By Ainsley To

DIGGING DEEPER

Machine learning and investing: the cautious seldom err or write great poetry

Machine learning brings many advantages to the investment world. It may be complex, but that's no reason to discard it.

Machine learning (ML) is the nebulous intersection of computer science and statistics. But is it a new reality for investors or just hype that will fade into a new "AI winter"?

My old rule of thumb to differentiate substance from marketing, was simple: If it was mainly written in Python, it was of potential substance. If it was mainly written in PowerPoint, it was likely "Artificial Intelligence" (aka marketing skulduggery).

After conceding that this was not a robust approach, and spending some time re-educating myself (getting my hands dirty and building the models from scratch), my position has slowly evolved from outright cynic to sceptical enthusiast.

I should emphasise that there is no substitute for putting in the time and effort to learn the details (Robert Tibshirani's *Elements of Statistical Learning*, first published in 2001, would be a great starting point).

Going through individual ML algorithms is beyond the scope of this article. My goal, instead, is to provide (in the simplest terms possible) some explication for investors, using investing analogies and concepts familiar to those in finance. Or, failing that, perhaps to help mildly improve your cocktail party soundbites on the topic.

To paraphrase George Box's quote on models: "All analogies are wrong, but some are useful."

We can contrast an ML approach with that of traditional equity investors using a toy example:

- **Investor A** is a traditional value investor who believes that there is a relationship between valuation and expected returns – the lower the valuation of a stock, the higher its expected return.
- **Investor B** also believes that there is a non-linear relationship between valuation and expected returns – the lower the

valuation, the higher the expected return, until a certain point beyond which low valuations are a sign of financial distress (and thus have lower expected returns). Thus, she wants to buy cheap companies but avoid the very cheapest companies.

■ **Investor C** believes in using multiple metrics to forecast returns. In addition to value, she believes that measuring the quality of a company can be used to avoid value traps. Her view is that expensive companies that are low-quality will have low returns in the future.

Consider each investor's models of expected returns stylised in these charts. For Investor A, returns are linearly related to a single variable – a straight line between valuation and returns. Investor B also considers only valuation but in a non-linear manner – a curve with expected returns peaking and declining for the extremely cheap valuations. Investor C is the most complex to visualise since she is concerned with the interaction between value and quality, which requires a 3D chart – expected returns are low for expensive companies of low quality.

This example points to two patterns that ML can improve over linear models. It has the flexibility to capture non-linear relationships (such as in B's case) as well as interactions between variables (as with Investor C).

However, an ML process gets there in a different way than in our example – while our investors make assumptions about expected returns, an ML algorithm instead learns the relationships directly from the data.

ML can find non-linear interactions between as many variables as it is given, in whatever combinations best fit the data. The difficulty is that the patterns the algorithms learn are rarely interpretable for a human investor and increasingly difficult the more features you give it to train on – we can see in only three dimensions and additional inputs beyond our example will be in higher-dimensional space.

This is one of the main trade-offs in using ML – performance (making the best predictions) versus interpretability (understanding why they made those predictions). The broad church of ML models all vary along this spectrum, but the two that are known to have the best performance (random forests and neural networks) are also among the hardest to interpret.

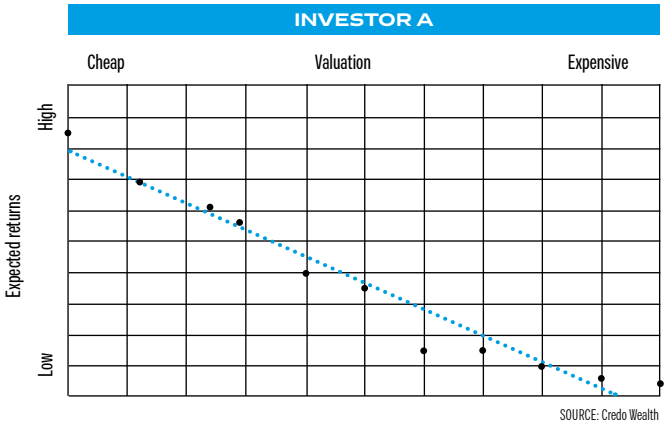
How do ML algorithms find patterns?

Every ML algorithm will at some stage involve numerical optimisation – a sophisticated form of trial and improvement. Running the same algorithm on the same data can give different results – it depends on what values are trialled and in what order, which are often randomly chosen.

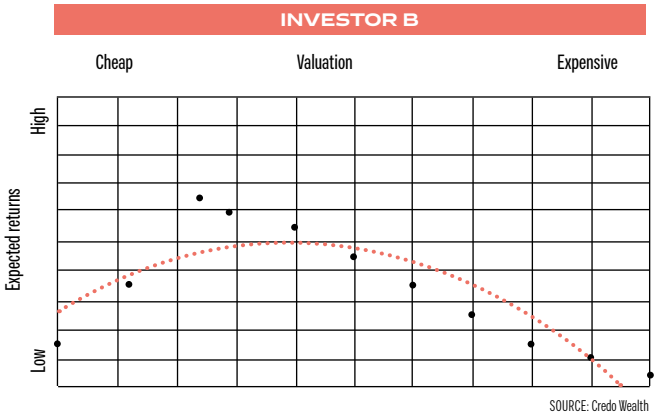
But, with the flexibility they are given, enough iterative trial and improvement will eventually get to an answer that looks good in sample (e.g. within the data used to train the algorithm). The problem in investing which we all know too well is that "past performance is not indicative of future performance". Overfitting is a significant risk in ML algorithms since they are trained to fit the past and they are given significant freedom to do so.

