HACKATHON
Asset management
2024

Applications of AI and Machine Learning in Portfolio Management

McGill-FIAM Asset Management
Hackathon Instructions

McGill | DESAUTELS FIAM STATION FINTECH MONTRÉAL

Introductory note: Russ Goyenko (McGill University)

September 10, 2024

Welcome to the inaugural Asset Management Hackathon! This annual event is uniquely designed to bridge the gap between Finance and Data Science through a cross-disciplinary student case competition – the hackathon. Unlike other hackathons, our focus is exclusively on asset management, providing a distinct experience.

The hackathon is divided into two stages. Below, you'll find details about the first challenge, which involves creating portfolio trading strategies using big data and machine learning (ML). We recognize that applying data science to finance problems can be complex, so we have provided supplementary materials covering the basic common practices of applying ML& big data for stock picking. We encourage you to thoroughly review these materials before proceeding. You may or may not use these materials to solve the first challenge. You are encouraged to innovate further on the methodologies and their applications to portfolio management. You are also not limited to the data we provide here. You are free to supplement this data with extra data you think can be useful. We, however, highly recommend using this data as the base since it is one of our judgement criteria for selecting finalists.

While the primary objective of this stage is general training, the ideas, coding, methodologies, and techniques you develop will be crucial for the second stage—the final hack. Based on your submissions, we will select 10 teams to advance to the finals. In the second stage, you will receive further training, access to industry advisors and mentors who apply technology to asset management daily, insights from leaders in the field, and additional financial data science training.

This initial challenge is designed to be as realistic as the final hack, providing a solid foundation for your future progress. Even if you don't advance to the finals, this stage will give you valuable exposure to the foundational applications of ML in asset management. It's a crucial starting point for gaining hands-on experience in applying AI/ML to asset management research. We also recommend inviting one of your professors to serve as an academic advisor to support your journey.

Looking ahead, the objective of the final hack is to bring your research experience that you gain in the first stage, as close as possible to real-life implementation. Note, we don't expect you to deliver AI-transformative solutions for portfolio management in such a short time. The leading financial institutions like Citadel, JP Morgan, BlackRock, and Blackstone have been heavily investing in the development of AI-driven portfolio management models for several years. Despite their vast resources, the new generation of AI portfolio management models is still in development.

However, that is what they say: *“**ChatGPT potentially could create stock portfolios**, or hasten the production of analyst presentations. **There’s even an exchange-traded fund planned around the concept.**”* *Source: Bloomberg, March 07, 2023*

What we expect is just for you to do your best.
Good Luck!

First Challenge Description and Supplementary Training Materials

Written by Russ Goyenko (McGill) and Chengyu Zhang (Shanghai Jiao Tong University) in close collaboration with Raphael Duguay (Yale), and Evan Jo (Queen's)

Introduction

The recent surge in financial data related to different stock characteristics gives investors new information to consider when building their portfolios. Recent studies have shown that investors who use big data and advanced machine learning (ML) techniques can achieve better results. The idea is straightforward: more data means better information, leading to smarter investment choices.

Traditional financial models can't handle big data, but ML is well-suited for this task. ML methods excel at processing large amounts of data, identifying the most important factors, and reducing the complexity of the information.

However, not all ML methods work well with financial data, which often has many variables but not a lot of clear signals. The goal of this challenge is to:

1. Find machine learning (ML) methods that work well for building portfolios.
2. Identify key financial factors that help pick better-performing stocks.
3. Choose stocks and decide how much to invest in each one to create an optimized portfolio and predict its future success.
4. Run a back-test—this means checking how your portfolio would have performed in the past based on your stock picks.

The coding language of this competition is Python

To help you get started, we'll give you a toolkit that includes basic Python codes, guidelines on the best ways to apply and train ML algorithms for predicting stock returns, and basic long-short portfolio trading strategies (with Python code) to evaluate their past performance. A key point—your portfolio tests must be done Out-of-Sample. This means you can't let the algorithm "see" the future data it's supposed to predict. We'll explain more about this below.

The discussion, data and codes hereafter are borrowing heavily from a research paper by Russ Goyenko and Chengyu Zhang, *The Joint Cross Section of Options and Stock Returns Predictability with Big Data and Machine Learning*, McGill University working paper[1].

