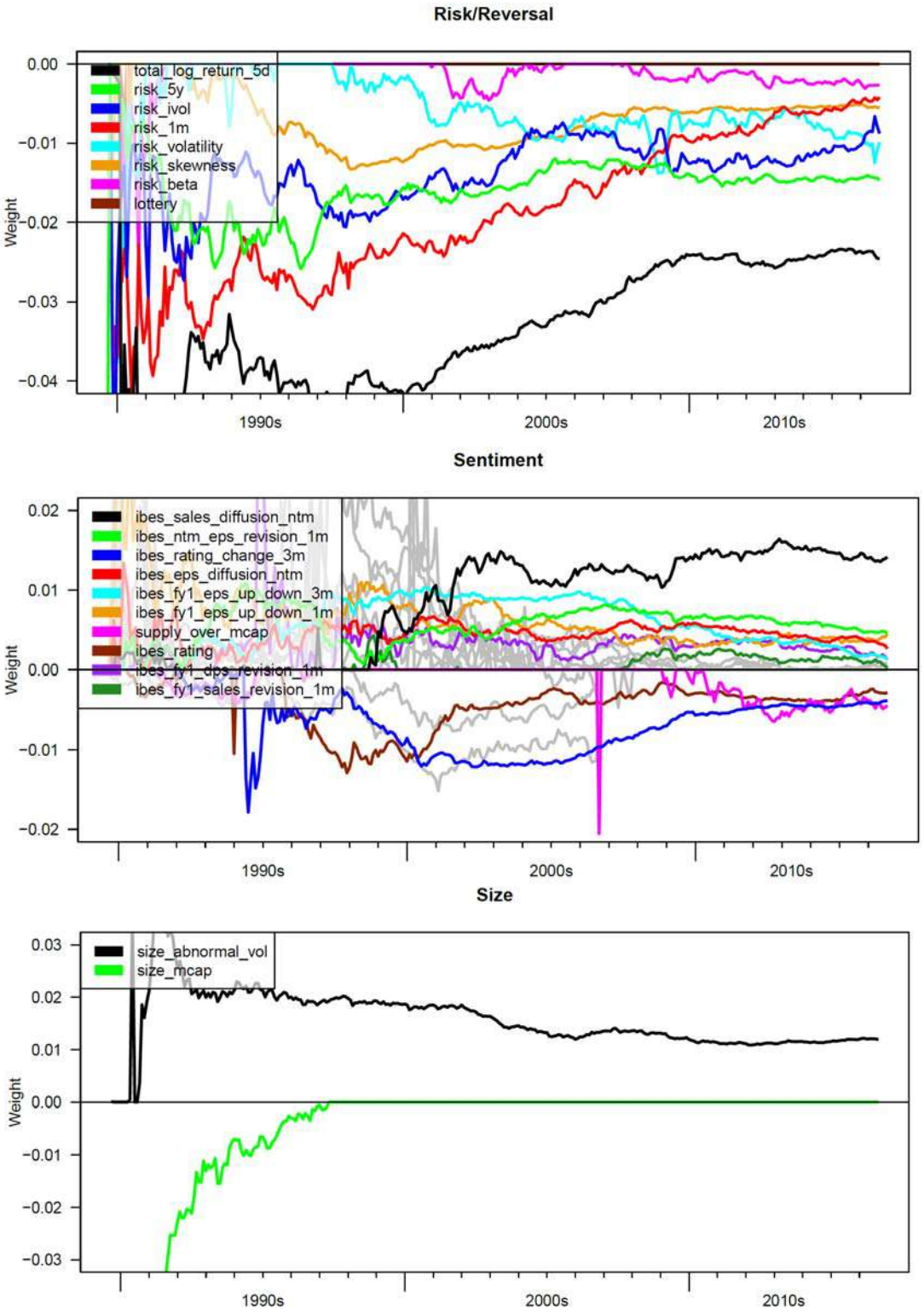
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 82: Weights of the individual factors (less important variables in grey)



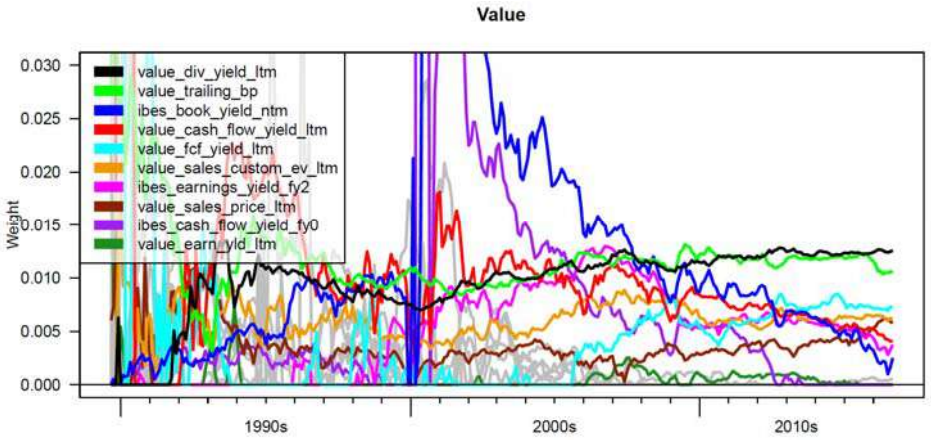
Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 83: Weights of the individual factors