In contrast to traditional linear regression models that measure their goodness of fit on the whole dataset, the practice in ML is to optimise out of sample prediction – performance is measured using data that was not used to train the algorithm. Cross-validation is the process of withholding data from the algorithm for testing performance after training. For example, you can split ten years of data into two sets: from 2009 to 2017 and from 2018 to 2019.

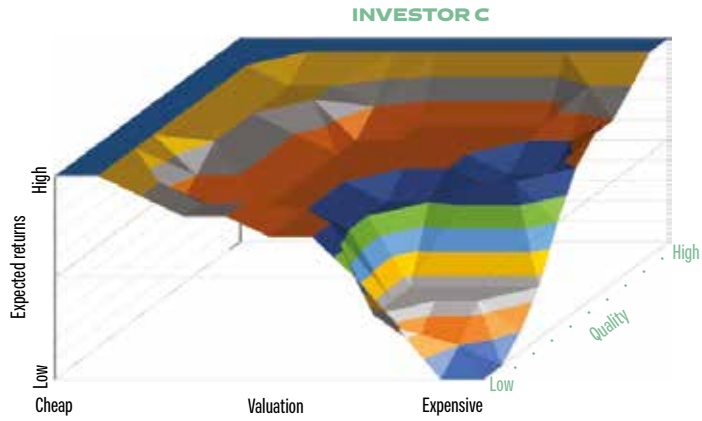
You create your strategy based on the training set (2009 to 2017), then see how it performs on the test set from 2017 to 2019 (data which was not involved in forming the strategy itself), providing a level of comfort that the algorithm is not simply overfitting noise in the training data. There are many other tools and techniques to constrain ML algorithms and avoid overfitting. The balancing act between overfitting (leading to variance in predictions) and constraining the algorithm (which increases bias in predictions) is another main consideration when using ML.



SOURCE: Credo Wealth



SOURCE: Credo Wealth



SOURCE: Credo Wealth

"If a method works, it should not be abandoned or dismissed just because theorists haven't yet figured out how to explain it."

Keep an open mind... but not so open that your brain falls out

Financial markets are extremely complex, with non-linear relationships and interactions between explanatory variables. It's therefore no surprise that simple linear models have difficulty capturing all the nuances.

However, the use of more complex tools is no guarantee of success due to properties that are unique to finance as a domain. The ratio of signal-to-noise in financial data is low by design. There are strong financial incentives to take advantage of any informational content in markets. When market participants act on this information, they drive prices and absorb the remaining amount of signal in the system – often to the point where it is too costly or risky to act on what is remaining.

This is why efficient markets can be approximated with random walks, since much of the movement in securities prices is due to news, which is by definition unpredictable.

Michael Brandt from Duke University gave an excellent illustration in his presentation, "We've got much less data than you think". The input data used for self-driving cars has a near-perfect signal and no noise. In stark contrast, the pictures show what the input into the algorithm would be if it had a similar signal-to-noise ratio of annual (1 to 3 signal to noise) and monthly financial data (1 to 10 signal to noise).

From a pure return forecasting perspective, it is unreasonable to expect that simply using the same algorithms from other domains will produce similar results in financial data.

Conclusion

ML generalises methods we already know to allow for non-linearity and interaction effects. Investors have been familiar with facets of ML for decades now – Bryan Kelly from Yale elegantly highlighted how sequential sorting in the famous Fama French factor portfolios are simple tree models (the building blocks for random forests).

Ensemble learning (combining ML models to produce an aggregate forecast) utilises the benefit of diversifying away uncorrelated errors, a concept that should be familiar to most portfolio managers. With some careful engineering, there are applications for ML across many parts of the investment process – not just the narrow return prediction context that I've focused on in this article.

There is credible concern over the dangers of overfitting. But this is not unique to ML – given the p-hacking (or selective reporting) epidemic in academic finance, one can argue we crossed the Rubicon of overfitting with traditional econometric models long ago.

A more difficult obstacle is the interpretability issue, which is particularly tangible for myself, as a practitioner who suffers the perpetual anxieties of alpha decay and understands the logistical realities of investment committees.

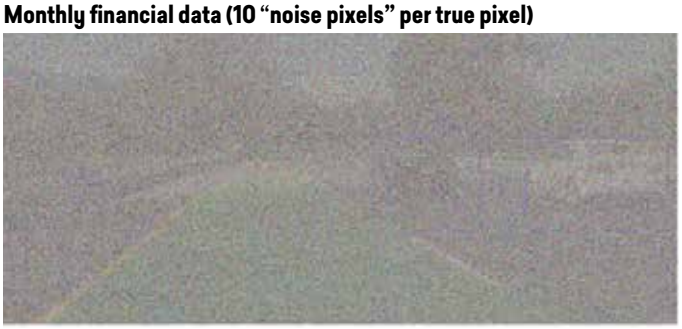
A pioneer in deep learning, Yann LeCun, once said: "There is a need for better theoretical understanding of deep learning. But if a method works, it should not be abandoned or dismissed just because theorists haven't yet figured out how to explain it."

Self-driving cars (Perfect signal-to-noise ratio)

≠

Annual financial data (3 "noise pixels" per true pixel)

Monthly financial data (10 "noise pixels" per true pixel)



SOURCE: Michael Brandt's presentation, "We've got much less data than you think"

My sentiment on ML in investing can be summarised by Yann LeCun's view and Einstein's famous quote to, "Keep things as simple as possible but no simpler."

Financial markets are one of the most complex puzzles human beings have ever encountered. To even stand a chance, we need to explore tools that are equal to the task. The aspiration is to do so with the appropriate level of pragmatism and intellectual honesty. ■

Ainsley To is head of the multi-asset team at Credo Wealth.