Data

Your task involves trading stocks from U.S. companies, specifically common shares of large companies (those above the NYSE median market size, see the file: hackathon_sample_v2.csv). The data covers the period from January 2000 to December 2023, with monthly updates. We're using monthly data instead of daily or intraday data to align with typical institutional asset management practices. Each month, you'll

[1] Available on ssrn: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3747238

have about 1,000 stocks to choose from, and your portfolio should include ***between 50 and 100 stocks (long and short positions combined)***. Your goal is to select stocks for your investment strategy using big data and machine learning (ML). Note, your portfolio should not exceed 100 stock names and be at least 50 names at all times.

The hardest part of working with data is often gathering it, but we've already done that for you (the main data to use is hackathon_sample_v2.csv). For each month and each stock, you have access to 147 firm-specific characteristics or signals (their names are already imputed into a csv file which we use in our python codes: factor_char_list.csv).[2] These signals represent around 40 years of research and industry practices on what factors are good predictors of stock returns. The characteristics cover a wide range of data, including financial fundamentals, past stock performance, price trends, liquidity, trading costs, and credit risks. You can find their definitions and explanations in Appendix A of this document, along with references to the research papers that introduced them. Most of these terms should be familiar if you've taken introductory Finance or Investment courses. If you're unsure about any factors, ChatGPT "has the knowledge" to explain them.

Here is an example on how to prompt ChatGPT about signals:
You: What Bello and Lin 2011 tell about inventory growth for future stock return predictability
ChatGPT:

Bello and Lin (2011) explore the relationship between inventory growth and future stock returns. Their research suggests that inventory growth can be a predictor of future stock returns, particularly in the context of firms' operational efficiency and market expectations.

Key findings from Bello and Lin (2011) include:

1. **Negative Relationship**: They find a generally negative relationship between inventory growth and future stock returns. Firms with higher inventory growth tend to experience lower future stock returns. This is because rising inventories might indicate declining sales, inefficient operations, or potential overproduction, which are often viewed negatively by the market.
2. **Economic Interpretation**: The negative relationship is consistent with the idea that inventory growth signals declining demand or inefficiencies in production and inventory management, which the market tends to penalize. Conversely, firms that manage their inventories effectively or have inventory reductions often signal improving demand or operational efficiency, leading to higher future returns.
3. **Cross-Sectional Analysis**: The study's analysis is cross-sectional, meaning it examines how inventory growth impacts stock returns across different firms. The results suggest that inventory growth can be a useful factor in predicting cross-sectional differences in future stock returns.

Overall, Bello and Lin (2011) argue that inventory growth is a valuable indicator that investors can use to gauge future performance, particularly in assessing a firm's operational health and efficiency.

[2] This data set is originally constructed by Jensen, Kelly, and Pedersen (2022). We used Bryan Kelly's github material https://github.com/bkelly-lab/ReplicationCrisis to compute most of characteristics

Note that all predictors are lagged by one month from time t, while all returns, the predicted variable (stock_exret), are from time t+1. Thus, this is a truly predictive exercise. Note, stock_exret is monthly access return over risk free rate. This is different from gross returns which also include risk free rate.

What do you want to predict?

The main goal for our main example is to predict the next month's stock returns. If you can make accurate predictions, you'll want to buy stocks that are expected to have the highest returns, like the top 50 stocks. You can also hedge by short-selling the 50 stocks you predict will have the lowest or most negative returns. You'll make these trades at the beginning of the month and check the results at the end of the month. If your predictions were accurate, you should see positive profits, measured by portfolio Alpha or simply the monthly return on this strategy. In other words, you can attempt to maximize either risk-adjusted returns or simple returns. The Python code (portfolio_analysis_hackathon.py) for this strategy is included in the toolkit we've provided, along with comments to help you understand it.

Predicting stock returns is challenging. Another approach is to predict company fundamentals that could lead to higher returns. For example, when a company releases better-than-expected earnings, its stock price might jump. At the time of writing, Palo Alto Networks' stock rose nearly 10% after they announced strong Q2 2024 earnings and gave a positive outlook for Q3.[3] You could try predicting things like next quarter's earnings, or even better, earnings surprises instead. We've provided next quarter, denoted Q1, analyst forecasts and the actual earnings released for the next quarter.[4] The difference between the two is the earnings surprise. If you correctly predict a positive earnings surprise, there's a good chance the stock price will rise. You can find more discussion on predicting earnings surprises in Chapados et al (2023).

