1. ON MACHINE LEARNING APPLICATIONS IN INVESTMENTS

Mike Chen, PhD
Head, Alternative Alpha Research, Robeco

Weili Zhou, CFA
Head, Quant Equity Research, Robeco

Introduction

In recent years, machine learning (ML) has been a popular technique in various domains, ranging from streaming video and online shopping recommendations to image detection and generation to autonomous driving. The attraction and desire to apply machine learning in finance are no different.

- "The global AI fintech market is predicted to grow at a CAGR of 25.3% between 2022 and 2027" (Columbus 2020).
- "A survey of IT executives in banking finds that 85% have a 'clear strategy' for adopting AI in developing new products and services" (Nadeem 2018).

Putting aside the common and widespread confusion between artificial intelligence (AI) and ML (see, e.g., Cao 2018; Nadeem 2018), the growth of ML in finance is projected to be much faster than that of the overall industry itself, as the previous quotes suggest. Faced with this outlook, practitioners may want answers to the following questions:

- What does ML bring to the table compared with traditional techniques?
- How do I make ML for finance work? Are there special considerations? What are some common pitfalls?
- What are some examples of ML applied to finance?

In this chapter, we explore how ML can be applied from a practitioner's perspective and attempt to answer many common questions, including the ones above.[1]

The first section of the chapter discusses practitioners' motivations for using ML, common challenges in implementing ML for finance, and solutions. The second section discusses several concrete examples of ML applications in finance and, in particular, equity investments.

Motivations, Challenges, and Solutions in Applying ML in Investments

In this section, we discuss reasons for applying ML, the unique challenges involved, and how to avoid common pitfalls in the process.

Motivations

The primary attraction of applying ML to equity investing, as with almost all investment-related endeavors, is the promise of higher risk-adjusted return. The hypothesis is that these techniques, explicitly designed for prediction tasks based on high-dimensional data and without any functional form specification, should excel at predicting future equity returns.

Emerging academic literature and collective practitioner experience support this hypothesis. In recent years, practitioners have successfully applied ML algorithms to predict equity returns, and ML-based return prediction algorithms have been making their way into quantitative investment models. These algorithms have been used worldwide in both developed and emerging markets, for large-cap and small-cap investment universes, and with single-country or multi-country strategies.[2] In general, practitioners have found that ML-derived alpha models outperform those generated from more traditional linear models[3] in predicting cross-sectional equity returns.

[1]Readers interested in the theoretical underpinnings of ML algorithms, such as random forest or neural networks, should read Hastie, Tibshirani, and Friedman (2009) and Goodfellow, Bengio, and Courville (2016).

[2]There are also numerous academic studies on using ML to predict returns. For example, ML techniques have been applied in a single-country setting by Gu, Kelly, and Xiu (2020) to the United States, by Abe and Nakayama (2018) to Japan, and by Leippold, Wang, and Zhou (2022) to China's A-share markets. Similarly, in a multi-country/regional setting, ML has been applied by Tobek and Hronec (2021) and Leung, Lohre, Mischlich, Shea, and Stroh (2021) to developed markets and by Hanauer and Kalsbach (2022) to emerging markets.

[3]For linear equity models, see, for example, Grinold and Kahn (1999).

© 2023 CFA Institute Research Foundation. All rights reserved.

In addition to predicting equity returns, ML has been used to predict intermediate metrics known to predict future returns. For example, practitioners have used ML to forecast corporate earnings and have found ML-derived forecasts to be significantly more accurate and informative than other commonly used earnings prediction models.[4] Another use of ML in equity investing developed by Robeco's quantitative researchers has been to predict not the entire investment universe's return but the returns of those equities that are likely to suffer a severe price drop in the near future. Investment teams at Robeco have found that ML techniques generate superior crash predictions compared with those from linear models using traditional metrics, such as leverage ratio or distance to default.[5]

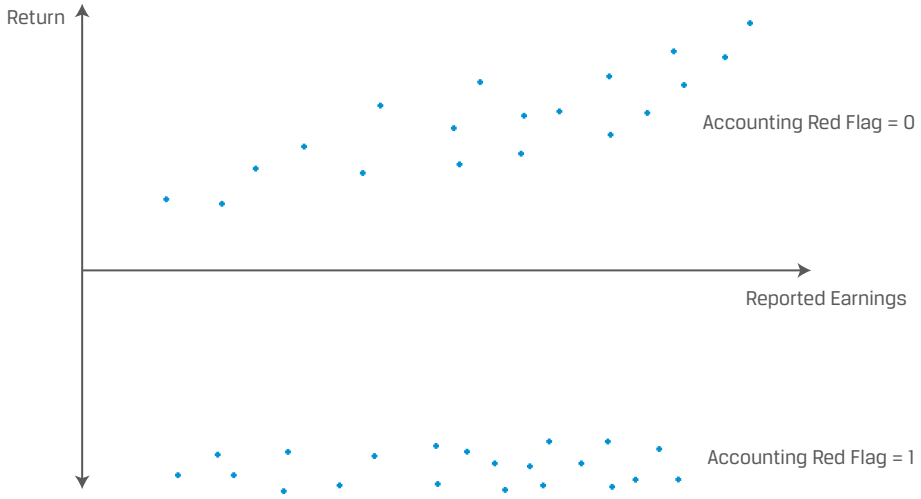
What drives the outperformance of ML over other known quantitative techniques? The main conclusion from practitioners and academics is that because ML algorithms do not prespecify the functional relationship between the prediction variables (equity return, future earnings, etc.) and the predictors (metrics from financial statements, past returns, etc.), ML algorithms are not constrained to a linear format as is typical of other techniques but, rather, can uncover interaction and nonlinear relationships between the input features and the output variable(s).

Interaction and nonlinear effects

The interaction effect occurs when the prediction outputs cannot be expressed as a linear combination of the individual inputs because the effect of one input depends on the value of the other ones. Consider a stylistic example of predicting equity price based on two input features: reported earnings and an accounting red flag, where the red-flag input is binary: 0 (no cause of concern) and 1 (grave concern). The resulting ML output with these two inputs may be that when the red-flag input is 0, the output is linearly and positively related to reported earnings; in contrast, when the red-flag input is 1, the output is a 50% decrease in price regardless of the reported earnings. **Exhibit 1** illustrates this stylistic example.

ML prediction can also outperform the traditional linear model prediction due to nonlinear effects. There are many empirically observed nonlinear effects that linear models cannot model. For example, there is a nonlinear relationship between a firm's credit default swap (CDS) spread and its equity returns because equity can be framed as an embedded call option on a firm's assets, thereby introducing nonlinearity.[6] **Exhibit 2** illustrates this example. Many ML algorithms, particularly neural networks,[7] explicitly

Exhibit 1. Illustration of the Interaction Effect between Accounting Red Flags and Equity Returns



Source: Robeco.

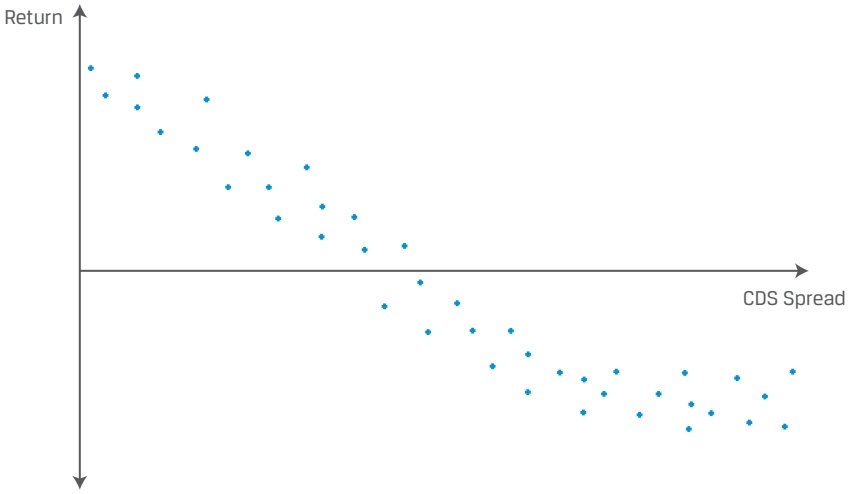
[4]This conclusion was replicated and supported also by academics. For example, see Cao and You (2021).

[5]For more information on using ML for crash prediction, see Swinkels and Hoogteijling (2022).

[6]For more details, see Birge and Zhang (2018).

[7]Neural networks incorporate nonlinearity through activation functions in each neuron. Without activation functions, neural networks, regardless of their depth, reduce down to traditional linear models commonly used in finance. For more on activation functions, see Goodfellow et al. (2016).

Exhibit 2. Illustration of the Nonlinear Relationship between a CDS and Equity Returns



Source: Robeco.

introduce nonlinearity into the model setup, facilitating nonlinearity modeling.

