FACTOR INVESTING - PART 1.3

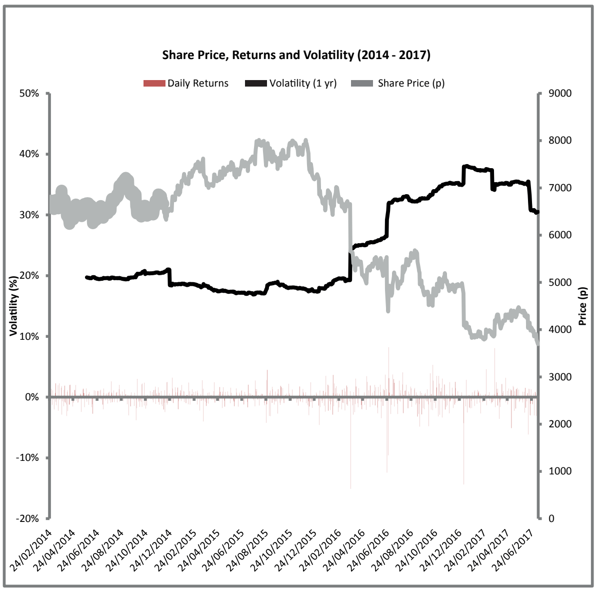
Using Machine Learning in Factor Investing

By Michael G. Kollo, PhD, Chief Research Strategist, Rosenberg Equities

All factor investment strategies bring a promise of a desired outcome, such as performance above a benchmark, higher dividends, or lower volatility through the business cycle. One example of a factor portfolio is the popular low volatility or SmartBeta product, which promises to create a portfolio that will exhibit lower price volatility by picking stocks that will, on average, share the low-volatility characteristic.

Forecasting future stock-price volatility can require forecasting future business success or failure using fundamental data, and some modelling skill. An example of the difficulties in forecasting future business success is illustrated by the performance of Next PLC, a UK-based retail company whose price volatility rose significantly over 2016 and 2017 as it faced sudden inflationary pressures as a result of sterling's weakness following the EU referendum, and sectoral shifts away from clothing expenditure.[1]

Figure 1: Example case of rising volatility



Source: Bloomberg, Rosenberg Equities, 30 June 2017

[1] No representation is made whether Next PLC was or is an investment recommendation. For illustrative purposes only.

A complimentary approach is to focus on statistical outcomes: will a stock have high (price) volatility in the future, and can we forecast jumps in price volatility?

Volatility jumps are highly non-linear events, and therefore require tools (empirical models) that can handle unusual non-linear outcomes. We therefore turn to 'neural nets', a branch of machine learning, to help us identify stocks whose prices will experience jumps in volatility (so-called 'torpedo' stocks).

Our aim will be to identify and avoid these 'torpedo' stocks and therefore improve the outcome of our low-volatility portfolio.

Methodology and setup

Our machine learning problem requires that we set up an objective at the beginning, which we define as 'y':

1. 'y' = 0 when the stock is low-volatility, the stock stays low volatility over the next year.

2. 'y' = 1 when the stock is low-volatility, but then stock becomes high volatility over the next year.

This provides the target variable (y) in our framework, which is binary. We now need to select variables that we think can forecast ahead of time if a stock will experience this volatility jump.

We need to be careful here: machine learning algorithms are thorough in their use of data and will try to use any inputs that we provide in just about any permutation, and in non-linear ways. Giving too many input data options will improve the algorithm's power, but are more likely to result in over-fitting[2].

We used a collection of input variables that we believe could affect a stock's volatility, including measures of the stocks' historic beta, volatility, distress risk, dividend risk, and excessive valuation.

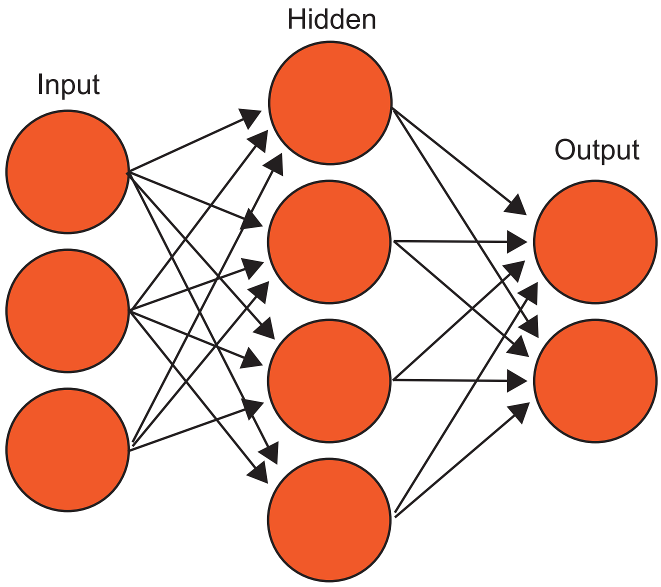
The fundamental motivation behind these variables has to be clear in order for us to understand the output from the model. By using dividend risk, for example, as an input into the model we are suggesting that the fundamental risks faced by firms that are at risk of non-payment of their dividends, are potentially the same risks that a firm whose volatility is likely to rise. Unlike a typical regression, however, we are not saying how dividend risk is related to future rises in volatility, and letting the machine learning algorithm establish this relationship.

We chose two different machine learning algorithms to use: neural nets and random forests.