As a company may be covered by multiple analysts, in the data file (hackathon_sample_v2.csv), the consensus of analyst forecasts for the earnings-per-share next quarter are available in the column eps_meanest (mean of analyst forecasts) or eps_medest (median of analyst forecasts). eps_stdevest provides the standard deviation of analyst forecasts, which tells you how disperse the forecasts are. If you choose to work with actual earnings-per-share reported in the next quarter, they are available in the column eps_actual. Note that both analyst forecasts and actual earnings are already from the future (just like stock returns, stock_exret), you can directly use them as left-hand side variables without any adjustments.

Of course, predicting stock performance is part science and part art. While a positive earnings surprise usually means the stock won't lose value in the long term, there are no guarantees.[5] You can also choose to predict other fundamentals like Price to Sales, Price to Earnings, or other ratios. Just make sure your predictions are forward-looking (for time $t+1$) while your other variables are based on past data (time t). What counts the most here is your idea, or your investment thesis: which fundamental(s) you think will

[3] Source: https://www.marketwatch.com/articles/palo-alto-earnings-stock-price-b1e38c83?mod=mw_quote_news
[4] Note, Q1 abbreviation stands for 1 quarter from now, not Q1 of a calendar year
[5] As it happened with NVIDIA Q2/2024 earnings announcement, where the revenues topped the expectations but Q3 outlook looked less aggressive than before.

drive the most of your strategy's performance.

If your university has access to WRDS (https://wrds-www.wharton.upenn.edu/), you can have access to other data to merge with those provided (in case you use extra data – please include in your submission the data cleaning code, along with a detailed description of the source data). For example, van Binsbergen, Han, and Lopez-Lira (2023) use sixty-seven financial ratios, such as the book-to-market ratio and dividend yields, obtained from the Financial Ratios Suite by Wharton Research Data Services. The authors also describe how they predict EPS (earnings-per-share). You can merge the data by the following stocks' identifiers: *permno* or *cusip*, and date (common practice is to use the year-month pair). The stock's *permno* or *cusip* are unique firm identifier in the data, you just need to control for the date to have them correctly aligned in calendar time. Aside from key identifiers, dates, company information, and stock characteristics, there are other data which were used during the data construction process. You can safely ignore them if they are not specifically mentioned in this note.

Even without using other extra data, the data provided already includes investment strategies that aim to perform well against the market portfolio. This means the data has useful signals and strategies to outperform S&P500, but the exact results you can achieve are unknown. Your task is to develop the best possible investment strategy, with no restrictions on turnover, leverage, or trading costs.

Choice of ML algorithms & Training

New machine learning (ML) algorithms are being created almost every day. There are anywhere from a dozen to over a hundred algorithms available, depending on how they are improved. The choice of which algorithm to use is up to you. This is where your creativity comes in: based on your investment idea, which technology best supports your idea with data?

In finance, researchers often prefer supervised or semi-supervised machine learning, where we guide the machine on what to learn. For simplicity, we'll explain the most common approaches using basic linear ML methods. The discussions and code we'll provide are from Goyenko and Zhang (2022), following the foundational work by Gu et al. (2020).

As an example we implement the following machine learning methods: LASSO of Tibshirani (1996), Elastic Net (EN) of Zou and Hastie (2005), Ridge of Hoerl and Kennard (1970).

We analyze the predictive power of machine learning algorithms for stock returns We therefore define a return on an asset in the most general form as:

$$\mathrm{r}_{\mathrm{i,t+1}} = \mathrm{E}(\mathrm{r}_{\mathrm{i,t+1}}) + \epsilon_{\mathrm{i,t+1}} \tag{1}$$

where:

$$E\left(r_{i,t+1}\right) = g^{*}(z_{i,t}) \tag{2}$$

Stocks are indexed as $i = 1, \ldots, N_t$, and months as $t = 1, \ldots, T$. Function $g^{*}(\cdot)$ represents the machine learning algorithm. It maintains the same form over time and across different assets, and leverages information from the entire panel. $z_{i,t}$ is what you feed the machine learning algorithm. In mathematical language, $z_{i,t}$ is a P dimensional vector of predictors (147 stock characteristics every month *t*). Equation (2) says that your machine learning algorithm attempts to predict the return of each stock for the next month, using information from the current month.