Empirically, academics and practitioners have found that the interaction effect accounts for a large portion of ML techniques' outperformance, while the jury is still out on whether nonlinear effects contribute positively to the outperformance. A well-known study in the field by Gu, Kelly, and Xiu (2020, p. 2258) found that "the favorable performance of [ML algorithms] indicates a benefit to allowing for potentially complex interactions among predictors." In the study, comparing the performance of purely linear models with that of generalized linear models, the authors note that "the generalized linear model … fails to improve on the performance of purely linear methods (R^2_{OOS} of 0.19%). The fact that this method uses spline functions of individual features, but includes no interaction among features, suggests that univariate expansions provide little incremental information beyond the [purely] linear model" (Gu et al. 2020, p. 2251). However, other studies have found that both interaction and nonlinear effects contribute positively to ML models' outperformance (see, e.g., Abe and Nakayama 2018; Swinkels and Hoogteijling 2022; Choi, Jiang, and Zhang 2022).

Find relationships from the data deluge

Another attraction of applying ML to financial markets is the promise of having the algorithm discover relationships not specified or perhaps not known by academics and practitioners—that is, *data mining*, which historically has been a pejorative in quantitative finance circles. Another term being used, perhaps with a more positive connotation, is "knowledge discovery."

With the ongoing information and computing revolution and the increased popularity of quantitative finance, the amount of financial data is growing at a rapid pace. This increased amount of data may or may not embody relevant information for investment purposes. Since many of the data types and sources are new, many investors do not have a strong prior opinion on whether and how they can be useful. Thus, ML algorithms that are designed to look for relationships have become attractive for practitioners and academics in the hope that, even without specifying a hypothesis on the economic relationship, the ML algorithm will figure out the link between inputs and outputs. Although ML algorithms have built-in mechanisms to combat overfitting or discovering spurious correlations between input features and output predictions,[8] caution must still be taken to avoid discovering nonrepeatable and nonreplicable relationships and patterns. Later in this chapter, we will address this issue further and consider other challenges practitioners face when implementing ML in live portfolios. But first, we will discuss what makes the financial market different from other domains in which ML has shown tremendous success.

Unique Challenges of Applying ML in Finance

When applying ML for investments, great care must be taken because financial markets differ from domains where

[8]Examples include *k*-fold cross-validation, dropout, and regularization. For deeper discussions, see Hastie et al. (2009) and Goodfellow et al. (2016).

ML has made tremendous strides. These differences can mitigate many of the specific advantages ML algorithms enjoy, making them less effective in practice when applied in real-life situations. A few of these differences follow.

Signal-to-noise ratio and system complexity

Financial data have a low signal-to-noise ratio. For a given security, any one metric is generally not a huge determinant of how that security will perform. For example, suppose Company XYZ announced great earnings this quarter. Its price can still go down after the announcement because earnings were below expectations, because central banks are hiking interest rates, or because investors are generally long the security and are looking to reduce their positions. Compare this situation with a high signal-to-noise domain—for example, streaming video recommendation systems. If a person watches many movies in a specific genre, chances are high that the person will also like other movies in that same genre. Because financial returns compress high-dimensional information and drivers (company-specific, macro, behavioral, market positioning, etc.) into one dimension, positive or negative, the signal-to-noise ratio of any particular information item is generally low.

It is fair to say that the financial market is one of the most complex man-made systems in the world. And this complexity and low signal-to-noise ratio can cause issues when ML algorithms are not applied skillfully. Although ML algorithms are adept at detecting complex relationships, the complexity of the financial market and the low signal-to-noise ratio that characterizes it can still pose a problem because they make the true relationship between drivers of security return and the outcome difficult to detect.

Small vs. big data

Another major challenge in applying ML in financial markets is the amount of available data. The amount of financial data is still relatively small compared with many domains in which ML has thrived, such as the consumer internet domain or the physical sciences. The data that quantitative investors traditionally have used are typically quarterly or monthly. And even for the United States, the market with the longest available reliable data, going back 100 years, the number of monthly data points for any security we might wish to consider is at most 1,200. Compared with other domains where the amount of data is in the billions and trillions, the quantity of financial data is minuscule. To be fair, some of the newer data sources, or "alternative data," such as social media posts or news articles, are much more abundant than traditional financial data. However, overall, the amount of financial data is still small compared with other domains.

The small amount of financial data is a challenge to ML applications because a significant driver of an ML algorithm's power is the amount of available data (see Halevy, Norvig, and Pereira 2009). Between a simple ML algorithm trained on a large set of data versus a sophisticated ML algorithm trained on a relatively smaller set of data, the simpler algorithm often outperforms in real-life testing. With a large set of data, investors applying ML can perform true cross-validation and out-of-sample testing to minimize overfitting by dividing input data into different segments. The investor can conduct proper hyperparameter tuning and robustness checks only if the amount of data is large enough. The small amount of financial data adds to the challenges of applying ML to financial markets mentioned earlier—the financial markets' high system complexity and low signal-to-noise ratio.

Stationarity vs. adaptive market, irrationality

Finally, what makes financial markets challenging for ML application in general is that markets are nonstationary. What we mean is that financial markets adapt and change over time. Many other domains where ML algorithms have shined are static systems. For example, the rules governing protein unfolding are likely to stay constant regardless of whether researchers understand them. In contrast, because of the promised rewards, financial markets "learn" as investors learn what works over time and change their behavior, thereby changing the behavior of the overall market. Furthermore, because academics have been successful over the last few decades in discovering and publishing drivers of market returns—for example, Fama and French (1993)—their research also increased knowledge of *all* market participants and such research changes market behavior, as noted by McLean and Pontiff (2016).

The adaptive nature of financial markets means not only that ML algorithms trained for investment do not have a long enough history with which to train the model and a low signal-to-noise ratio to contend with but also that the rules and dynamics that govern the outcome the models try to predict also change over time. Luckily, many ML algorithms are adaptive or can be designed to adapt to evolving systems. Still, the changing system calls into question the validity and applicability of historical data that can be used to train the algorithms—data series that were not long enough to begin with. To further complicate the issue, financial markets are man-made systems. Their results are the collective of individual human actions, and human beings often behave irrationally—for example, making decisions based on greed or fear.[9] This irrationality characteristic does not exist in many of the other domains in which ML has succeeded.

[9]There have been various studies on how greed and fear affect market participants' decision-making process. See, for example, Lo, Repin, and Steenbarger (2005).

How to Avoid Common Pitfalls When Applying ML in Finance

This section addresses some potential pitfalls when applying ML to financial investment.[10]

Overfitting

Because of the short sample data history, overfitting is a significant concern when applying ML techniques in finance. This concern is even stronger than when applying traditional quantitative techniques, such as linear regression, because of the high degrees of freedom inherent in ML algorithms. The result of overfitting is that one may get a fantastic backtest, but out-of-sample, the results will not live up to expectations.

There are some common techniques used in ML across all domains to combat overfitting. They include cross-validation, feature selection and removal, regularization, early stopping, ensembling, and having holdout data.[11] Because of the lower sample data availability in finance, some of these standard techniques might not be applicable or work as well in the financial domain as in others.

However, there are also advantages to working in the financial domain. The most significant advantage is human intuition and economic domain knowledge. What we mean by this is that investors and researchers applying machine learning can conduct "smell tests" to see whether the relationships found by ML algorithms between input features and output predictions make intuitive or economic sense. For example, to examine the relationship between inputs and outputs, one can look at Shapley additive explanation (SHAP) value, introduced by Lundberg and Lee (2017). SHAP value is computed from the average of the marginal contribution of the feature when predicting the targets, where the marginal contribution is computed by comparing the performance after withholding that variable from the feature set versus the feature set that includes the variable.

Exhibit 3 plots SHAP values from Robeco's work on using ML to predict equity crashes,[12] where various input features are used to predict the probability of financial distress of various stocks in the investment universe. The color of each dot indicates the sign and magnitude of a feature, where red signals a high feature value and blue denotes a low feature value. Take Feature 25, for example. As the feature value increases (as indicated by the color red), the ML algorithm predicts a higher probability of financial distress. And as the feature value decreases (as indicated by the color blue), the ML algorithm predicts a lower probability of financial distress. With this information, experienced investors can apply their domain knowledge to see whether the relationship discovered by the ML algorithm makes sense. For example, if Feature 25, in this case, is the leverage ratio and Feature 1 is distance to default,[13] then the relationship may make sense. However, if the features are flipped (Feature 25 is distance to default and Feature 1 is the leverage ratio), then it is likely that the ML algorithm made a mistake, possibly through overfitting.

Another approach to mitigate overfitting and having ML algorithms find spurious correlations is to try to eliminate it from the start. *Ex post* explanation via SHAP values and other techniques is useful, but investors can also apply their investment domain knowledge to curate the input set to select those inputs likely to have a relationship with the prediction objective. This is called "feature engineering" in ML lingo and requires financial domain knowledge. As an example, when we are trying to predict stock crash probability, fundamental financial metrics such as profit margin and debt coverage ratio are sensible input features for the ML algorithm, but the first letter of the last name of the CEO, for example, is likely not a sensible input feature.

