NEWTON

Investment Management

BNY MELLON | INVESTMENT MANAGEMENT

SEVEN DEADLY SINS OF QUANTITATIVE INVESTING USING MACHINE LEARNING

March 2023

Dimitri Curtil
David Lu

The views and opinions expressed in this document are those of Newton and should not be construed as a recommendation or investment advice. For institutional investors only. Please read the important disclosure at the back of this document.

THE TEMPTATIONS BEHIND THE RISE OF THE MACHINES

In recent years, the world has seen rapid advancements in computing power, enabling and expanding interests in the field of artificial intelligence and machine learning (ML). Unleashed from the ivory towers of academia and Silicon Valley tech giants, previously taboo techniques and approaches such as supervised learning, unsupervised learning and genetic algorithms have gained greater acceptance.

Various practitioners from within the quantitative investment industry, as well as part-time, data science-savvy investment aficionados, have tapped into these new tools. The relative simplicity and economic rationale of traditional linear, structural models have given way to this rise of the machines. Two factors have abetted this escalation: a massive proliferation of datasets (both non-traditional and unstructured) and the availability of out-of-the-box auto-ML programs and packages. Now, even a relative novice can mine vast amounts of raw datasets in search of the Holy Grail: untapped alpha. Herein lurks danger and temptation, in our view. With such powerful tools and a relatively young and nascent field, there are even fewer guardrails to help practitioners avoid some common pitfalls of ML-based quantitative investing.

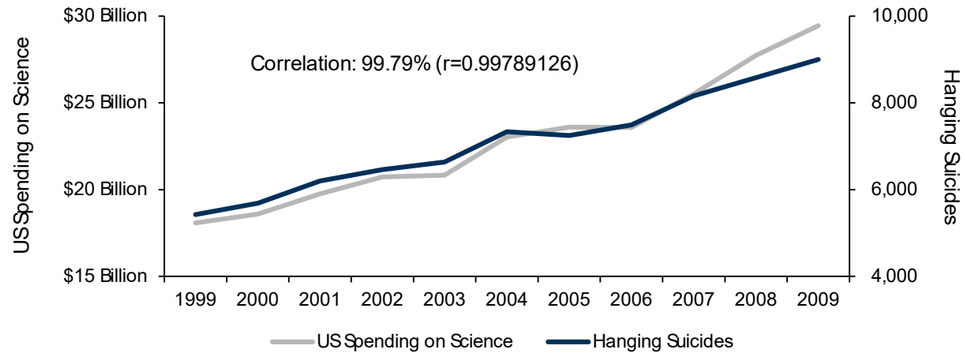
The following discussion is intended for a broad audience, including portfolio managers and quantitative researchers who have some appreciation of the quantitative methods applied to investment problems. We hope it will be especially poignant for those who may have already explored one or more ML methods in their search for alpha. This is intended to be a high-level discussion of some common pitfalls and is by no means an exhaustive treatment of all the nuances and granularities involved in applying ML methods toward an investment problem. Most of the pitfalls also apply to traditional quantitative research.

In an effort to bring some levity to an otherwise serious topic, we give you the seven 'deadly sins' of machine learning.

Sin #1. Greed (Overfit): Machine learning may find patterns where there should be none

Below is an example of a spurious correlation, where patterns are found purely by chance without any underlying rationale for the existence of a causal relationship.

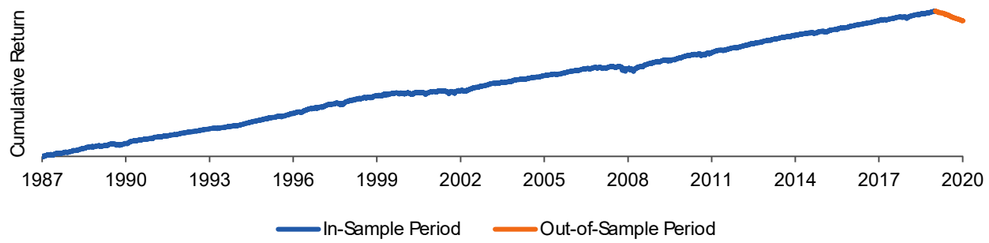
US Spending on Science, Space, and Technology Correlates With Suicides by Hanging, Strangulation, and Suffocation



Source: tylervigen.com. Data sources: U.S. Office of Budget and Management and Center for Disease Control & Prevention.