The most common approach in machine learning literature is to "tune" hyperparameters adaptively using the data from the validation sample. Hyperparameters include the penalization parameters in lasso, ridge and elastic net. Tuning parameters are estimated from the validation sample taking into account estimated model coefficients, where the coefficients are estimated from the training data alone. The third, the testing sub-sample, is used for neither estimation nor tuning, and is truly out of sample evaluation of model's predictive performances. For further details about tuning please refer to Goyenko and Zhang (2022) as well as to the python code provided (penalized_linear_hackathon.py) where we specify the grid for finetuning parameters.

Performance evaluation

Following Gu et al (2020), we first evaluate the statistical performance of predictability by calculating the out-of-sample, *OOS*, R^2 as

$$R^2_{OOS} = 1 - \frac{\sum_{(i,t)\in\mathcal{T}_3}(r_{i,t+1} - \hat{r}_{i,t+1})^2}{\sum_{(i,t)\in\mathcal{T}_3} {r_{i,t+1}}^2}$$

where $\mathcal{T}_3$ refers to the "test" periods, $r_{i,t+1}$ refers to the realized return of stock *i* at time *t+1*, and $\hat{r}_{i,t+1}$ refers to the predicted return of the same stock during the same period. Note that we don't subtract the historical mean in the denominator like you would normally see for out-of-sample R^2. The implication here is that the benchmark we compare our predictions against is not the historical mean of the stock returns, but rather zero, which means there is no predictability for stock returns at all. This is because relying on the historical mean to construct a portfolio actually delivers worse performance than simply randomly selecting stocks in the market. It also means that, if you achieve a positive out-of-sample R^2, no matter how small it is, you are capturing some predictability in the market. In fact, the out-of-sample R^2 of stock returns are not impressive, usually ranging between 1% and 2%, even with more complex models such as neural networks. The implementation of the above formula can be found near the end of the file penalized_linear_hackathon.py. **In your own analysis and in the submission deck, please do provide *OOS* R^2 of the methodology you are using.**

Training Procedures

We regularly update the model parameters using an expanding window approach. This allows the model to consider more recent data in order to make better predictions. More specifically, we first split our sample (01/2000 to 12/2023) into first 8 years of training sample, 01/2000 to 12/2007, $\mathcal{T}_1$, and two years of validation sample, 01/2008 to 12/2009, $\mathcal{T}_2$, with our first out of sample prediction, $\mathcal{T}_3$, for 01/2010 to 12/2010. We then expand the training sample by one year (01/2000 to 12/2008), roll the validation sample by one year (01/2009 to 12/2010), and produce the forecast for the next out of sample year (01/2011 to 12/2011), and so on, until we reach the end of the sample period. In the end, we should have 14 years of **monthly** out-of-sample predictions (from 2010 to 2023). While we refit/retrain (*or simply update with* *the new information*) the model every year to produce new out-of-sample predictions, the predictions themselves are monthly, e.g. we expect you to predict the stocks returns for each month in 2010, and so on. The exact implementation of the expanding window exercise can be found in the code (penalized_linear_hackathon.py, around lines 65 – 85).

Trading Strategy Portfolio Evaluations

While the statistical performance provides some preliminary ideas about the accuracy of the predictions, investors care more about the economical benefits that the model provides. That is, whether we can use the predictions to construct a portfolio that generates superior returns. A simple long-short strategy can be built by first sorting the stocks based on the predicted returns. At the beginning of every month (time *t*), we rank the stocks from low to high by their predicted returns. We then equally divide the stocks into ten buckets/portfolios, where the first portfolio contains stocks we predict to perform the worst, and the 10^{th} portfolio contains the stocks we predict to perform the best. We would buy the best/top portfolio and short-sell the worst/first portfolio, assuming our holding of each stock is of the same dollar amount (equal weights), and check the performance of this strategy at the end of the month (time *t+1*). Note that this strategy also has zero cost, where the long positions are fully financed by the money we get from the short positions.[6] We repeat this procedure for each month in the out-of-sample period. An alternative strategy would be to simply buy 50 stocks for which we have the highest predicted returns, and short sell 50 stocks for which we have the lowest predicted returns. The implementation can be found in portfolio_analysis_hackathon.py, starting from line 13.

Now we have the historical performance of our portfolio strategy during the out-of-sample period. Aside from raw portfolio returns, we also care about several other performance metrices, such as Sharpe ratio, portfolio alpha, market beta, information ratio, maximum one-month loss, maximum drawdown, and turnover. The definition and formulas of these measure can be found in Goyenko and Zhang (2022), and the implementation is also provided in portfolio_analysis_hackathon.py.