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

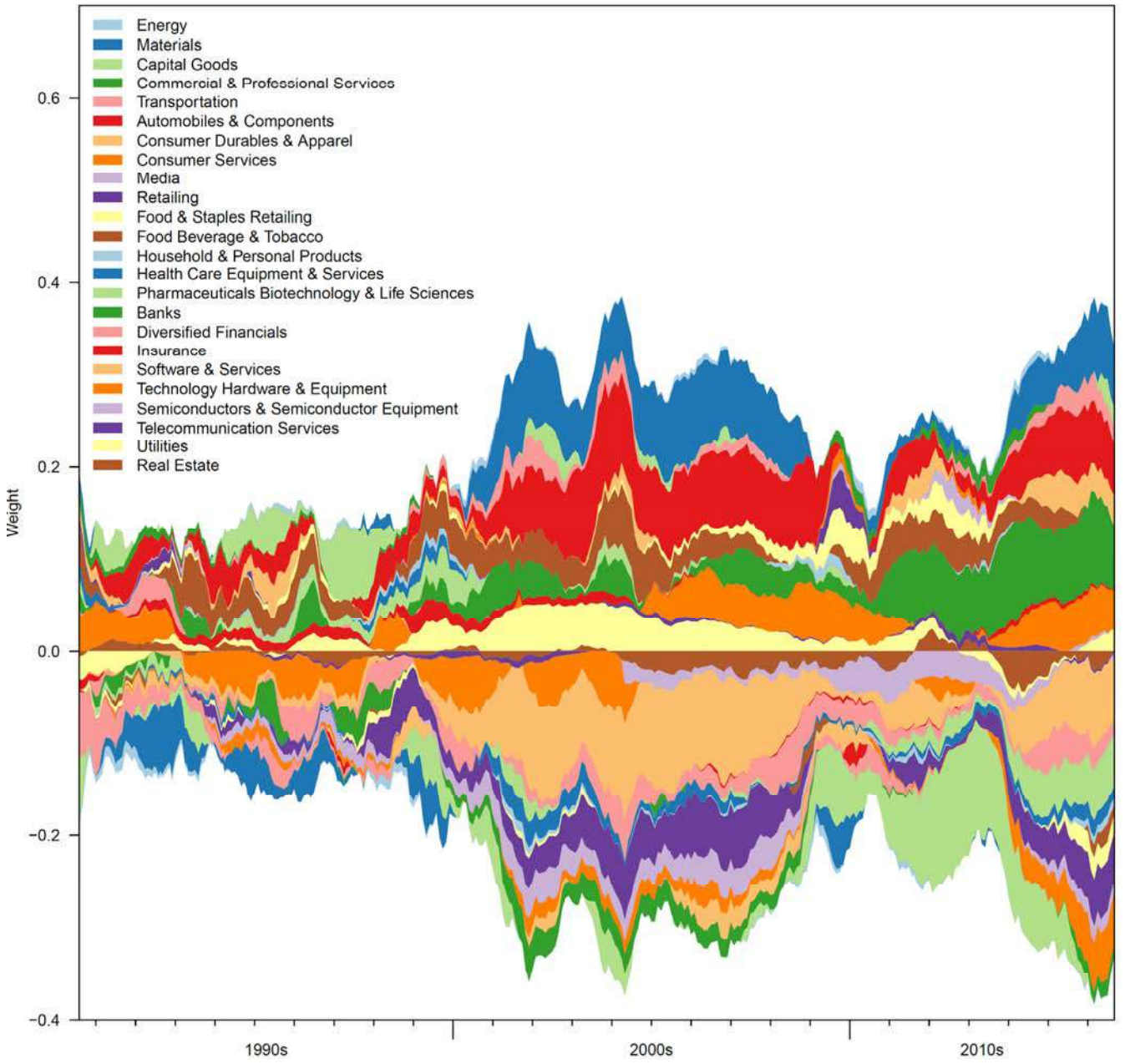
Figure 84: Weights of the individual factors



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 85 shows the current net sector composition of the long-short portfolio: Hardware, Banks, Automobiles and Materials are net long, while Software, Pharmaceuticals, Retail and Consumer services are net short – the model has significant sector biases.