Replicability

There is a debate about whether there is a replicability crisis in financial research.[14] The concerns about replicability are especially relevant to results derived from applying ML because, in addition to the usual reasons for replicability difficulties (differences in universe tested, *p*-hacking, etc.), replicating ML-derived results also faces the following challenges:

- The number of tunable variables in ML algorithms is even larger than in the more traditional statistical techniques.
- ML algorithms are readily available online. Investors can often download open-sourced algorithms from the internet without knowing the algorithm's specific version used in the original result, random seed, and so on. In addition, if one is not careful, different implementations of the same algorithm can have subtle differences that result in different outcomes.

To avoid replicability challenges, we suggest ML investors first spend time building up a robust data and code

[10]For additional readings, see, for example, López de Prado (2018); Arnott, Harvey, and Markowitz (2019); Leung et al. (2021).

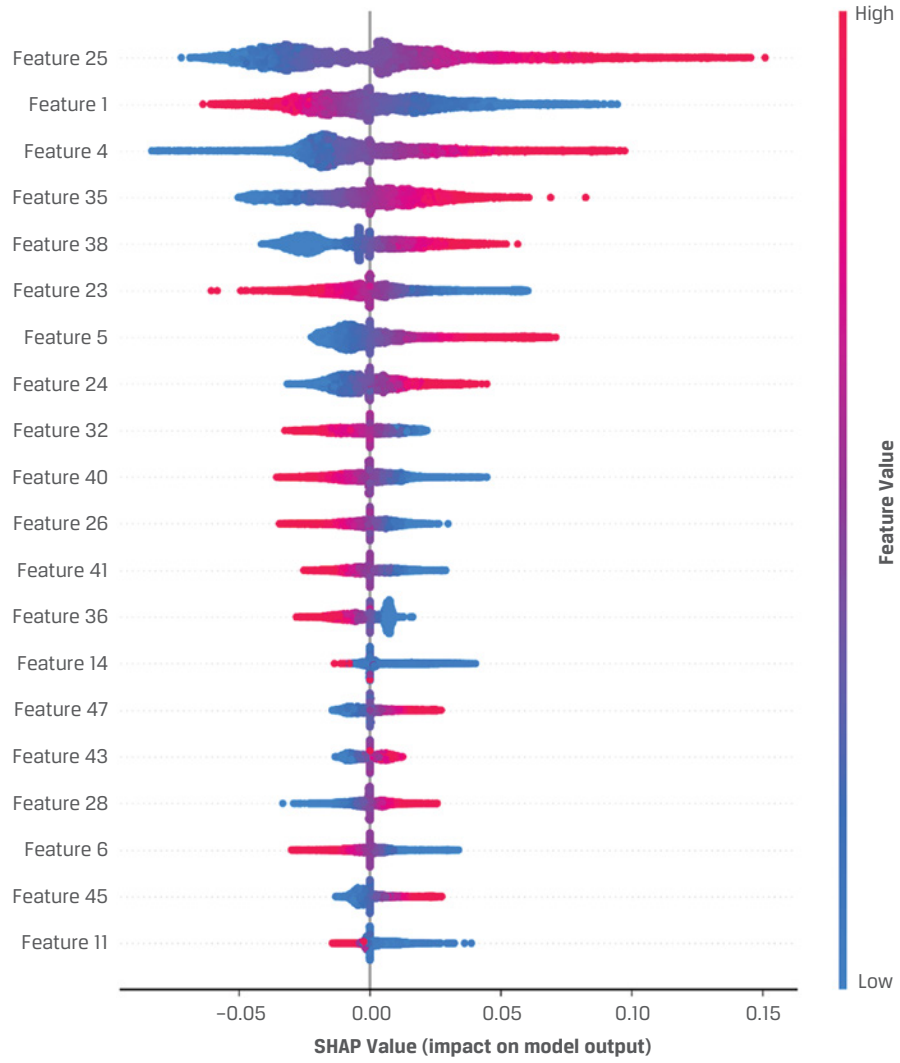
[11]See Hastie et al. (2009) and Goodfellow et al. (2016) for more discussion on these techniques.

[12]For more details, see Swinkels and Hoogteijling (2022).

[13]See Jessen and Lando (2013) for more discussion on distance to default.

[14]See Harvey, Liu, and Zhu (2016) and Hou, Xue, and Zhang (2020) for more discussion on the replicability crisis in finance.

Exhibit 3. SHAP Value between Input Features and ML Output Predicting Financial Distress



Source: Swinkels and Hoogteijling (2022).

infrastructure to conduct ML research and experiments. This includes, but is not limited to, the following:

- A robust system for code version control, a way to specify and fix all parameters used in the algorithm, including ML algorithm version number, investment universe tested, hyperparameters, feature sets used, and so on
- Documentation of all the tested iterations and hypotheses, including those that failed to show good results (in other words, showing the research graveyards)

Such documentation listed above may not be possible regarding publicly disclosed results, but it should at least be attempted when discussing and verifying results within the same organization.

Lookahead bias/data leakage

Lookahead bias is another commonly known issue for experienced quant investors that applies to ML. An example would be that if quarterly results are available only 40 days after the quarter end, the quarterly data should be used only when available historically[15] and not at the quarter end date.

[15]This is called "point-in-time" in the quant investment industry.

Other cases can be more subtle. For example, if one conducts research over a period that includes the tech bubble and its subsequent crash (2000–2002) and the investment universe tested does not include information technology (IT) firms, then it is incumbent upon the investor to provide a sensible reason why IT firms are not included.

Related to lookahead bias and a more general problem directly related to ML is the problem of data leakage. Data leakage occurs when data used in the training set contain information that can be used to infer the prediction, information that would otherwise not be available to the ML model in live production.

Because financial data occur in time series, they are often divided chronologically into training, validation, and testing sets. ML predictive modeling aims to predict outcomes the model has not seen before. One form of data leakage can occur if information that should be in one set ends up in another among the three sets of data (training, validation, and testing). When this occurs, the ML algorithm is evaluated on the data it has seen. In such cases, the results will be overly optimistic and true out-of-sample performance (that is, the live performance of the algorithm) will likely disappoint. This phenomenon is called "leakage in data."

Another type of data leakage is called "leakage in features." Leakage in features occurs when informative features about the prediction outcome are included but would otherwise be unavailable to ML models at production time, even if the information is not in the future. For example, suppose a company's key executive is experiencing serious health issues but the executive has not disclosed this fact to the general public. In that case, including that information in the ML feature set may generate strong backtest performance, but it would be an example of feature leakage.

Various techniques can be applied to minimize the possibility of data leakage. One of the most basic is to introduce a sufficient gap period between training and validation sets and between validation and testing. For example, instead of having validation sets begin immediately after the end of the training set, introduce a time gap of between a few months and a few quarters to ensure complete separation of data. To prevent data leakage, investors applying ML should think carefully about what is available and what is not during the model development and testing phases. In short, common sense still needs to prevail when applying ML algorithms.

Implementation gap

Another possible pitfall to watch out for when deploying ML algorithms in finance is the so-called implementation gap, defined as trading instruments in the backtest that are either impossible or infeasible to trade in live production. An example of an implementation gap is that the ML algorithm generates its outperformance in the backtest mainly from small- or micro-cap stocks. However, in live trading, either these small- or micro-cap stocks may be too costly to trade because of transaction costs or there might not be enough outstanding shares available to own in the scale that would make a difference to the strategy deploying the ML algorithm. As mentioned, implementation affects all quant strategies, not just those using ML algorithms. But ML algorithms tend to have high turnover, increasing trading cost associated with smaller market-cap securities. Similar to small- or micro-cap stocks, another example of an implementation gap is shorting securities in a long–short strategy. In practice, shorting stocks might be impossible or infeasible because of an insufficient quantity of a given stock to short or excessive short borrowing costs.[16]

Explainability and performance attribution

A main criticism of applying machine learning in finance is that the models are difficult to understand. According to Gu et al. (2020, p. 2258), "Machine learning models are often referred to as 'black boxes,' a term that is in some sense a misnomer, as the models are readily inspectable. However, they are complex, and this is the source of their power and opacity. Any exploration of the interaction effect is vexed by vast possibilities for identity and functional forms for interacting predictors."

Machine learning for other applications might not need to be fully explainable at all times; for example, if the ML algorithm suggests wrong video recommendations based on the viewers' past viewing history, the consequences are not catastrophic. However, with billions of investment dollars on the line, asset owners demand managers that use ML-based algorithms explain how the investment decisions are made and how performance can be attributed. In recent years, explainable machine learning has emerged in the financial domain as a focus topic for both practitioners and academics.[17]

The fundamental approach to recent explainable machine learning work is as follows:

1. For each input feature, f_i, in the ML algorithm, fix its value as x. Combine this fixed value for f_i with all other sample data while replacing feature f_i with the value x. Obtain the prediction output.
2. Average the resulting predictions. This is the partial prediction at point x.

[16]For additional readings, see, for example, Avramov, Chordia, and Goyal (2006); López de Prado (2018); Hou et al. (2020); Avramov, Cheng, and Metzker (2022).