Estimation of portfolio Alpha and Beta

We define the alpha as the intercept from a simple linear regression where you regress your portfolio excess over risk free rate returns on the excess over risk free rate returns of S&P500. That is

$$R_{p,t} - r_{f,t} = \alpha + \beta(R_{SP500,t} - r_{f,t}) + \epsilon_t$$

where $R_{p,t}$ is your strategy portfolio monthly returns, r_f is risk free rate, and $R_{SP500,t}$ is monthly returns on S&P500 index. Since the returns you predict are pre-adjusted by subtracting the risk rate (exret stands for excess return), you can directly use your portfolio return as the left-hand side. The data file mkt_ind.csv contains the monthly data for risk free rate and S&P500 returns.

The intercept is Alpha or risk adjusted return, and the slope coefficient is the beta. These concepts are normally covered in the Introductory finance classes so we can skip further elaborations. When you want to annualize Alpha (as an output from the regression it is monthly), you can multiply it by 12.

[6] This is of course only a theoretical exercise as in reality you have to post capital on the margin account, assume borrowing rates, be exposed to margin calls and etc.

Discussion

It's important to remember that we're providing you with a basic toolkit, and you don't have to stick to the long-short strategy. For example, you can choose to have only a long position by selecting 50 to 100 stocks each month based on your predictions. If you'd like, you can also opt for only a short position, though it's generally harder to perform well using just shorts during this sample period. You can also mix strategies, going long in some months and using a long-short approach in others. Just make sure to explain what drives your decision to "switch" strategies and how you predict whether the next month should be long-only or long-short.

Whatever strategy you choose, it's crucial that you avoid using forward-looking information. In other words, you can't use actual events from month t+1 to make decisions in month t. That's why, along with your final submission decks, we're asking you to submit your Python codes. *We'll be checking to make sure there's no misuse of forward-looking information in your strategy*.

It is also important to recognize that we give you a start-up toolkit for ML methodology and how we apply it to finance data, and the codes with further details. Purely replicating our codes for the final submission decks will not be sufficient. You can get started with them, see what kind of performance you obtain with pure linear algorithms, and then build on it with your improvements.

Trading Criteria

You should always be invested in min 50 stocks or max 100 stocks (between 50 to 100 holdings).

You have no trading restrictions except rebalancing your portfolio. You can rebalance once per month, once per quarter or half a year. In the toolkit and examples provided we rebalance every month and report monthly portfolio turnover. The objective here is an active portfolio management strategy. At the very least you have to rebalance some (not all) of your holdings once every half-year.

Reports

Designing an investment strategy is the holy grail of the quantitative asset management industry. Most initial attempts either fail or prove to be non-tradable in real life due to market frictions. These frictions will prevent you from achieving high performance. For example, if you have to rebalance 100% of your portfolios every single month, most of your positive performance (also known as Alpha) will be erased by trading costs.

However, it does not matter at this stage, as it is a constant struggle for all quantitative investment professionals. What matters is to implement an idea and discuss the idea, and what you learned from it and what potential improvements can be done.

Therefore, we want you to give it your best effort and present your most promising idea, and its execution with the data we provide. Your creativity in setting up the training and the methodology, and how you use the data will be important criteria for evaluation.

Guidelines for the Deck

A rule of thumb, a deck, or 5 min pitch normally should not exceed 5 pages. Please use power point slide format.

Here is what to include:

Page 1: Executive Summary.

Summarize your strategy, ML algorithm(s) chosen and portfolio performance vs S&P500 (the returns of S&P500 for out-of-sample, OOS, testing period are included in toolkit and are also used within portfolio_analysis_hackathon.py code)

Page 2: Describe your investment strategy: Long, Long-Short, or mixed. What predictive signals you use to form the strategy. Present your top 10 holdings on average over OOS testing period, 01/2010 to 12/2023. Plot cumulative performance returns of your trading strategy vs S&P500 for OOS testing period, 01/2010 to 12/2023

Page 3. Data and Methodology: describe the data and methodology. Try to justify why you chose a specific ML approach. How do you structure your training? If there is any new model architecture or training approach you introduce – please describe it here. If you chose to supplement our data with extra data (it is optional but we would be happy to see it) – please describe the data, whether they turned out to be valuable signals or not. Also here is the place to present *OOS* R^2 statistics for the overall sample for ML algorithms you used.