Using neural nets

For neural nets, we allowed for a single hidden layer as this provided a strong fit given our forecast[3] using a feed-forward methodology. This is a fairly standard setup for a machine-learning algorithm and strikes a balance between interpretability and model power. The output layer here is a probability that the company will be a 'torpedo' stock.

Figure 2: Neural nets



Source: Rosenberg Equities

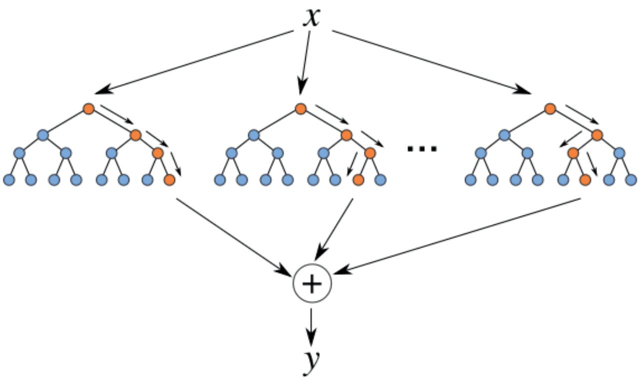
Using random forests

For our second model, we used a random forest. This creates 'trees', which are like conditions. For example, if a company has had falling earnings and has cut its dividends, then its probability of having a spike in volatility is x%. The graphic below shows the different 'trees', each with a different number of branches (conditions) and a different structure of those conditions (e.g. different questions, and order of questions being asked). Each tree takes every company and 'passes' it through its branches, returning a probability that the stock is likely to be spiking in volatility.

An analogy is a game of 21 questions, where the contestant can ask any question that requires an answer of 'yes' or 'no' to try to work out what the games-master is thinking about. Now imagine there are thousands of contestants, and they can only ask five questions each, after which they must provide a probabilistic guess as to what the game-master is thinking about.

[2] Over-fitting is a term that refers to the case where a statistical technique, like a regression or a neural net, fits statistical relationships very closely within a sample of data, but it's conclusions are not applicable to more general dataset. [3] More hidden layers can allow for more non-linearity in the use of the input data, and systems like deep learning can use many more than we have used here. As the systems grow in layers and improve in power, they also become increasing opaque and difficult to interpret.

Figure 3: Random forests



Source: Rosenberg Equities

At the end, the algorithm combines the weighted average of all the estimates (guesses) across all the trees, and provides a single probability that the stock will be a 'torpedo' in the future.

Results and conclusions

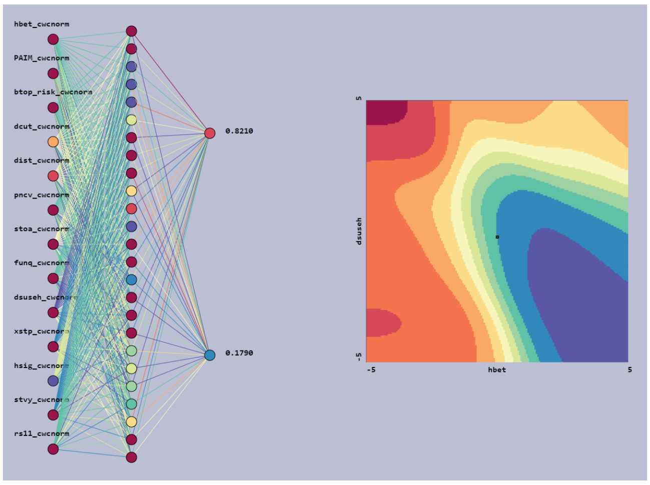
We used several machine learning techniques to try to forecast where a stock is likely to have an extreme return reaction. Our analysis using machine learning techniques proved to be a generous improvement over a simpler, linear model. In backtests, the naïve model proved to be 73% accurate in predicting extreme return reactions, but machine learning techniques managed to improve this to 86%, or a 13% improvement, which is sizable for this kind of model.[4]

This could enable us to remove particularly risky stocks from our low-volatility strategy before they experienced a significant jump in volatility.

Before moving further, we spent some time interpreting the results, especially from the neural nets. These methodologies often lend themselves to 'black box' type criticisms, so we built some data visualisations (see Figure 4) and re-estimated the model using more conventional linear techniques to see how the non-linearities of the neural nets behaved, also known as creating 'white-boxes'.[5]

The data visualisation helps the research track how a given input would change the expected output of a model, through the non-linear system that is a neural net.

Figure 4: Data Visualisation



Source: Rosenberg Equities

The focus on methodology in this piece is to highlight how more modern, data-mining techniques can augment a more traditional, and fundamentally-based quantitative modelling approach. We would argue that adopting these techniques can lead to new insights if there is no established economic framework for statistical relationships and the researcher is happy to let the data determine the statistical (model) relationship.

In this case, transparency becomes critical for the deployment and maintenance of these techniques. While the models above can be readily accessible through publicly available code libraries, their interpretation and deployment to manage wealth and retirement assets has to be carefully considered, and anticipated, before they are put into practice.

[4] Please note that the referenced machine learning forecasting model is not used in production and is for academic purposes only. No representation is made about current and future Rosenberg Equities production models or investment strategies and products. [5] www.sas.com, 'Surrogate models'.

GM170029_GLOBAL 08/2017 Design & Production: AXA IM London Corporate Communications

INVESTMENT MANAGERS