[17]In addition to the SHAP values discussed in Lundberg and Lee (2017), recent works in the area of explainable machine learning include Li, Turkington, and Yazdani (2020); Li, Simon, and Turkington (2022); and Daul, Jaisson, and Nagy (2022).

3. Now range the fixed value x over feature f_i's typical range to plot out the resulting function. This is called the "partial dependence response" to feature f_i.

This response plot, an example of which is illustrated in **Exhibit 4**, can then be decomposed into linear and nonlinear components.

Similarly, one can estimate the pairwise-interaction part of the ML algorithm, computed using joint partial prediction of features f_i and f_j, by subtracting the partial prediction of each feature independently. An example of the pairwise-interaction result is shown in **Exhibit 5**.

Exhibits 3–5 allow ML investors to understand how the input features affect the output prediction. To conduct performance attribution of an ML portfolio and decompose it into the various parts (linear, interaction, and nonlinear), one can extract the partial dependence responses and form portfolios from them. With this approach, one can get return attribution, as shown in **Exhibit 6**.

Sample ML Applications in Finance

We have seen the common pitfalls when applying ML to finance and the strategies to mitigate them. Let us now look at examples of ML applied to financial investing.[18]

Predicting Cross-Sectional Stock Returns

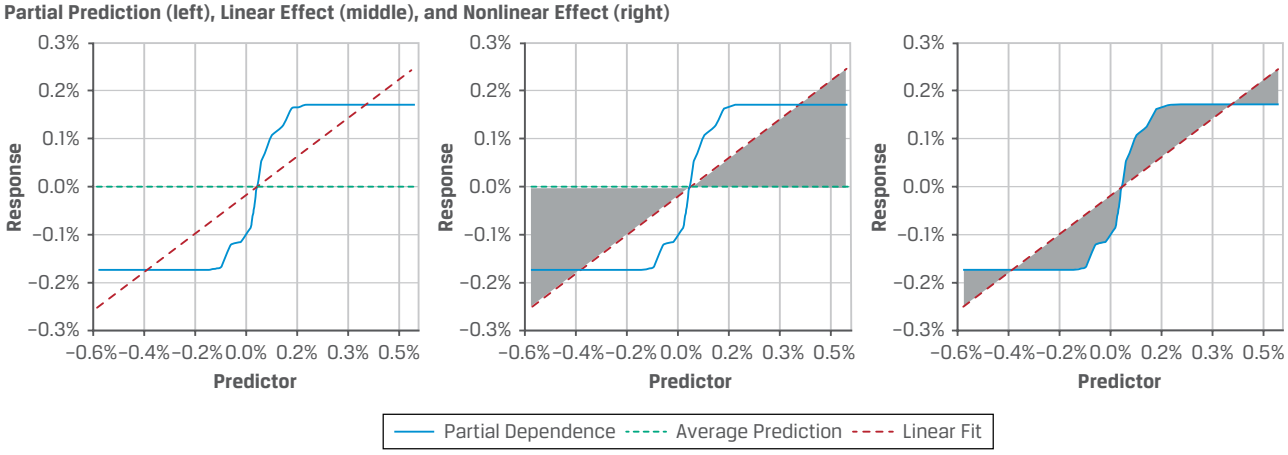
In this section, we discuss specifically using ML to predict cross-sectional stock returns.

Investment problem

Perhaps the most obvious application of ML to financial investments is to directly use ML to predict whether each security's price is expected to rise or fall and whether to buy or sell those securities. Numerous practitioners and academics have applied ML algorithms to this prediction task.[19]

The ML algorithms used in this problem are set up to compare cross-sectional stock returns. That is, we are interested in finding the relative returns of securities in our investment universe rather than their absolute returns.[20] The ML algorithms make stock selection decisions rather than country/industry timing and allocation decisions. Stock selection is an easier problem than the timing and allocation problem, because the algorithms have more data to work with.[21]

Exhibit 4. Linear and Nonlinear Decomposition of an ML Algorithm's Output to Input Feature f_i



Source: Li, Turkington, and Yazdani (2020).

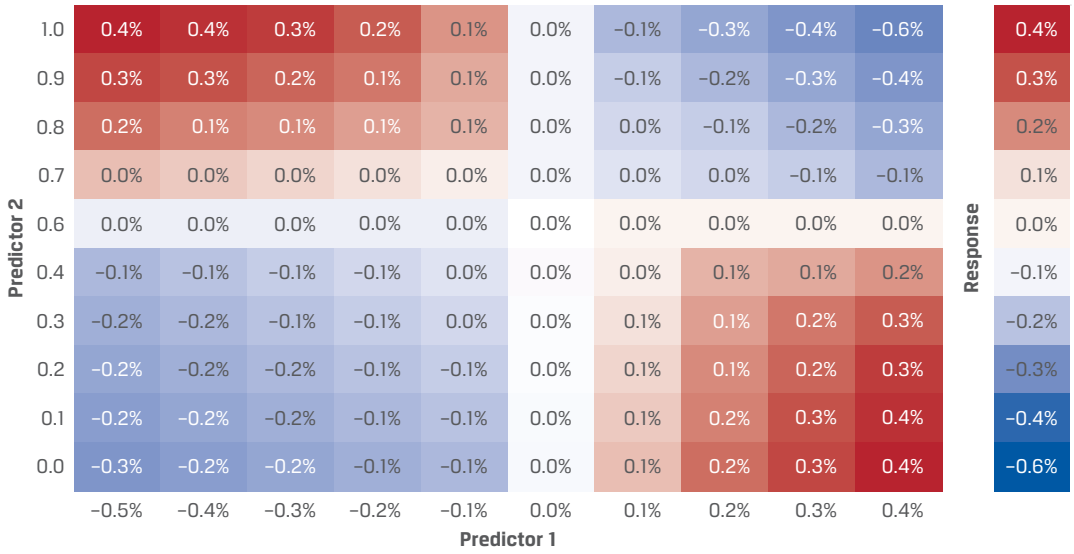
[18]For more general applications of ML to finance, see López de Prado (2019).

[19]This problem has been studied in numerous recent papers—for example, Abe and Nakayama (2018); Rasekhschaffe and Jones (2019); Gu et al. (2020); Choi, Jiang, and Zhang (2022); Hanauer and Kalsbach (2022).

[20]In addition to cross-sectional returns, Gu et al. (2020) also study the time-series problem.

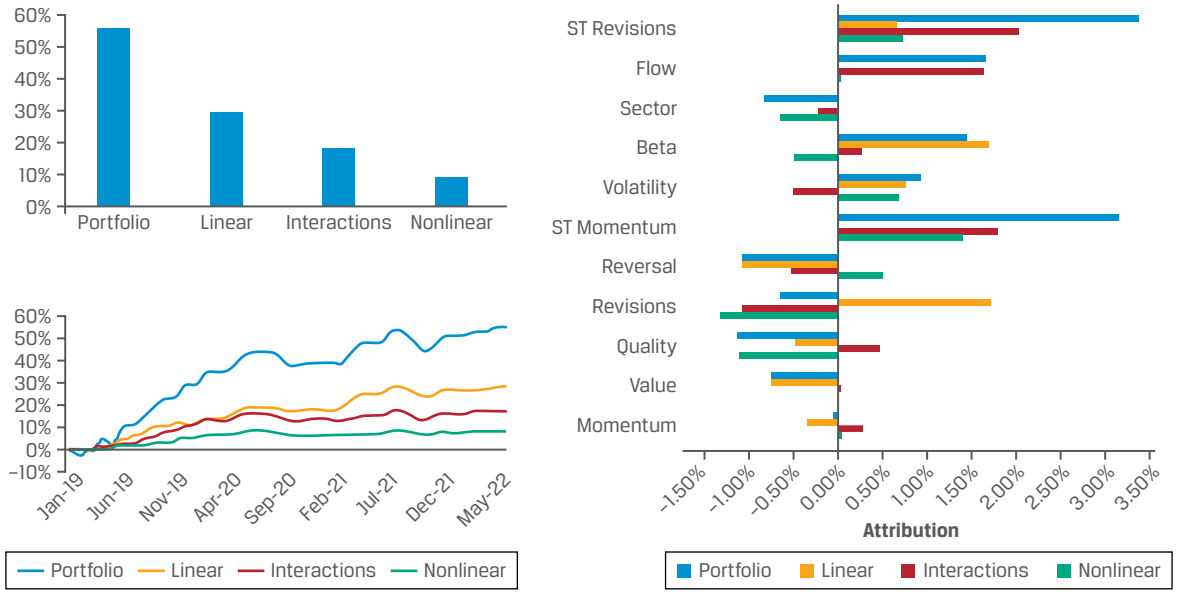
[21]However, when compared with other fields, the amount of data here is still miniscule, as noted before.

Exhibit 5. Example of the Pairwise-Interaction Effect



Source: Li, Simon, and Turkington (2022).

Exhibit 6. Example ML Portfolio Return Attribution



Source: Robeco.