Page 4. Portfolio Performance statistics for OOS testing period, 01/2010 to 12/2023 for your portfolio vs S&P 500

At the very least you must report the following portfolio performance statistics (their computation is provided in portfolio_analysis_hackathon.py) vs corresponding statistics of S&P500 for the same time period:

- Average annualized portfolio returns
- Annualized portfolio standard deviation
- Annualized Alpha (market risk-adjusted return, for your portfolio only)
- Sharpe Ratio (annualized)
- Information Ratio (annualized, for your portfolio only)[7]
- Maximum drawdown,
- Maximum one-month loss
- *Portfolio Turnover* (for your portfolio only)

[7] Sharpe and Information ratios are already annualized in the code provided, portfolio_analysis_hackathon.py. Th annualize standard deviation – you need to multiply it by $\sqrt{12}$. To annualize Alpha or average return, multiply it by 12.

Page 5. The discussion of your strategy. Did it perform the way you trained it and did it meet your expectations? What are the main fundamental signals contributing to the performance of your portfolio? What are the most profitable positions (stocks) that drove the performance, and why. What are the macro-economic events that contributed to the performance. Potential improvements that you could make to this strategy

Following the 5 pages deck, *you can attach an Appendix not exceeding another 5 pages*. You are free to put in Appendix anything you think will help us to evaluate your work better, or any details that you could not include in the main deck. Please feel free to use any visuals that can help the presentation.

Final Submission Package

1. Your deck with Appendix in one PDF file. Please use power point slide format and then convert it to pdf file.
2. Your Python codes – can be several files but clearly identify the main run file. PLEASE ZIP YOUR CODES IN ONE FOLDER AND UPLOAD AS ONE ZIPPED FILE. The submission website will not be able to accept individual python codes. Please use the following style guide for your codes: https://google.github.io/styleguide/pyguide.html and be explicit in your commenting.
3. Prepare CVs for each team member – it is a part of submission, and the final team members' registration.

Our Evaluation Criteria

We'll be evaluating your project based on two main things: your investment idea (which financial factors you're focusing on and predicting) and the machine learning tools you choose to bring this idea to life.

You can get creative with your idea. We've suggested selecting stocks based on return predictions, but you could also predict other factors like earnings surprises, price-earnings ratios, or price-sales ratios that might drive future returns. The more original your idea, the more we'll appreciate your effort.

You can also innovate with the technology. We've provided basic examples using simple linear ML algorithms, which are easy to understand but may not capture the full complexity of financial data. You can explore alternatives, like feed-forward neural networks or other deep learning methods. Trying out different technologies is part of your innovation and contribution.

You can also get creative with the data. While we provide you with quantitative data, you could use other types, like text data from corporate filings (10Q/10K reports), available here.

Lastly, we'll compare your portfolio's performance to the market (S&P 500). Since you don't have restrictions on trading or leverage, it's possible to beat the S&P 500 during the sample period. We'll consider your portfolio's performance, but the main focus will be on your originality, choice of technology, and overall execution of the back-test.

Remember, it's not just about having the best return—what matters is your unique idea, the tools you use, and how well you put it all together.

References

Chopados, N, Z. Fan, R. Goyenko, I. Laradji, F. Liu, and Zhang C., 2023, Can AI Read the Minds of Corporate Executives? McGill University working papper

Goyenko, R, and Zhang C. (2022). The Joint Cross Section of Options and Stock Returns Predictability with Big Data and Machine Learning, McGill University working paper

Gu, S., Kelly, B., and Xiu, D. (2020). Empirical asset pricing via machine learning. The Review of Financial Studies, 33(5):2223–2273.

Jensen, T. I., B. Kelly, and L. H. Pedersen. 2022. Is there a replication crisis in finance? Journal of Finance

Hoerl, A. E., and R. W. Kennard (1970): "Ridge regression: Biased estimation for nonorthogonal problems," Technometrics, 12(1), 55–67.

Tibshirani, R. (1996): "Regression shrinkage and selection via the lasso," Journal of the Royal Statistical Society: Series B (Methodological), 58(1), 267–288.

Zou, H., and T. Hastie (2005): "Regularization and variable selection via the elastic net," Journal of the Royal Statistical Society: Series B (statistical methodology), 67(2), 301–320.