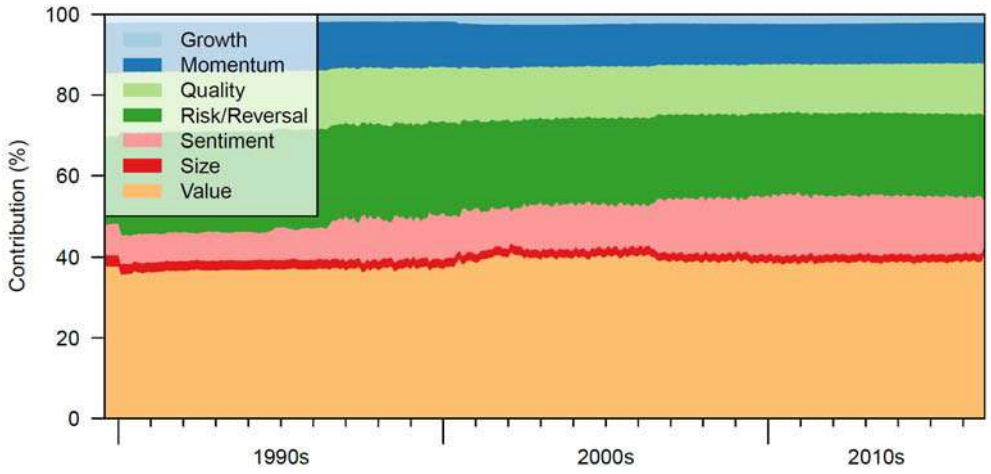
Figure 85: Sector exposure of the constrained linear model, over time



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

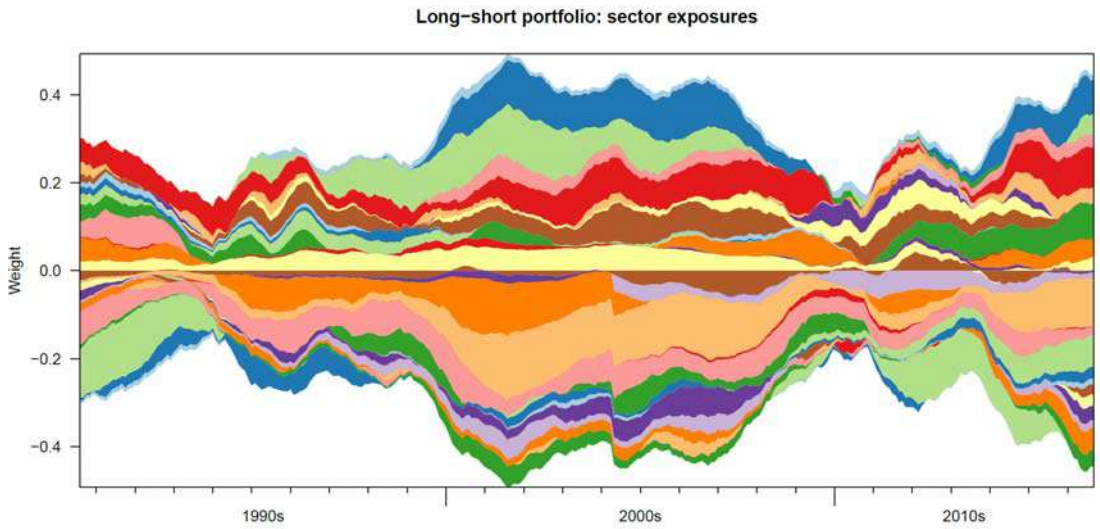
Figure 86 and Figure 87 show that the mboost model (here, estimated on our in-sample period, 2000-2011), gives similar results

Figure 86: Contributions to the mboost model, over time



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

Figure 87: Sector exposure of the mboost model, over time



Source: Factset, S&P, IBES, MarkIt, Thomson Reuters, Bloomberg Finance LP, Deutsche Bank Quantitative Strategy

When we started to write this report, we wanted to show two things:

- First, that machine-learning tools needed extra care in finance, that the traditional ways of preventing overfitting were ineffective;
- Second, that machine-learning tools were superior to older, statistical methods. While we confirmed the first point, the second is... still a work in progress.

Conclusion

In this report, we have examined a few machine learning algorithms (mostly penalized regression, decision trees, generalized additive models and boosting – but we also mentioned random forests and support vector machines) and laid out the main steps to apply them to financial data:

- Use a baseline model, e.g., a single financial ratio (e.g., E/P), or an unweighted average of a few known investment signals;
- Split your training data in at least two (and ideally three) samples: one to fit the model, one to fit the hyperparameters (and one to see how the final model fares); once you have selected your final model, re-estimate it on the whole dataset before using it;
- Do something with the missing values: use an algorithm that accepts them, or discard the corresponding observations, or replace them with some forecast or a random value;
- Avoid predictors with ties;
- At least uniformize the predictors – but see [3] for more options;
- Uniformize the variable to predict;
- Fine-tune the hyperparameters of your model of choice using grid search or, better, random search or Bayesian optimization; do not look for a single best model, but a family of increasingly better (and complex) models – a regularization path;
- The best model still overfits the data (the difference between in- and out-of-sample performance is too large; parts of the model that should be interpretable are not): constrain it further, i.e., choose one earlier on the regularization path;
- If you can find several unrelated promising models, combine them.

In particular, traditional ways of preventing overfitting are ineffective, or even dangerous, in finance.

While, contrary to our expectations, we have not retained any machine learning model, they still have some value and can help understand the data better:

- The lasso can be used to select variables, if you want a small number of variables, or if you know that many of the variables you have are irrelevant but do not know which ones;
- Likewise, boosting, with 1-variable non-linear models, can be used to select transformations of the variables – those transformed variables can then be included in other models;
- Decision trees do not have a very good predictive power, but they are eminently interpretable: they can be used to make sense of complex, black-box models.