Methodology

This problem is set up with the following three major components:

1. Investment universe: US, international, emerging market, and so on
2. ML algorithms: (boosted) trees/random forests, neural networks with *l* layers, and so on, and ensembles thereof
3. Feature set: typical financial metrics that are used in linear models, such as various price ratios (value), profitability (quality), and past returns (momentum)

Note that for Item 3, by using a very limited feature set, the ML investor is essentially applying her domain knowledge and imposing a structure on the ML algorithm to counter the limited data challenge discussed in the previous section.

Results

There are five consistent results that various practitioner and academic studies have found. First and foremost, there is a statistically significant and economically meaningful outperformance of ML algorithm prediction versus the traditional linear approach. For example, in forming a long–short decile spread portfolio from (four-layer) neural network–generated stock return prediction, a well-known study in equity return prediction (Gu et al. 2020) found that the strategy has an annualized out-of-sample Sharpe ratio of 1.35 under value weighting and 2.45 under equal weighting. For comparison, the same portfolio constructed from ordinary least squares (OLS) prediction with the same input features produced a Sharpe ratio of 0.61 and 0.83 for value weighting and equal weighting, respectively. **Exhibit 7** shows the value-weighting results.

Exhibit 7. Out-of-Sample Performance of Benchmark OLS Portfolio vs. Various ML Portfolios, Value Weighting

	OLS-3+H				PLS				PCR			
	Pred.	Avg.	Std. Dev.	Sharpe Ratio	Pred.	Avg.	Std. Dev.	Sharpe Ratio	Pred.	Avg.	Std. Dev.	Sharpe Ratio
Low (L)	−0.17	0.40	5.90	0.24	−0.83	0.29	5.31	0.19	−0.68	0.03	5.98	0.02
2	0.17	0.58	4.65	0.43	−0.21	0.55	4.96	0.38	−0.11	0.42	5.25	0.28
3	0.35	0.60	4.43	0.47	0.12	0.64	4.63	0.48	0.19	0.53	4.94	0.37
4	0.49	0.71	4.32	0.57	0.38	0.78	4.30	0.63	0.42	0.68	4.64	0.51
5	0.62	0.79	4.57	0.60	0.61	0.77	4.53	0.59	0.62	0.81	4.66	0.60
6	0.75	0.92	5.03	0.63	0.84	0.88	4.78	0.64	0.81	0.81	4.58	0.61
7	0.88	0.85	5.18	0.57	1.06	0.92	4.89	0.65	1.01	0.87	4.72	0.64
8	1.02	0.86	5.29	0.56	1.32	0.92	5.14	0.62	1.23	1.01	4.77	0.73
9	1.21	1.18	5.47	0.75	1.66	1.15	5.24	0.76	1.52	1.20	4.88	0.86
High (H)	1.51	1.34	5.88	0.79	2.25	1.30	5.85	0.77	2.02	1.25	5.60	0.77
H – L	1.67	0.94	5.33	0.61	3.09	1.02	4.88	0.72	2.70	1.22	4.82	0.88
	ENet+H				GLM+H				RF			
	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio
Low (L)	−0.04	0.24	5.44	0.15	−0.47	0.08	5.65	0.05	0.29	−0.09	6.00	−0.05
2	0.27	0.56	4.84	0.40	0.01	0.49	4.80	0.35	0.44	0.38	5.02	0.27
3	0.44	0.53	4.50	0.40	0.29	0.65	4.52	0.50	0.53	0.64	4.70	0.48
4	0.59	0.72	4.11	0.61	0.50	0.72	4.59	0.55	0.60	0.60	4.56	0.46
5	0.73	0.72	4.42	0.57	0.68	0.70	4.55	0.53	0.67	0.57	4.51	0.44
6	0.87	0.85	4.60	0.64	0.84	0.84	4.53	0.65	0.73	0.64	4.54	0.49
7	1.01	0.87	4.75	0.64	1.00	0.86	4.82	0.62	0.80	0.67	4.65	0.50
8	1.16	0.88	5.20	0.59	1.18	0.87	5.18	0.58	0.87	1.00	4.91	0.71
9	1.36	0.80	5.61	0.50	1.40	1.04	5.44	0.66	0.96	1.23	5.59	0.76
High (H)	1.66	0.84	6.76	0.43	1.81	1.14	6.33	0.62	1.12	1.53	7.27	0.73
H – L	1.70	0.60	5.37	0.39	2.27	1.06	4.79	0.76	0.83	1.62	5.75	0.98

(*continued*)

Exhibit 7. Out-of-Sample Performance of Benchmark OLS Portfolio vs. Various ML Portfolios, Value Weighting (*continued*)

	GBRT+H				NN1				NN2			
	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio
Low (L)	−0.45	0.18	5.60	0.11	−0.38	−0.29	7.02	−0.14	−0.23	−0.54	7.83	−0.24
2	−0.16	0.49	4.93	0.35	0.16	0.41	5.89	0.24	0.21	0.36	6.08	0.20
3	0.02	0.59	4.75	0.43	0.44	0.51	5.07	0.35	0.44	0.65	5.07	0.44
4	0.17	0.63	4.68	0.46	0.64	0.70	4.56	0.53	0.59	0.73	4.53	0.56
5	0.34	0.57	4.70	0.42	0.80	0.77	4.37	0.61	0.72	0.81	4.38	0.64
6	0.46	0.77	4.48	0.59	0.95	0.78	4.39	0.62	0.84	0.84	4.51	0.65
7	0.59	0.52	4.73	0.38	1.11	0.81	4.40	0.64	0.97	0.95	4.61	0.71
8	0.72	0.72	4.92	0.51	1.31	0.75	4.86	0.54	1.13	0.93	5.09	0.63
9	0.88	0.99	5.19	0.66	1.58	0.96	5.22	0.64	1.37	1.04	5.69	0.63
High (H)	1.11	1.17	5.88	0.69	2.19	1.52	6.79	0.77	1.99	1.38	6.98	0.69
H − L	1.56	0.99	4.22	0.81	2.57	1.81	5.34	1.17	2.22	1.92	5.75	1.16
	NN3				NN4				NN5			
	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio	Pred	Avg	Std. Dev.	Sharpe Ratio
Low (L)	−0.03	−0.43	7.73	−0.19	−0.12	−0.52	7.69	−0.23	−0.23	−0.51	7.69	−0.23
2	0.34	0.30	6.38	0.16	0.30	0.33	6.16	0.19	0.23	0.31	6.10	0.17
3	0.51	0.57	5.27	0.37	0.50	0.42	5.18	0.28	0.45	0.54	5.02	0.37
4	0.63	0.66	4.69	0.49	0.62	0.60	4.51	0.46	0.60	0.67	4.47	0.52
5	0.71	0.69	4.41	0.55	0.72	0.69	4.26	0.56	0.73	0.77	4.32	0.62
6	0.79	0.76	4.46	0.59	0.81	0.84	4.46	0.65	0.85	0.86	4.35	0.68
7	0.88	0.99	4.77	0.72	0.90	0.93	4.56	0.70	0.96	0.88	4.76	0.64
8	1.00	1.09	5.47	0.69	1.03	1.08	5.13	0.73	1.11	0.94	5.17	0.63
9	1.21	1.25	5.94	0.73	1.23	1.26	5.93	0.74	1.34	1.02	6.02	0.58
High (H)	1.83	1.69	7.29	0.80	1.89	1.75	7.51	0.81	1.99	1.46	7.40	0.68
H − L	1.86	2.12	6.13	1.20	2.01	2.26	5.80	1.35	2.22	1.97	5.93	1.15

Notes: OLS-3+H is ordinary least squares that preselect size, book-to-market, and momentum using Huber loss rather than the standard l2 loss. PLS is partial least squares. PCR ENet+H, GLM+H, RF, GBRT+H, and NN1 to NN5 are neural networks with one to five hidden layers.

Source: Gu et al. (2020).

The second consistent result is that what drives the outperformance of ML-based prediction versus that of linear models is that ML algorithms not only are limited to linear combinations of feature sets but can formulate higher-order functional dependencies, such as nonlinearity and interaction. To test whether higher-order effects can contribute to security prediction, one can compare linear machine learning models (for example, LASSO and RIDGE) with models that consider nonlinear and complex interaction effects (such as trees and neural networks). Investors at Robeco have found that higher-order machine learning models outperform their simpler linear competitors. The outperformance of higher-order machine learning models was also confirmed by academics,[22] as evident from Exhibit 7.

Between nonlinearity and interactions, interactions have the greater impact on model performance. This also can be seen in Exhibit 7, where the performance of the generalized linear model with Huber loss (GLM+H) is inferior to those models that consider interaction, boosted trees, and neural networks.

Third, ML investors have found that the features that most determine prediction outcomes are remarkably consistent regardless of the specific ML algorithms used. This is somewhat of a surprise because the various ML algorithms, such as boosted trees and neural networks, use dissimilar approaches to arrive at their outcomes. However, the similar importance assigned to certain input features confirms that these are salient characteristics that drive cross-sectional stock returns. From Robeco's own experience and various published studies, the characteristics that dominate cross-sectional returns are found to be short-term reversals, stock and sector return momentum, return volatility, and firm size.