Jules H van Binsbergen, Xiao Han, Alejandro Lopez-Lira, Man versus Machine Learning: The Term Structure of Earnings Expectations and Conditional Biases, *The Review of Financial Studies*, Volume 36, Issue 6, June 2023, Pages 2361–2396, https://doi.org/10.1093/rfs/hhac085

Appendix A.

A.1 Stocks-Specific Features

Table 1: Stock-specific Features

Feature	Acronym	Reference
Firm age	age	Jiang Lee and Zhang (2005)
Liquidity of book assets	aliq_at	Ortiz-Molina and Phillips (2014)
Liquidity of market assets	aliq_mat	Ortiz-Molina and Phillips (2014)
Amihud Measure	ami_126d	Amihud (2002)
Book leverage	at_be	Fama and French (1992)
Asset Growth	at_gr1	Cooper Gulen and Schill (2008)
Assets-to-market	at_me	Fama and French (1992)
Capital turnover	at_turnover	Haugen and Baker (1996)
Change in common equity	be_gr1a	Richardson et al. (2005)
Book-to-market equity	be_me	Rosenberg Reid and Lanstein (1985)
Market Beta	beta_60m	Fama and MacBeth (1973)
Dimson beta	beta_dimson_21d	Dimson (1979)
Frazzini-Pedersen market beta	betabab_1260d	Frazzini and Pedersen (2014)
Downside beta	betadown_252d	Ang Chen and Xing (2006)
Book-to-market enterprise value	bev_mev	Penman Richardson and Tuna (2007)
The high-low bid-ask spread	bidaskhl_21d	Corwin and Schultz (2012)
Abnormal corporate investment	capex_abn	Titman Wei and Xie (2004)
CAPEX growth (1 year)	capx_gr1	Xie (2001)
CAPEX growth (2 years)	capx_gr2	Anderson and Garcia-Feijoo (2006)
CAPEX growth (3 years)	capx_gr3	Anderson and Garcia-Feijoo (2006)
Cash-to-assets	cash_at	Palazzo (2012)
Net stock issues	chcsho_12m	Pontiff and Woodgate (2008)
Change in current operating assets	coa_gr1a	Richardson et al. (2005)
Change in current operating liabilities	col_gr1a	Richardson et al. (2005)
Cash-based operating profits-to-book assets	cop_at	
Cash-based operating profits-to-lagged book assets	cop_atl1	Ball et al. (2016)
Market correlation	corr_1260d	Assness, Frazzini, Gormsen, Pedersen (2020)
Coskewness	coskew_21d	Harvey and Siddique (2000)
Change in current operating working capital	cowc_gr1a	Richardson et al. (2005)
Net debt issuance	dbnetis_at	Bradshaw Richardson and Sloan (2006)
Growth in book debt (3 years)	debt_gr3	Lyandres Sun and Zhang (2008)
Debt-to-market	debt_me	Bhandari (1988)
Change gross margin minus change sales	dgp_dsale	Abarbanell and Bushee (1998)
Dividend yield	div12m_me	Litzenberger and Ramaswamy (1979)
Dollar trading volume	dolvol_126d	Brennan Chordia and Subrahmanyam (1998)
Coefficient of variation for dollar trading volume	dolvol_var_126d	Chordia Subrahmanyam and Anshuman (2001)
Change sales minus change Inventory	dsale_dinv	Abarbanell and Bushee (1998)
Change sales minus change receivables	dsale_drec	Abarbanell and Bushee (1998)
Change sales minus change SG&A	dsale_dsga	Abarbanell and Bushee (1998)
Earnings variability	earnings_variability	Francis et al. (2004)
Return on net operating assets	ebit_bev	Soliman (2008)
Profit margin	ebit_sale	Soliman (2008)
Ebitda-to-market enterprise value	ebitda_mev	Loughran and Wellman (2011)
Hiring rate	emp_gr1	Belo Lin and Bazdresch (2014)