Another example shows the amplification of such spurious behavior in the age of machine learning. Below is a simulated timing strategy built using supervised ML methods. The goal was to time exposure to the S&P 500® Index—ideally being long when the S&P 500 is up and short when the S&P 500 is down. The model is fitted from 1987 to 2019 using daily data. The model performance is shown below.

In-Sample Overfit



Source: Newton.

There is a problem with the strategy, however. The input signals, or features in supervised learning parlance, are all random 'garbage' that should not in any way correlate with, or lead the price of, the S&P 500 Index.

Top Five Strategy Features	Feature Weight
1. Temperature in Washington DC	35%
2. The price of a Big Mac in the US	26%
3. The US death rate from heart disease	21%
4. Diurnal activity of rodents in the Grand Canyon	18%
5. Population growth of the San Francisco Bay Area	11%

Source: Newton.

The above is an example of how ML algorithms, by their very nature of being more powerful at identifying non-linear relationships relative to the linear models obtained with traditional econometric techniques, have a tendency to overfit the problem to noise during training. Invariably, if the trained model is then tested out of sample, the out-of-sample performance exhibits significant degradation as shown in the orange line in the chart above.

Aside from the more classical notion of 'in-sample' overfit, ML approaches, if not carefully applied, are also more likely to introduce insidious forms of overfit such as feature selection overfit and parameter choice overfit.

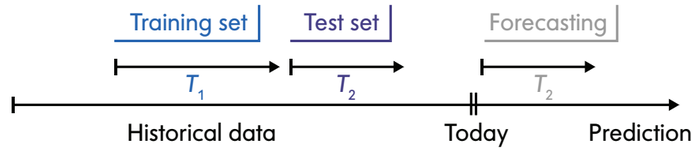
Feature selection overfit occurs when features are highly tuned to the training period, resulting in a near-perfect perfect fit, whereas out of sample they fail to generalize (as in the example above).

Parameter-choice overfit arises when calibration of the parameters (e.g. look back window, decay factor, coefficients, etc.) is so highly tuned that there is tendency to fail to generalize in real-world testing after the initial model building/training and calibration.

Sin #2. Lust (Data Snooping): Train model, peek ahead, adjust and repeat

Data snooping is the practice of looking at the testing or 'hold-out' dataset and applying tweaks to the ML algorithms and models to maximize hold-out performance. When done repeatedly, this practice contaminates and destroys the value of the testing or hold out data. This is a recognized issue for quantitative back testing in general, but it is likely to be even more pronounced when deploying ML algorithms.

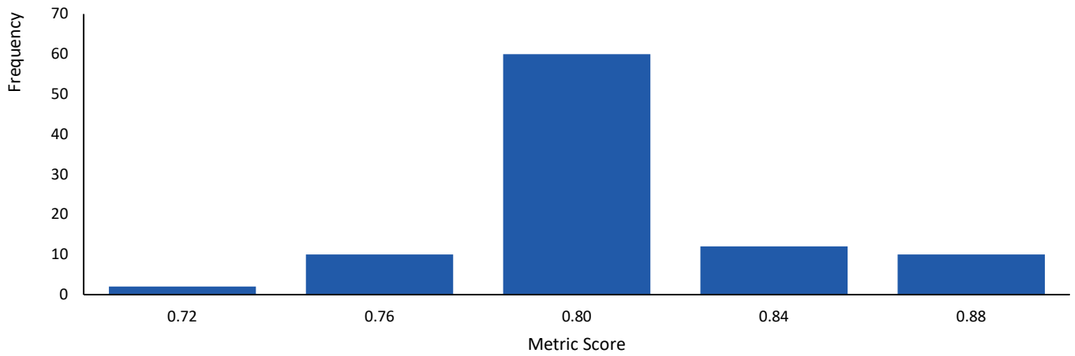
The below set up is typical. Training data is used to build a model, and test data is reserved for testing or cross validation. Finally, a hold-out dataset checks the performance of the model on data that has not been 'seen' by the model or undergone the iterations of training and testing. If the modeler repeats this cycle and repeatedly tweaks the model after 'peeking' at the hold-out dataset, this is a form of data snooping.