Fourth, simple ML algorithms outperform more complicated ML algorithms. This result is very likely because, since there are not much data to train on, the simpler models, due to their parsimonious nature, are less likely to overfit and thereby perform better out of sample. We confirm this observation from Exhibit 7, where the best out-of-sample Sharpe ratio is achieved by a neural network with four hidden layers.

Fifth, the more data there are, the better the ML prediction algorithm performs. This is fundamental to the nature of ML algorithms, as observed by Halevy et al. (2009) for general machine learning applications and confirmed by Choi, Jiang, and Zhang (2022) in a financial context.

Predicting Stock Crashes

In this section, we discuss the example of using ML to predict stock crashes.

Investment problem

Rather than predicting the rank order of future returns, another application for ML algorithms may simply be to predict the worst-performing stocks—that is, those most likely to crash.[23] The results of this prediction can be applied in such strategies as conservative equities,[24] where the securities most likely to crash are excluded from the investment universe. That is, win by not losing.

Methodology

A crash event for equities is defined as a significant drop in a stock's price relative to peers; thus, it is idiosyncratic rather than systematic when a group of stocks or the market crashes. The ML prediction is set up as follows:

1. Investment universe: US, international, emerging market, and so on
2. ML algorithms: logistic regression, random forest, gradient boosted tree, and ensemble of the three
3. Feature set: various financial distress indicators—such as distance to default, volatility, and market beta—in addition to traditional fundamental financial metrics

As one can see, the problem setup is very similar to that of cross-sectional stock return prediction. The main differences are the objective function of the ML algorithm (ML prediction goal) and the input feature set (feature engineering). Care should be taken in determining both components to ensure the prediction algorithm solves the stated problem and performs well out of sample.

Results

The performance of the portfolio of stocks with the highest distress probability versus that of the market is shown in **Exhibit 8**. We see that the performance of the ML algorithm is greater than that obtained using traditional approaches for both developed and emerging markets.

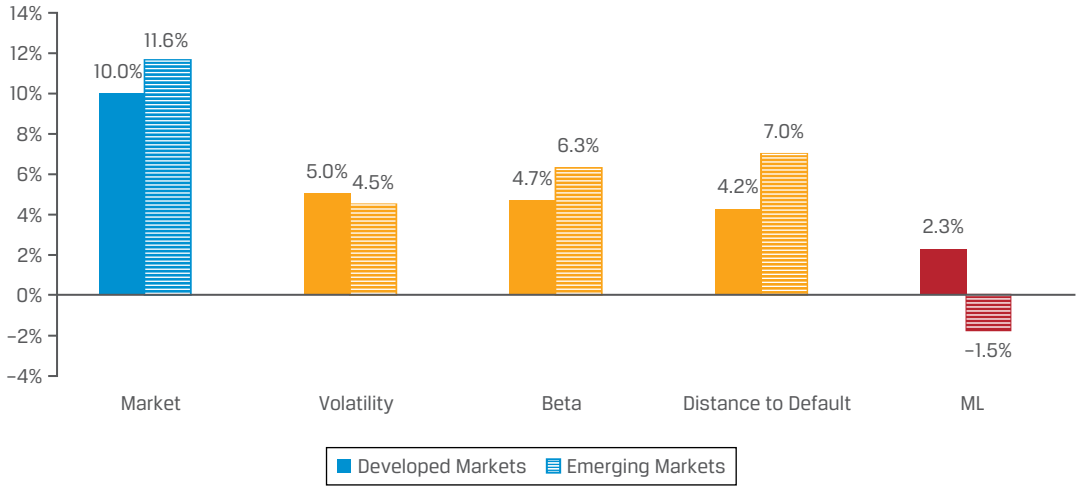
Looking at the sector composition of the likely distressed stocks, shown in **Exhibit 9**, we see that the ML algorithm choices are reasonable, as technology stocks dominated during the bursting of the dot-com bubble in the early 2000s and financial stocks dominated during the Global Financial Crisis of 2008. Overall, we see a wide sector dispersion for the likely distressed stocks, indicating that the prediction return is mostly from stock selection rather than sector allocation.

[22]For more discussion, see Choi, Jiang, and Zhang (2022) and Gu et al. (2020).

[23]This ML prediction task is studied in Swinkels and Hoogteijling (2022).

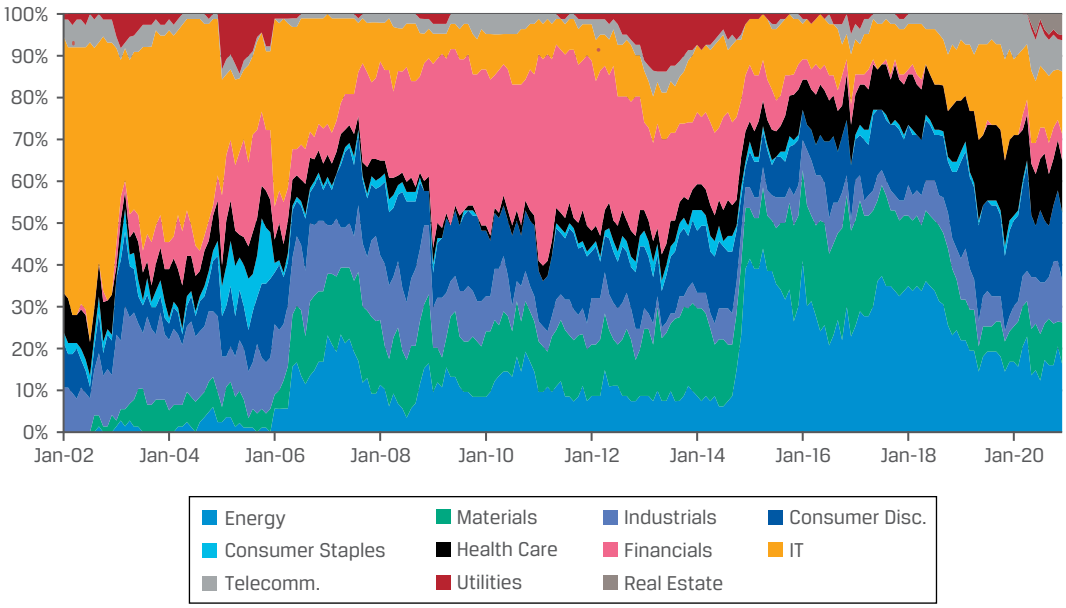
[24]Also called low-volatility equity, among other names.

Exhibit 8. Market Return vs. Return of a Portfolio Consisting of Likely Distressed Stocks, Estimated under Various Prediction Approaches



Source: Swinkels and Hoogteijling (2022).

Exhibit 9. Sector Composition of the ML-Predicted Likely Distressed Stocks



Source: Swinkels and Hoogteijling (2022).

Predicting Fundamental Variables

In this section, we discuss the example of using ML to predict company fundamentals.

Investment problem

Predicting stock returns is notoriously hard. As mentioned previously, there are potentially thousands of variables (dimensions) that can affect a stock's performance—investor sentiment, path dependency, and so on. The various factors, endogenous and exogenous, ultimately get translated into only a one-dimensional response—higher return (up) or lower return (down). An easier task may be to use ML algorithms to predict company fundamentals, such as return on assets and corporate earnings. Company fundamentals also have the additional beneficial characteristic of being more stable than stock returns, making them

better suited for ML predictions. Because of these reasons, company fundamentals prediction is another popular application of ML algorithms in the financial domain. In this section, we look at the findings from a popular study[25] where ML algorithm-predicted earnings forecasts are compared with those from more traditional approaches.

Methodology

The investment universe is the US stock market, excluding the financial and utility sectors. The study was conducted over the period 1975–2019.

Six different ML models were tested: three linear ML models (OLS, LASSO, and ridge regression) and three nonlinear ML models (random forest, gradient boosting regression, and neural networks). Six traditional models were used as a benchmark to compare against ML models: random walk, autoregressive model, models from Hou, van Dijk, and Zhang (2012; HVZ) and So (2013; SO), the earnings persistence model, and the residual income model. Various ensembles of these models were also tested.

The feature set is composed of 28 major financial statement line items and their first-order differences. So, there are 56 features in total.

Results

The results are shown in **Exhibit 10**. Consistent with the results for cross-sectional stock returns, Cao and You (2021) found that machine learning models give more accurate earnings forecasts. The linear ML models are more accurate than the benchmark traditional models (by about 6%), and the nonlinear ML models are more accurate than the linear ML models (by about 1%–3% on top of the linear model). Not only are the ML models more accurate; the traditional models autoregression, HVZ, SO, earnings persistence, and residual income were not more accurate than the naive random walk model. The ensemble models, traditional or machine learned, were more accurate than the individual models alone, with the order of accuracy preserved: ML ensemble beating traditional methods, ensemble and nonlinear ML ensemble beating linear ML ensemble.

The ML models' better performance can be attributed to the following:

- They are learning economically meaningful predictors of future earnings. One can make this conclusion by examining feature importance through such tools as Shapley value.
- The nonlinearity and interaction effects are useful in further refining the accuracy of forward earnings predictions, as evidenced by the higher performance of nonlinear ML models compared with linear ML models.