As a last note, our stance on machine learning algorithms may have been too conservative:

- Often, apparently insignificant parameters of those algorithms play a big role: it may be possible to squeeze more performance out of these algorithms;
- Is our rejection of un-interpretable models with apparent good out-of-sample performance legitimate? Could it be that those models capture something un-interpretable but investable? We do not have the answer to that question (yet).

In forthcoming reports, we will continue our exploration of machine learning and plan to examine, among other topics:

- More models, in particular neural networks;
- Ensembling;
- Bayesian optimization for hyper-parameter fine-tuning;
- Other uses of Bayesian optimization, for instance to directly optimize the information ratio of a strategy, instead of the quality of the return forecasts.

References

[1] *Learning from data*
Y.S. Abu-Mostafa, Caltech, 2012
https://work.caltech.edu/telecourse.html,

[2] *JavaPlex tutorial*
H. Adams and A. Tausz, 2015
http://www.math.colostate.edu/~adams/research/javaplex_tutorial.pdf

[3] *Factor neutralization and beyond*
M.A. Alvarez et al., Deutsche Bank Quantitative Research, 2010.

[4] *Learning with submodular functions: a convex optimization perspective*
F. Bach, 2013
https://arxiv.org/abs/1111.6453

[5] *Bayesian reasoning and machine learning*
D. Barber, Cambridge University Press, 2012
http://web4.cs.ucl.ac.uk/staff/D.Barber/pmwiki/pmwiki.php?n=Brml.HomePage

[6] *Data stream mining: a practical approach*
A. Bifet and R. Kirkby, 2009
http://www.cs.waikato.ac.nz/~abifet/MOA/StreamMining.pdf

[7] *Getting the insiders' track*
K. Le Binh et al., Deutsche Bank Quantitative Research, 2016.

[8] *A survey of predictive modelling under imbalanced distributions*
P. Branco et al., 2015
http://arxiv.org/abs/1505.01658

[9] *Topology and data*
G. Carlsson, 2009
http://www.ams.org/images/carlsson-notes.pdf

[10] *Topological pattern recognition for point cloud data*
G. Carlsson, 2013
http://math.stanford.edu/~gunnar/actanumericathree.pdf

[11] *A boosting approach for automated trading*
G. Creamer and Y. Freund, 2006
http://papers. ssrn.com/sol3/papers.cfm?abstract_id=938042

[12] *Convolutional neural networks for visual recognition*
A. Karpathy et al., Stanford, 2016
http://cs231n.stanford.edu/

[13] *Introduction to the R package TDA*
B.T. Fasy et al., 2015
https://cran.r-project.org/web/packages/TDA/vignettes/article.pdf

[14] *Do we need hundreds of classifiers to solve real-world classification problems?*
M. Fernández-Delgado et al., 2014
http://jmlr.org/papers/ volume15/delgado14a/delgado14a.pdf

[15] *Barcodes: the persistent topology of data*
R. Ghrist, 2008
https://www.math.upenn.edu/~ghrist/preprints/barcodes.pdf

[16] *Three examples of applied and computational homology*
R. Ghrist, 2008
https://www.math. upenn.edu/~ghrist/preprints/nieuwarchief.pdf

[17] *The ladder: a reliable leaderboard for machine learning competitions*
M. Hardt and A. Blum, 2015
http://jmlr.org/proceedings/papers/v37/blum15.pdf

[18] *Statistical learning*
T. Hastie and R. Tibshirani, 2015
http://www.r-bloggers.com/in-depth-introduction-to-machine-learning-in-15-hours-of-expert-videos/

[19] *Elements of statistical learning*
T. Hastie, R. Tibshirani, and J. Friedman, Springer Verlag, 2009
http:// statweb.stanford.edu/~tibs/ElemStatLearn/

[20] *Statistical learning with sparsity: the lasso and generalizations*
T. Hastie, R. Tibshirani, and M. Wainwright, CRC Press, 2015.
http://web.stanford.edu/~hastie/StatLearnSparsity/

[21] *Neural networks for machine learning*
G. Hinton, 2012
https://www.coursera.org/learn/neural-networks

[22] *An introduction to statistical learning with applications in R*
G. James, D. Witten, T. Hastie, and R. Tibshirani, Springer Verlag, 2014.
http://www-bcf.usc.edu/~gareth/ISL/

[23] *caret: Classification and regression training*
M. Kuhn, 2016
https://CRAN.R-project.org/package=caret

[24] *Cubist models for regression,*
M. Kuhn et al., 2012
https://cran.r-project.org/web/packages/Cubist/vignettes/cubist.pdf

[25] *Facing our risk aversions*
A. Lau et al., Deutsche Bank Quantitative Research, 2014.