Repeated in-sample training, followed by assessing out-of-sample performance, destroys the value of the out-of-sample hold out (ideally, one should only 'touch once' the out-of-sample data).

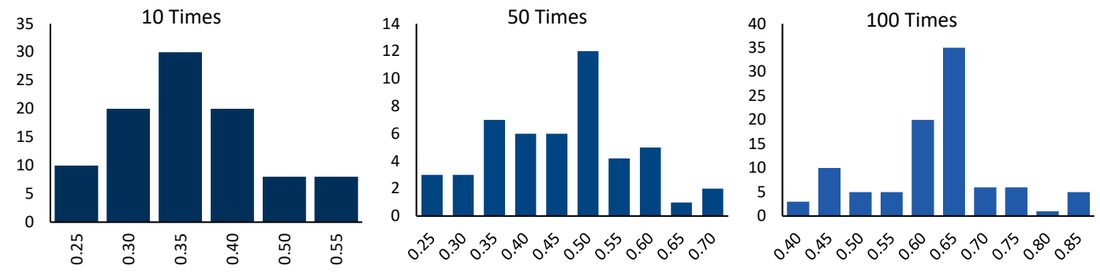
The impact of repeated in-sample (IS) and out-of-sample (OOS) train/look iterations is features with random noise instead of the desired real feaatures.

Trained on 'Real' Features



Source: IRIS Classification Dataset. Random forest classifier trained on 'real' features test on OOS Hit rate (mean ~=0.8)

Trained on 70% Random Noise



Increasing IS/OOS train/look iteration: Training is done on 70% random noise data (only 0.37 correlated to the 'real' features)

Sin #3. Anger: Irrationality when the fancy ML method does not work

ML approaches require a level of understanding and appreciation. At times, there can be an unreasonable expectation that ML techniques should be all powerful. When a desired outcome is not reached, anger, confusion and irrational behavior can set in. As with any technique, ML algorithms have advantages and limitations. One should understand them before getting emotional about outcomes that are difficult to understand or fall short of those that were initially anticipated.

One circumstance that can bring forth negative emotion is a support vector machine (SVM) model. Adverse effects can arise from a lack of scaling in the feature inputted, and the need to understand that SVM is a scale-dependent algorithm such that all features must to be pre-scaled. Applying scale-dependent algorithms requires special care, as the choice of scaling and how to scale is often not clear cut. For example, using in-sample scaling (versus a rolling-based pre-scaling) often gives better model performance, but is itself another sin: look-ahead bias.

The example below shows that context matters. One may be tempted to say that Model 1 is better, but as shown below, Model 1 happens to be an 'easier' problem where the baseline/random guess performance is quite good. Even though Model 2 only gets a slightly above-random guess, it is the superior model when adjusted for background random noise. Background 'noise' in this case simply refers to the performance of a random guess model that picks the image classification based purely on luck for the problem at hand. In the case of an animal versus an inanimate object, the degree of difficulty is smaller than that of distinguishing between cats and dogs, and thus the two problems have different levels of background noise. In controlling for background noise, we can judge that the performance of a real model is better than a random guess model for this particular problem.

Hit Rate (Prior Matters/Context Matters)
Which Model Is Better?

Source: Newton.