The takeaway from this study is the same as in the previous two examples. That is, ML models can provide value on top of traditional models (especially due to nonlinearity and interaction components), and ensembling is one of the closest things to a free lunch in ML, much like diversification for investing.

NLP in Multiple Languages

So far, we have discussed problems where ML algorithms are used for prediction. Another major category for ML applications is textual language reading, understanding, and analysis, which is called "natural language processing" (NLP). Modern NLP techniques use neural networks to achieve the great capability improvements they have made in recent years. NLP is discussed in other chapters of this book, so we will not discuss the techniques extensively here, but we will discuss one NLP application that can be interesting for practitioners.

Investment problem

Investing is a global business, and much of the relevant information for security returns is written in the local language. In general, a global portfolio may invest in 20–30 different countries,[26] while a typical investor may understand only two or three languages, if that. This fact presents a problem, but fortunately, it is a problem that we can attempt to solve through ML algorithms.

From the perspective of Western investors, one language of interest is Chinese. The Chinese A-share market is both large and liquid. But understanding Chinese texts on A-share investing can be challenging because Chinese is not an alphabetical language and it follows very different grammatical constructs than English. In addition, since retail investors dominate the Chinese A-share market,[27] a subculture of investment slang has developed, where terms used are often not standard Chinese, thereby compounding the problem for investors without a strong local language understanding.

In a paper by Chen, Lee, and Mussalli (2020), the authors applied ML-based NLP techniques to try to understand investment slang written in Mandarin Chinese by retail investors in online stock discussion forums. We discuss the results of that paper here.

[25]For more details, see Cao and You (2021). The problem of ML company fundamentals prediction was also examined in Alberg and Lipton (2017).

[26]The MSCI All Country World Index (ACWI) covers stocks from 26 countries.

[27]Some studies have concluded that in the A-share market, retail trading volume can be up to 80% of the total. In recent years, institutional market trading has proportionally increased as the Chinese market matures.

Exhibit 10. Prediction Accuracy Results from Cao and You (2021)

	Mean Absolute Forecast Errors				Median Absolute Forecast Errors			
		Comparison with RW				Comparison with RW		
	Average	DIFF	*t*-Stat	%DIFF	Average	DIFF	*t*-Stat	%DIFF
Benchmark model								
RW	0.0764				0.0309			
Extant models								
AR	0.0755	−0.0009	−2.51	−1.15%	0.0308	−0.0001	−0.22	−0.24%
HYZ	0.0743	−0.0022	−3.63	−2.82%	0.0311	0.0002	0.64	0.76%
EP	0.0742	−0.0022	−2.79	−2.85%	0.0313	0.0004	1.02	1.42%
RI	0.0741	−0.0023	−3.15	−3.07%	0.0311	0.0002	0.66	0.74%
SO	0.0870	0.0105	5.19	13.78%	0.0347	0.0039	5.50	12.56%
Linear machine learning models								
OLS	0.0720	−0.0045	−5.04	−5.83%	0.0306	−0.0002	−0.60	−0.73%
LASSO	0.0716	−0.0048	−5.32	−6.31%	0.0304	−0.0004	−1.11	−1.43%
Ridge	0.0718	−0.0047	−5.19	−6.11%	0.0305	−0.0003	−0.87	−1.08%
Nonlinear machine learning models								
RF	0.0698	−0.0066	−6.44	−8.64%	0.0296	−0.0012	−3.10	−3.97%
GBR	0.0697	−0.0068	−6.08	−8.86%	0.0292	−0.0016	−4.23	−5.34%
ANN	0.0713	−0.0051	−5.38	−6.67%	0.0310	0.0001	0.24	0.38%
Composite models								
COMP_EXT	0.0737	−0.0027	−3.89	−3.58%	0.0311	0.0002	0.56	0.66%
COMP_LR	0.0717	−0.0047	−5.25	−6.16%	0.0305	−0.0004	−1.02	−1.33%
COMP_NL	0.0689	−0.0075	−6.99	−9.87%	0.0292	−0.0017	−3.92	−5.55%
COMP_ML	0.0693	−0.0071	−7.12	−9.35%	0.0294	−0.0015	−3.75	−4.81%

Notes: RW stands for random walk. AR is the autoregressive model. HVZ is the model from Hou et al. (2012). SO is the model from So (2013). EP is the earnings persistence model. RI is the residual income model. RF stands for random forests. GBR stands for gradient boost regression. ANN stands for artificial neural networks. COMP_EXT is an ensemble of traditional models. COMP_LR is an ensemble of linear ML models. COMP_NL is an ensemble of nonlinear ML models. COMP_ML is an ensemble of all ML models.

Source: Cao and You (2021).

Methodology

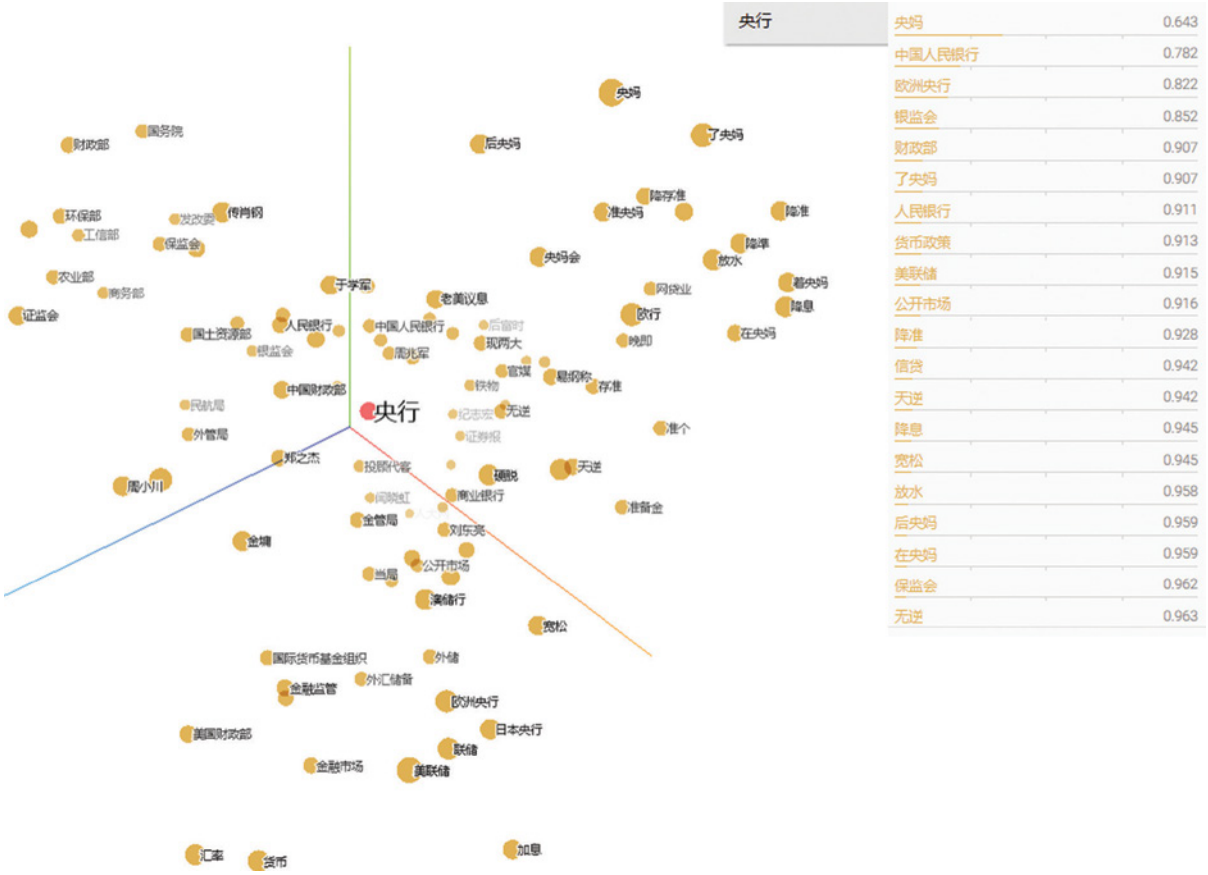
1. Download and process investment blogs actively participated in by Chinese A-share retail investors.
2. Apply embedding-based NLP techniques and train on the downloaded investment blogs.
3. Starting with standard Chinese sentiment dictionaries, look for words surrounding these standard Chinese sentiment words. By the construct of the embedding models, these surrounding words often have the same contextual meaning as the words in the standard sentiment dictionaries, whether they are standard Chinese or slang. An example of this is shown in **Exhibit 11**.

Results

Using this technique, we can detect investment words used in standard Chinese and the slang used by retail investors. In Exhibit 11, the red dot illustrates the Chinese word for "central bank." The column on the right side of Exhibit 11 shows the words closest to this word, and the closest is the word that translates to "central mother" in Chinese. This is a slang word often used by Chinese retail investors as a substitute for the word "central bank" because central banks often take actions to calm down market tantrums, much like a mother does to her children when they have tantrums.