[26] *Introduction to machine learning,*
N. Lawrence, Machine Learning Summer School, Iceland, 2014
http://mlss2014.hiit.fi/slides.php

[27] *Interpretable classifiers using rules and Bayesian analysis: building a better stroke prediction model*
B. Letham et al., 2015
http://arxiv. org/abs/1511.01644

[28] *Intelligible models for classification and regression,*
Y. Lou et al. 2012
http://www.cs.cornell.edu/~yinlou/papers/lou-kdd12.pdf

[29] *How to scale up kernel methods to be as good as deep neural nets,*
Z. Lu 12 al., 2014
https://arxiv.org/abs/1411.4000

[30] *Recurrence plots for the analysis of complex systems*
N. Marwan et al., 2007
http://www.recurrence-plot.tk/

[31] *Machine learning: A probabilistic perspective*
K.R. Murphy, MIT Press, 2012.

[32] *Gradient boosting machines, a tutorial*
A. Natekin and A. Knoll, 2013
http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3885826/

[33] *Machine learning*
A. Ng, Stanford, Coursera, 2012
https://www.coursera.org/learn/ machine-learning

[34] *A roadmap for the computation of persistent homology*
N. Otter et al., 2015
https://arxiv.org/abs/1506.08903

[35] *Forecast combinations in R using the ForecastCombinations package*
E. Raviv, 2011
http://eranraviv.com/wp-content/uploads/2011/09/VIG3-1.pdf

[36] *Computational topology for point data: Betti numbers of α-shapes*
V. Robins, 2002
http://people.physics.anu.edu.au/~vbr110/papers/lnp.pdf

[37] *What is the shape of the risk-return relation?*
A. Rossi and A. Timmermann, 2010
http://papers.ssrn.com/sol3/papers.cfm?abstract_id=1364750

[38] *Confidence bounds of recurrence-based complexity measures*
S. Schinkel et al., 2009
https://www.pik-potsdam.de/members/kurths/publikationen/2009/phys-lett-a-schinkel.pdf

[39] *FaceNet: a unified embedding for face recognition and clustering*
F. Schroff et al., 2015
https://arxiv.org/abs/1503.03832

[40] *Selfieboost: a boosting algorithm for deep learning*
S. Shalev-Shwarts, 2014
http://arxiv. org/abs/1411.3436,

[41] *Reinforcement learning*
D. Silver, University College London, 2015
http://www0.cs.ucl. ac.uk/staff/d.silver/web/Teaching.html

[42] *Machine learning markets,*
A. Storkey, 2011
http://homepages.inf.ed.ac.uk/amos/
publications/Storkey2011MachineLearningMarkets.pdf,

[43] *Are random classifiers truly the best classifiers?*
M. Wainberg et al., 2016
http://jmlr.org/papers/volume17/15-374/15-374.pdf

[44] *Falling rule lists*
F. Wang and C. Rudin, 2014
http://arxiv.org/abs/1411.5899

[45] *The rise of the machines*
S. Wang et al., Deutsche Bank Quantitative Research, 2012

[46] *The rise of the machines II*
S. Wang et al., Deutsche Bank Quantitative Research, 2013

[47] *The rise of the machines III*,
S. Wang et al., Deutsche Bank Quantitative Research, 2014

[48] *What the no free lunch theorems really mean; how to improve search algorithms*
D.H. Wolpert, 2012
http://www.santafe.edu/media/workingpapers/12-10-017.pdf

[49] *No free lunch theorems for optimization*
D.H. Wolpert and W.G. Macready, 1997
https:// ti.arc.nasa.gov/m/profile/dhw/papers/78.pdf,

[50] *Scalable Bayesian rule lists*
H. Yang et al., 2016
http://arxiv.org/ abs/1602.08610

[51] *Quality investing in Asia: seeing the forest through the trees*
V. Zoonekynd et al., Deutsche Bank Quantitative Research, 2014.

[52] *A first momentum playbook*
V. Zoonekynd et al., Deutsche Bank Quantitative Research, 2014.

[53] *Framing momentum*
V. Zoonekynd et al., Deutsche Bank Quantitative Research, 2014.

[54] *Timing momentum*
V. Zoonekynd et al., Deutsche Bank Quantitative Research, 2015.

Appendix 1

Important Disclosures

Additional information available upon request

*Prices are current as of the end of the previous trading session unless otherwise indicated and are sourced from local exchanges via Reuters, Bloomberg and other vendors . Other information is sourced from Deutsche Bank, subject companies, and other sources. For disclosures pertaining to recommendations or estimates made on securities other than the primary subject of this research, please see the most recently published company report or visit our global disclosure look-up page on our website at http://gm.db.com/ger/disclosure/DisclosureDirectory.eqsr

Analyst Certification

The views expressed in this report accurately reflect the personal views of the undersigned lead analyst(s). In addition, the undersigned lead analyst(s) has not and will not receive any compensation for providing a specific recommendation or view in this report. Vincent Zoonekynd/Khoi LeBinh/Ada Lau/Hemant Sambatur

Hypothetical Disclaimer

Backtested, hypothetical or simulated performance results have inherent limitations. Unlike an actual performance record based on trading actual client portfolios, simulated results are achieved by means of the retroactive application of a backtested model itself designed with the benefit of hindsight. Taking into account historical events the backtesting of performance also differs from actual account performance because an actual investment strategy may be adjusted any time, for any reason, including a response to material, economic or market factors. The backtested performance includes hypothetical results that do not reflect the reinvestment of dividends and other earnings or the deduction of advisory fees, brokerage or other commissions, and any other expenses that a client would have paid or actually paid. No representation is made that any trading strategy or account will or is likely to achieve profits or losses similar to those shown. Alternative modeling techniques or assumptions might produce significantly different results and prove to be more appropriate. Past hypothetical backtest results are neither an indicator nor guarantee of future returns. Actual results will vary, perhaps materially, from the analysis.