It depends on the underlying problem. For example, Model 1 may be predicting image of cat vs piece of wood (easier problem) and Model 2 may be predicting cat vs dog (more difficult problem).

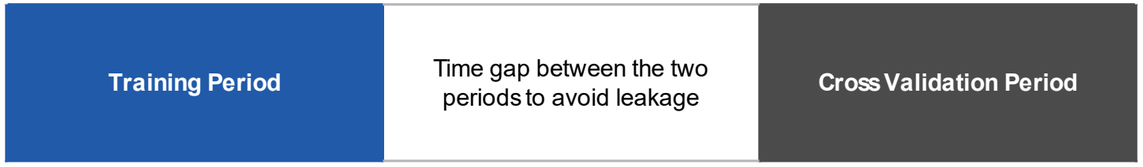
If the random-guess model gets 0.62 hit rate for 1, and 0.2 hit rate for 2, then Model 2 is much better than Model 1, given the underlying difficulty of the problem.

Sin #4. Gluttony (indiscriminate use of ML): One size does not fit all

Due to the power of the ML algorithms and the more recent proliferation of auto-ML packages, there is a tendency to apply one's favorite out-of-the-box ML algorithm to all problems at hand. This is especially dangerous if ne does not understand the nuances of ML algorithms.

One commonly used out-of-the box ML learner is called random forest. This is a powerful technique for many problems, but it does have pitfalls when it comes to financial applications. Random forest uses RANDOM sub-sample shuffling, in that the data is assumed to be homogenous. After the shuffling, the time sequence order is destroyed, which presents unique problems when modelling timeseries data.

For instance, the illustration below shows a simple two-fold training, cross validation scheme without any time gap in between the training period and the cross validation period. As such, there is 'leakage' that exists between the two sample blocks when there are time series-based signals, such as momentum, used as one of the input features. A more appropriate scheme is to leave out a 'time gap' that is at least the same length as the longest momentum signal (e.g., if longest momentum window used is 63 days, then the gap should be at least 63 days between the training and cross-validation periods).



More appropriate sampling schemes should seek to have some of the following features:

- No random shuffle of the time sequence of data (for time series modelling).
- No overlapping period between sub-sample periods used for each training or cross validation testing to avoid problems of 'leakage.'
- More balanced datasets between training and testing to minimize instances of unintended bias when looking at the same model selection criteria on testing datasets versus training datasets.

Sin #5. Sloth (bad data practices): Garbage in, garbage out

As seen in the earlier example, ML will not work if the underlying data is bad. There are various forms of bad data practices. Below are some of the most common, with suggestions for overcoming them.

Missing Data/Outliers

Missing data: Most ML algorithms do not work well with missing data. As a result, preprocessing is usually needed to impute the missing data.

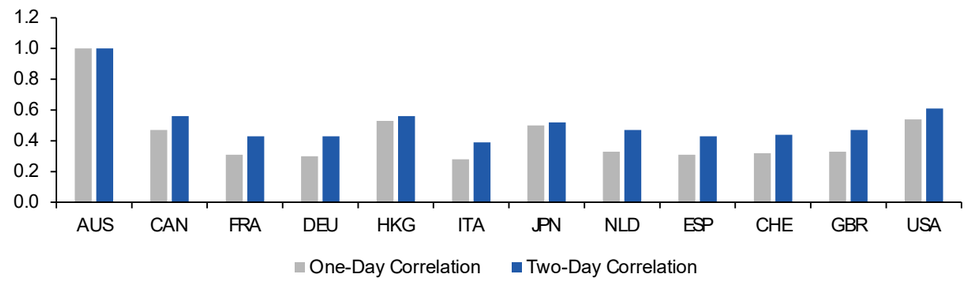
Outliers: Similar to more traditional techniques, the presence of a few outliers can have drastic effects on the fit of a model. As such, prior to running a ML algorithm, it is advisable to handle outliers via truncation or winsorization.