Table 1 continued from previous page

Feature	Acronym	Reference
Equity duration	eq_dur	Dechow Sloan and Soliman (2004)
Net equity issuance	eqnetis_at	Bradshaw Richardson and Sloan (2006)
Equity net payout	eqnpo_12m	Daniel and Titman (2006)
Net payout yield	eqnpo_me	Boudoukh et al. (2007)
Payout yield	eqpo_me	Boudoukh et al. (2007)
Pitroski F-score	f_score	Piotroski (2000)
Free cash flow-to-price	fcf_me	Lakonishok Shleifer and Vishny (1994)
Change in financial liabilities	fnl_gr1a	Richardson et al. (2005)
Gross profits-to-assets	gp_at	Novy-Marx (2013)
Gross profits-to-lagged assets	gp_atl1	
Intrinsic value-to-market	intrinsic_value	Frankel and Lee (1998)
Inventory growth	inv_gr1	Belo and Lin (2011)
Inventory change	inv_gr1a	Thomas and Zhang (2002)
Idiosyncratic skewness from the CAPM	iskew_capm_21d	
Idiosyncratic skewness from the Fama-French 3-factor model	iskew_ff3_21d	Bali Engle and Murray (2016)
Idiosyncratic skewness from the q-factor model	iskew_hxz4_21d	
Idiosyncratic volatility from the CAPM (21 days)	ivol_capm_21d	
Idiosyncratic volatility from the CAPM (252 days)	ivol_capm_252d	Ali Hwang and Trombley (2003)
Idiosyncratic volatility from the Fama-French 3-factor model	ivol_ff3_21d	Ang et al. (2006)
Idiosyncratic volatility from the q-factor model	ivol_hxz4_21d	
Kaplan-Zingales index	kz_index	Lamont Polk and Saa-Requejo (2001)
Change in long-term net operating assets	lnoa_gr1a	Fairfield Whisenant and Yohn (2003)
Change in long-term investments	lti_gr1a	Richardson et al. (2005)
Market Equity	market_equity	Banz (1981)
Mispricing factor: Management	mispricing_mgmt	Stambaugh and Yuan (2016)
Mispricing factor: Performance	mispricing_perf	Stambaugh and Yuan (2016)
Change in noncurrent operating assets	ncoa_gr1a	Richardson et al. (2005)
Change in noncurrent operating liabilities	ncol_gr1a	Richardson et al. (2005)
Net debt-to-price	netdebt_me	Penman Richardson and Tuna (2007)
Net total issuance	netis_at	Bradshaw Richardson and Sloan (2006)
Change in net financial assets	nfna_gr1a	Richardson et al. (2005)
Earnings persistence	ni_ar1	Francis et al. (2004)
Return on equity	ni_be	Haugen and Baker (1996)
Number of consecutive quarters with earnings increases	ni_inc8q	Barth Elliott and Finn (1999)
Earnings volatility	ni_ivol	Francis et al. (2004)
Earnings-to-price	ni_me	Basu (1983)
Quarterly return on assets	niq_at	Balakrishnan Bartov and Faurel (2010)
Change in quarterly return on assets	niq_at_chg1	
Quarterly return on equity	niq_be	Hou Xue and Zhang (2015)
Change in quarterly return on equity	niq_be_chg1	
Standardized earnings surprise	niq_su	Foster Olsen and Shevlin (1984)
Change in net noncurrent operating assets	nncoa_gr1a	Richardson et al. (2005)
Net operating assets	noa_at	Hirshleifer et al. (2004)
Change in net operating assets	noa_gr1a	Hirshleifer et al. (2004)
Ohlson O-score	o_score	Dichev (1998)
Operating accruals	oaccruals_at	Sloan (1996)
Percent operating accruals	oaccruals_ni	Hafzalla Lundholm and Van Winkle (2011)
Operating cash flow to assets	ocf_at	Bouchard, Krüger, Landier and Thesmar (2019)
Change in operating cash flow to assets	ocf_at_chg1	Bouchard, Krüger, Landier and Thesmar (2019)

Table 1 continued from previous page

Feature	Acronym	Reference
Asset tangibility	tangibility	Hahn and Lee (2009)
Tax expense surprise	tax_gr1a	Thomas and Zhang (2011)
Share turnover	turnover_126d	Datar Naik and Radcliffe (1998)
Coefficient of variation for share turnover	turnover_var_126d	Chordia Subrahmanyam and Anshuman (2001)
Altman Z-score	z_score	Dichev (1998)
Number of zero trades with turnover as tiebreaker (6 months)	zero_trades_126d	Liu (2006)
Number of zero trades with turnover as tiebreaker (1 month)	zero_trades_21d	Liu (2006)
Number of zero trades with turnover as tiebreaker (12 months)	zero_trades_252d	Liu (2006)

* Subject to change without notice.