Regulatory Disclosures

1.Important Additional Conflict Disclosures

Aside from within this report, important conflict disclosures can also be found at https://gm.db.com/equities under the "Disclosures Lookup" and "Legal" tabs. Investors are strongly encouraged to review this information before investing.

2.Short-Term Trade Ideas

Deutsche Bank equity research analysts sometimes have shorter-term trade ideas (known as SOLAR ideas) that are consistent or inconsistent with Deutsche Bank's existing longer term ratings. These trade ideas can be found at the SOLAR link at http://gm.db.com.

Additional Information

The information and opinions in this report were prepared by Deutsche Bank AG or one of its affiliates (collectively "Deutsche Bank"). Though the information herein is believed to be reliable and has been obtained from public sources believed to be reliable, Deutsche Bank makes no representation as to its accuracy or completeness.

If you use the services of Deutsche Bank in connection with a purchase or sale of a security that is discussed in this report, or is included or discussed in another communication (oral or written) from a Deutsche Bank analyst, Deutsche Bank may act as principal for its own account or as agent for another person.

Deutsche Bank may consider this report in deciding to trade as principal. It may also engage in transactions, for its own account or with customers, in a manner inconsistent with the views taken in this research report. Others within Deutsche Bank, including strategists, sales staff and other analysts, may take views that are inconsistent with those taken in this research report. Deutsche Bank issues a variety of research products, including fundamental analysis, equity-linked analysis, quantitative analysis and trade ideas. Recommendations contained in one type of communication may differ from recommendations contained in others, whether as a result of differing time horizons, methodologies or otherwise. Deutsche Bank and/or its affiliates may also be holding debt or equity securities of the issuers it writes on. Analysts are paid in part based on the profitability of Deutsche Bank AG and its affiliates, which includes investment banking revenues.

Opinions, estimates and projections constitute the current judgment of the author as of the date of this report. They do not necessarily reflect the opinions of Deutsche Bank and are subject to change without notice. Deutsche Bank research analysts sometimes have shorter-term trade ideas that are consistent or inconsistent with Deutsche Bank's existing longer term ratings. These trade ideas for equities can be found at the SOLAR link at http://gm.db.com. A SOLAR idea represents a high conviction belief by an analyst that a stock will outperform or underperform the market and/or sector delineated over a time frame of no less than two weeks. In addition to SOLAR ideas, the analysts named in this report may have from time to time discussed with our clients, including Deutsche Bank salespersons and traders, or may discuss in this report or elsewhere, trading strategies or ideas that reference catalysts or events that may have a near-term or medium-term impact on the market price of the securities discussed in this report, which impact may be directionally counter to the analysts' current 12-month view of total return as described herein. Deutsche Bank has no obligation to update, modify or amend this report or to otherwise notify a recipient thereof if any opinion, forecast or estimate contained herein changes or subsequently becomes inaccurate. Coverage and the frequency of changes in market conditions and in both general and company specific economic prospects makes it difficult to update research at defined intervals. Updates are at the sole discretion of the coverage analyst concerned or of the Research Department Management and as such the majority of reports are published at irregular intervals. This report is provided for informational purposes only. It is not an offer or a solicitation of an offer to buy or sell any financial instruments or to participate in any particular trading strategy. Target prices are inherently imprecise and a product of the analyst's judgment. The financial instruments discussed in this report may not be suitable for all investors and investors must make their own informed investment decisions. Prices and availability of financial instruments are subject to change without notice and investment transactions can lead to losses as a result of price fluctuations and other factors. If a financial instrument is denominated in a currency other than an investor's currency, a change in exchange rates may adversely affect the investment. Past performance is not necessarily indicative of future results. Unless otherwise indicated, prices are current as of the end of the previous trading session, and are sourced from local exchanges via Reuters, Bloomberg and other vendors. Data is sourced from Deutsche Bank, subject companies, and in some cases, other parties.

The Deutsche Bank Research Department is independent of other business areas divisions of the Bank. Details regarding our organizational arrangements and information barriers we have to prevent and avoid conflicts of interest with respect to our research is available on our website under Disclaimer found on the Legal tab.