The embedded words also exhibit the same embedded word vector arithmetic made famous by the following example (see Mikolov, Yih, and Zweig 2013): King – Man + Woman ≈ Queen. For example, **Exhibit 12** shows the following embedded Chinese word relationship: Floating red – Rise + Fall ≈ Floating green.[28]

Exhibit 11. Embedded Chinese Words from A-Share Investor Blogs Projected onto a 3-D Space



Source: Chen et al. (2020).

Exhibit 12. Chinese Word Embedding Still Preserves Vector Arithmetic

飘红 – 涨 + 跌 ≈ 飘绿

Source: Chen et al. (2020).

[28]In contrast to most of the world's markets, in the Chinese stock market, gains are colored red whereas losses are colored green.

This study illustrates that ML techniques not only are useful for numerical tasks of results prediction but can also be useful in other tasks, such as foreign language understanding.

Conclusion

In this chapter, we gave an overview of how ML can be applied in financial investing. Because there is a lot of excitement around the promise of ML for finance, we began the chapter with a discussion on how the financial market is different from other domains in which ML has made tremendous strides in recent years and how it would serve the financial ML practitioner to not get carried away by the hype. Applying ML to the financial market is different from applying ML in other domains in that the financial market does not have as much data, the market is nonstationary, the market can often behave irrationally because human investor emotions are often a big driver of market returns, and so on. Given these differences, we discussed several common pitfalls and potential mitigation strategies when applying machine learning to financial investing.

In the second half of the chapter, we discussed several recent studies that have applied ML techniques to investment problems. Common findings of these studies are as follows:

- ML techniques can deliver performance above and beyond traditional approaches if applied to the right problem.
- The source of ML algorithms' outperformance includes the ability to consider nonlinear and interaction effects among the input features.
- Ensembling of ML algorithms often delivers better performance than what individual ML algorithms can achieve.

We showed that in addition to predicting numerical results, ML could also help investors in other tasks, such as sentiment analysis or foreign language understanding. Of course, the applications discussed here are only a small subset of what ML can do in the financial domain. Other possible tasks include data cleaning, fraud detection, credit scoring, and trading optimization.

Machine learning is a powerful set of tools for investors, and we are just at the beginning of the journey of applying ML to the investment domain. Like all techniques, machine learning is powerful only if applied to the right problems and if practitioners know the technique's limits. Having said that, we believe one can expect to see a lot more innovation and improved results coming out of this space going forward.

References

Abe, M., and H. Nakayama. 2018. "Deep Learning for Forecasting Stock Returns in the Cross-Section." In *Advances in Knowledge Discovery and Data Mining,* edited by D. Phung, V. Tseng, G. Webb, B. Ho, M. Ganji, and L. Rashidi, 273–84. Berlin: Springer Cham.

Alberg, J., and Z. Lipton. 2017. "Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals." Cornell University, arXiv:1711.04837 (13 November). https://arxiv.org/abs/1711.04837.

Arnott, R., C. Harvey, and H. Markowitz. 2019. "A Backtesting Protocol in the Era of Machine Learning." *Journal of Financial Data Science* 1 (1): 64–74.

Avramov, D., S. Cheng, and L. Metzker. 2022. "Machine Learning vs. Economic Restrictions: Evidence from Stock Return Predictability." *Management Science* (29 June).

Avramov, D., T. Chordia, and A. Goyal. 2006. "Liquidity and Autocorrelations in Individual Stock Returns." *Journal of Finance* 61 (5): 2365–94.

Birge, J., and Y. Zhang. 2018. "Risk Factors That Explain Stock Returns: A Non-Linear Factor Pricing Model." Working paper (9 August).

Cao, K., and H. You. 2021. "Fundamental Analysis via Machine Learning." Emerging Finance and Financial Practices eJournal.

Cao, Larry. 2018. "Artificial Intelligence, Machine Learning, and Deep Learning: A Primer." *Enterprising Investor* (blog; 13 February). https://blogs.cfainstitute.org/investor/2018/02/13/artificial-intelligence-machine-learning-and-deep-learning-in-investment-management-a-primer/.

Chen, M., J. Lee, and G. Mussalli. 2020. "Teaching Machines to Understand Chinese Investment Slang." *Journal of Financial Data Science* 2 (1): 116–25.

Choi, D., W. Jiang, and C. Zhang. 2022. "Alpha Go Everywhere: Machine Learning and International Stock Returns." Working paper (29 November). Available at https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3489679.

Columbus, Louis. 2020. "The State of AI Adoption In Financial Services." *Forbes* (31 October). www.forbes.com/sites/louiscolumbus/2020/10/31/the-state-of-ai-adoption-in-financial-services/?sh=1a4f7be62aac.

Daul, S., T. Jaisson, and A. Nagy. 2022. "Performance Attribution of Machine Learning Methods for Stock Returns Prediction." *Journal of Finance and Data Science* 8 (November): 86–104.

Fama, E., and K. French. 1993. "Common Risk Factors in the Returns on Stocks and Bonds." *Journal of Financial Economics* 33 (1): 3–56.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep Learning*. Cambridge, MA: MIT Press.

Grinold, R., and R. Kahn. 1999. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk*. New York: McGraw Hill.

Gu, S., B. Kelly, and D. Xiu. 2020. "Empirical Asset Pricing via Machine Learning." *Review of Financial Studies* 33 (5): 2223–73.

Halevy, A., P. Norvig, and F. Pereira. 2009. "The Unreasonable Effectiveness of Data." *IEEE Intelligent Systems* 24 (2): 8–12.

Hanauer, M., and T. Kalsbach. 2022. "Machine Learning and the Cross-Section of Emerging Market Stock Returns." Working paper (5 December).

Harvey, C., Y. Liu, and H. Zhu. 2016. "… and the Cross-Section of Expected Returns." *Review of Financial Studies* 29 (1): 5-68.

Hastie, T., R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer.

Hou, K., M. van Dijk, and Y. Zhang. 2012. "The Implied Cost of Capital: A New Approach." *Journal of Accounting and Economics* 53 (3): 504–26.

Hou, K., C. Xue, and L. Zhang. 2020. "Replicating Anomalies." *Review of Financial Studies* 33 (5): 2019–133.

Jessen, C., and D. Lando. 2013. "Robustness of Distance-to-Default." 26th Australasian Finance and Banking Conference 2013 (16 August).

Leippold, M., Q. Wang, and W. Zhou. 2022. "Machine Learning in the Chinese Stock Market." *Journal of Financial Economics* 145 (2): 64–82.

Leung, E., H. Lohre, D. Mischlich, Y. Shea, and M. Stroh. 2021. "The Promises and Pitfalls of Machine Learning for Predicting Stock Returns." *Journal of Financial Data Science* 3 (2): 21–50.

Li, Y., Z. Simon, and D. Turkington. 2022. "Investable and Interpretable Machine Learning for Equities." *Journal of Financial Data Science* 4 (1): 54–74.

Li, Y., D. Turkington, and A. Yazdani. 2020. "Beyond the Black Box: An Intuitive Approach to Investment Prediction with Machine Learning." *Journal of Financial Data Science* 2 (1): 61–75.

Lo, A., D. Repin, and B. Steenbarger. 2005. "Greed and Fear in Financial Markets: A Clinical Study of Day-Traders." NBER Working Paper No. w11243.

López de Prado, M. 2018. "The 10 Reasons Most Machine Learning Funds Fail." *Journal of Portfolio Management* 44 (6): 120–33.

López de Prado, M. 2019. "Ten Applications of Financial Machine Learning." Working paper (22 September).

Lundberg, S., and S. Lee. 2017. "A Unified Approach to Interpreting Model Predictions." NIPS 17: Proceedings of the 31st Conference on Neural Information Processing Systems: 4768–77.

McLean, R., and J. Pontiff. 2016. "Does Academic Research Destroy Stock Return Predictability?" *Journal of Finance* 71 (1): 5–32.

Mikolov, T., W. Yih, and G. Zweig. 2013. "Linguistic Regularities in Continuous Space Word Representations." Proceedings of NAACL-HLT 2013: 746–51. www.aclweb.org/anthology/N13-1090.pdf.

Nadeem, Monisa. 2018. "Clearing the Confusion: AI vs. Machine Learning vs. Deep Learning." Global Tech Council (25 November). www.globaltechcouncil.org/artificial-intelligence/clearing-the-confusion-ai-vs-machine-learning-vs-deep-learning/.

Rasekhschaffe, K., and R. Jones. 2019. "Machine Learning for Stock Selection." *Financial Analysts Journal* 75 (3): 70–88.

So, E. 2013. "A New Approach to Predicting Analyst Forecast Errors: Do Investors Overweight Analyst Forecasts?" *Journal of Financial Economics* 108 (3): 615–40.

Swinkels, L., and T. Hoogteijling. 2022. "Forecasting Stock Crash Risk with Machine Learning." White paper, Robeco.

Tobek, O., and M. Hronec. 2021. "Does It Pay to Follow Anomalies Research? Machine Learning Approach with International Evidence." *Journal of Financial Markets* 56 (November).