Look-Ahead Bias

Data revisions: Using point in time source or collection is preferred, otherwise use lag tests for robustness check.

Asynchronous returns across markets: This issue occurs for markets of different time zones. When using the end-of-day prices that are specified in different time zones, the correlations calculated from daily prices are erroneous.

Evidence of Some Return Asynchronicity of Australian Stocks vs Other Markets



Source: Bloomberg, as of 12/31/22.

Sin #6. Envy (chasing the latest fad): ML may be topical but it is not a silver bullet

As noted earlier, ML is not a silver bullet that will solve all the problems when searching for alpha. It can and will likely aid in the alpha-discovery process by making the process more systematic, more efficient and offering the ability to sift through and combine large-scale datasets. However, we believe it is most powerful when combined with a greater number of non-traditional datasets.

It is a fallacy to think that applying ML with the same set of signals discovered decades ago (which may have since been commoditized and much decayed) will radically improve the performance of models obtained with more traditional econometric techniques.

Sin #7. Pride: ML is powerful, but the market is king

As practitioners of quantitative investing, we believe in staying humble before the market. Time and time again, there have been well-publicized boom/bust episodes of geniuses and wonder boys, from the collapse of Long Term Capital Management to the subprime mortgage meltdown and the more recent flash crashes said to be caused by high-frequency traders. If there is any constant, it is the realization and understanding that the search for alpha is ongoing and requires constant innovation and critical thinking. Machine learning is simply a newer tool to aid in this endeavor, and fits best as another tool in the quantitative investors' toolkit.

Conclusion

Machine learning can be a powerful new tool in the search for alpha but its application requires special care. Unlike traditional econometric techniques and linear models, supervised ML techniques allow for more granular capture of nonlinearity and interactions among input features than a linear model can accomplish. With a linear model, in order to capture higher order interactions the model needs to specifically define interaction terms, which itself is both difficult to predetermine and often impacts the model's stability. ML algorithms are able to learn such nonlinearities and interactions without a need for predefinition. Such added power comes with the potential dangers we have articulated. When using machine learning, it is much easier to overfit the problem to a specific sample period and/or selection of models and parameter choices. Issues that exist with linear models (such as bad data practices and data snooping, etc.) may be exacerbated when applying machine learning techniques. Moreover, as practitioners of such powerful techniques, greater care must be given to interpreting the output from any ML model to avoid 'blackbox' results that are out of line with economic intuition. Despite all the potential pitfalls and caveats listed, when used appropriately, machine learning can be a powerful complement to more traditional modelling approaches in the search for new sources of alpha.

Glossary

Classifier: machine learning-speak for separating a distribution or set of discrete values into labelled classes based on predictive features, such that classes are more distinct from one and another, and can inform subsequent action/decision, e.g. long one class, short another other class.

Classification: the type of problem where one attempts to label the training data into two or more classes, and then predict the classes on the test data.

Regression/prediction: a problem where one attempts to provide a prediction on an output variable based on various input features. In the context of supervised learning, this also involves training a prediction model using training data, and then applying the prediction to the test data or out-of-sample data.

In-sample period (IS period): the period selected for training data to build the machine learning models

Out-of-sample period (OOS period): the period selected for applying the trained model to test performance on data that the model has not 'seen' before. This is a true test of the generalizability and performance of any supervised machine learning model.

Feature: a feature in supervised learning is generally equivalent to an independent variable in a linear model. It can be thought of as an input signal.

Overfit: the tendency to have high specificity to particular training period and combination of highly tuned models and parameter choices. Overfit leads to poor performance and lack of generalizability of the model out of sample.

Feature scaling: the form of data preprocessing where the input features are scaled so that no single feature or combination of features dominate the rest when distance-based methods are used, such as support vector machines (SVM).

Want to find out more?

Please contact our Consultant Relations and Business Development team:

Tel: +1 212 922 7777
Email: info@newtonim.com


@NewtonIM


Newton Investment Management

newtonim.com