Macroeconomic fluctuations often account for most of the risks associated with exposures to instruments that promise to pay fixed or variable interest rates. For an investor who is long fixed rate instruments (thus receiving these cash flows), increases in interest rates naturally lift the discount factors applied to the expected cash flows and thus cause a

loss. The longer the maturity of a certain cash flow and the higher the move in the discount factor, the higher will be the loss. Upside surprises in inflation, fiscal funding needs, and FX depreciation rates are among the most common adverse macroeconomic shocks to receivers. But counterparty exposure, issuer creditworthiness, client segmentation, regulation (including changes in assets holding limits for different types of investors), changes in tax policies, currency convertibility (which may constrain currency conversion, repatriation of profits and/or the liquidation of positions), and settlement issues related to local clearing houses are also important risk factors to be considered. The sensitivity of fixed income instruments to macroeconomic shocks may be mitigated by indexing the contracted cash flows to inflation, to FX depreciation, or to specified interest rates – these are common in emerging markets. It is important to note that the index fixings may -- by construction -- lag or mis-measure the actual move in the underlying variables they are intended to track. The choice of the proper fixing (or metric) is particularly important in swaps markets, where floating coupon rates (i.e., coupons indexed to a typically short-dated interest rate reference index) are exchanged for fixed coupons. It is also important to acknowledge that funding in a currency that differs from the currency in which coupons are denominated carries FX risk. Naturally, options on swaps (swaptions) also bear the risks typical to options in addition to the risks related to rates movements.

Derivative transactions involve numerous risks including, among others, market, counterparty default and illiquidity risk. The appropriateness or otherwise of these products for use by investors is dependent on the investors' own circumstances including their tax position, their regulatory environment and the nature of their other assets and liabilities, and as such, investors should take expert legal and financial advice before entering into any transaction similar to or inspired by the contents of this publication. The risk of loss in futures trading and options, foreign or domestic, can be substantial. As a result of the high degree of leverage obtainable in futures and options trading, losses may be incurred that are greater than the amount of funds initially deposited. Trading in options involves risk and is not suitable for all investors. Prior to buying or selling an option investors must review the "Characteristics and Risks of Standardized Options", at http://www.optionsclearing.com/about/publications/character-risks.jsp. If you are unable to access the website please contact your Deutsche Bank representative for a copy of this important document.

Participants in foreign exchange transactions may incur risks arising from several factors, including the following: (i) exchange rates can be volatile and are subject to large fluctuations; (ii) the value of currencies may be affected by numerous market factors, including world and national economic, political and regulatory events, events in equity and debt markets and changes in interest rates; and (iii) currencies may be subject to devaluation or government imposed exchange controls which could affect the value of the currency. Investors in securities such as ADRs, whose values are affected by the currency of an underlying security, effectively assume currency risk.
Unless governing law provides otherwise, all transactions should be executed through the Deutsche Bank entity in the investor's home jurisdiction.

United States: Approved and/or distributed by Deutsche Bank Securities Incorporated, a member of FINRA, NFA and SIPC. Analysts employed by non-US affiliates may not be associated persons of Deutsche Bank Securities Incorporated and therefore not subject to FINRA regulations concerning communications with subject companies, public appearances and securities held by analysts.

Germany: Approved and/or distributed by Deutsche Bank AG, a joint stock corporation with limited liability incorporated in the Federal Republic of Germany with its principal office in Frankfurt am Main. Deutsche Bank AG is authorized under German Banking Law and is subject to supervision by the European Central Bank and by BaFin, Germany's Federal Financial Supervisory Authority.

United Kingdom: Approved and/or distributed by Deutsche Bank AG acting through its London Branch at Winchester House, 1 Great Winchester Street, London EC2N 2DB. Deutsche Bank AG in the United Kingdom is authorised by the Prudential Regulation Authority and is subject to limited regulation by the Prudential Regulation Authority and Financial Conduct Authority. Details about the extent of our authorisation and regulation are available on request.

Hong Kong: Distributed by Deutsche Bank AG, Hong Kong Branch.

India: Prepared by Deutsche Equities India Pvt Ltd, which is registered by the Securities and Exchange Board of India (SEBI) as a stock broker. Research Analyst SEBI Registration Number is INH000001741. DEIPL may have received administrative warnings from the SEBI for breaches of Indian regulations.

Japan: Approved and/or distributed by Deutsche Securities Inc.(DSI). Registration number - Registered as a financial instruments dealer by the Head of the Kanto Local Finance Bureau (Kinsho) No. 117. Member of associations: JSDA, Type II Financial Instruments Firms Association and The Financial Futures Association of Japan. Commissions and risks involved in stock transactions - for stock transactions, we charge stock commissions and consumption tax by multiplying the transaction amount by the commission rate agreed with each customer. Stock transactions can lead to losses as a result of share price fluctuations and other factors. Transactions in foreign stocks can lead to additional losses stemming from foreign exchange fluctuations. We may also charge commissions and fees for certain categories of investment advice, products and services. Recommended investment strategies, products and services carry the risk of losses to principal and other losses as a result of changes in market and/or economic trends, and/or fluctuations in market value. Before deciding on the purchase of financial products and/or services, customers should carefully read the relevant disclosures, prospectuses and other documentation. "Moody's", "Standard & Poor's", and "Fitch" mentioned in this report are not registered credit rating agencies in Japan unless Japan or "Nippon" is specifically designated in the name of the entity. Reports on Japanese listed companies not written by analysts of DSI are written by Deutsche Bank Group's analysts with the coverage companies specified by DSI. Some of the foreign securities stated on this report are not disclosed according to the Financial Instruments and Exchange Law of Japan.

Korea: Distributed by Deutsche Securities Korea Co.

South Africa: Deutsche Bank AG Johannesburg is incorporated in the Federal Republic of Germany (Branch Register Number in South Africa: 1998/003298/10).

Singapore: by Deutsche Bank AG, Singapore Branch or Deutsche Securities Asia Limited, Singapore Branch (One Raffles Quay #18-00 South Tower Singapore 048583, +65 6423 8001), which may be contacted in respect of any matters arising from, or in connection with, this report. Where this report is issued or promulgated in Singapore to a person who is not an accredited investor, expert investor or institutional investor (as defined in the applicable Singapore laws and regulations), they accept legal responsibility to such person for its contents.

Taiwan: Information on securities/investments that trade in Taiwan is for your reference only. Readers should independently evaluate investment risks and are solely responsible for their investment decisions. Deutsche Bank research may not be distributed to the Taiwan public media or quoted or used by the Taiwan public media without written consent. Information on securities/instruments that do not trade in Taiwan is for informational purposes only and is not to be construed as a recommendation to trade in such securities/instruments. Deutsche Securities Asia Limited, Taipei Branch may not execute transactions for clients in these securities/instruments.

Qatar: Deutsche Bank AG in the Qatar Financial Centre (registered no. 00032) is regulated by the Qatar Financial Centre Regulatory Authority. Deutsche Bank AG - QFC Branch may only undertake the financial services activities that fall within the scope of its existing QFCRA license. Principal place of business in the QFC: Qatar Financial Centre, Tower, West Bay, Level 5, PO Box 14928, Doha, Qatar. This information has been distributed by Deutsche Bank AG. Related financial products or services are only available to Business Customers, as defined by the Qatar Financial Centre Regulatory Authority.

Russia: This information, interpretation and opinions submitted herein are not in the context of, and do not constitute, any appraisal or evaluation activity requiring a license in the Russian Federation.

Kingdom of Saudi Arabia: Deutsche Securities Saudi Arabia LLC Company, (registered no. 07073-37) is regulated by the Capital Market Authority. Deutsche Securities Saudi Arabia may only undertake the financial services activities that fall within the scope of its existing CMA license. Principal place of business in Saudi Arabia: King Fahad Road, Al Olaya District, P.O. Box 301809, Faisaliah Tower - 17th Floor, 11372 Riyadh, Saudi Arabia.

United Arab Emirates: Deutsche Bank AG in the Dubai International Financial Centre (registered no. 00045) is regulated by the Dubai Financial Services Authority. Deutsche Bank AG - DIFC Branch may only undertake the financial services activities that fall within the scope of its existing DFSA license. Principal place of business in the DIFC: Dubai International Financial Centre, The Gate Village, Building 5, PO Box 504902, Dubai, U.A.E. This information has been distributed by Deutsche Bank AG. Related financial products or services are only available to Professional Clients, as

defined by the Dubai Financial Services Authority.

Australia: Retail clients should obtain a copy of a Product Disclosure Statement (PDS) relating to any financial product referred to in this report and consider the PDS before making any decision about whether to acquire the product. Please refer to Australian specific research disclosures and related information at https://australia.db.com/australia/content/research-information.html

Australia and New Zealand: This research, and any access to it, is intended only for "wholesale clients" within the meaning of the Australian Corporations Act and New Zealand Financial Advisors Act respectively.

Additional information relative to securities, other financial products or issuers discussed in this report is available upon request. This report may not be reproduced, distributed or published without Deutsche Bank's prior written consent.

David Folkerts-Landau
Group Chief Economist and Global Head of Research

Raj Hindocha
Global Chief Operating Officer
Research

Michael Spencer
Head of APAC Research
Global Head of Economics

Steve Pollard
Head of Americas Research
Global Head of Equity Research

Anthony Klarman
Global Head of
Debt Research

Paul Reynolds
Head of EMEA
Equity Research

Dave Clark
Head of APAC
Equity Research

Pam Finelli
Global Head of
Equity Derivatives Research

Andreas Neubauer
Head of Research - Germany

Stuart Kirk
Head of Thematic Research

International Locations

Deutsche Bank AG
Deutsche Bank Place
Level 16
Corner of Hunter & Phillip Streets
Sydney, NSW 2000
Australia
Tel: (61) 2 8258 1234

Deutsche Bank AG
Große Gallusstraße 10-14
60272 Frankfurt am Main
Germany
Tel: (49) 69 910 00

Deutsche Bank AG
Filiale Hongkong
International Commerce Centre,
1 Austin Road West,Kowloon,
Hong Kong
Tel: (852) 2203 8888

Deutsche Securities Inc.
2-11-1 Nagatacho
Sanno Park Tower
Chiyoda-ku, Tokyo 100-6171
Japan
Tel: (81) 3 5156 6770

Deutsche Bank AG London
1 Great Winchester Street
London EC2N 2EQ
United Kingdom
Tel: (44) 20 7545 8000

Deutsche Bank Securities Inc.
60 Wall Street
New York, NY 10005
United States of America
Tel: (1) 212 250 2500