MACHINE LEARNING IN QUANT INVESTING: REVOLUTION OR EVOLUTION?

APRIL 2019

- **For quantitative investors, machine learning (ML) represents an important expansion of the analytical toolkit, providing substantial new flexibility. It has application throughout the investment process.**
- **ML represents an evolution rather than a revolution. For the foreseeable future, finance domain knowledge will remain essential in its application.**
- **ML's increased flexibility demands the embrace of a disciplined research process. For asset owners, the combination of additional flexibility plus the opacity of some ML methods increases the challenge of distinguishing true added value from overfit, data-mined results.**

As a society, we've become preoccupied with "the rise of machines"; the topic taps into some of the greatest hopes and fears of our digital age. In the financial world, though, the breathless and superficial nature of the broader cultural dialogue around it has clouded understanding as to what actual impact artificial intelligence, machine learning, and the explosion of data will have over the foreseeable future.

In this note, we ground the discussion by laying out our view as to how ML will reshape quantitative investing over the next few years. We contextualize ML approaches relative to traditional methods, considering their salient characteristics, principal advantages, and scope of application as well as challenges to their further integration. To provide a real-world sense of what it means to apply ML in quant investing, we present a brief case study involving the enhancement of a stock-selection signal.

ML significantly expands the analytical toolkit for sophisticated quantitative investors. We believe that it already is generating material additional value and will deliver more.

In summary, ML significantly expands the analytical toolkit for sophisticated quantitative investors. We believe that it already is generating material additional value and will deliver more. ML has application throughout the investment process, including non-linear forecasting, revealing structure in complex data, and derivation of quantitative metrics from qualitative information. But the application of ML to investing is more challenging than in many other contexts. As a result, financial domain knowledge and a rigorous and explicit research process will be essential for sustained success. Awareness of these considerations will help asset owners to distinguish true value-add from naive data mining.

ML IN CONTEXT

Artificial intelligence represents a broad goal: development of machines that make decisions at least as well as humans. Machine learning is an approach to achieving that objective: data-driven computer applications that train themselves to complete complex tasks without explicit instruction. (See the Appendix for a primer on three popular machine learning approaches.)

AI and ML concepts and methods are not new, and attempts to apply them to investing date back decades. In the 1990s, for example, numerous academic articles extolled the promise of neural networks to forecast stocks, currencies, and financial events.[1]

Today, the intense focus on ML reflects the confluence of several trends. First, many ML methods are computationally intensive, and computing costs have dropped dramatically. Commoditization of cloud computing has made thousands of processors available on-demand. Second, ML algorithms are now freely available in robust and well-documented open-source packages, dramatically lowering cost, time, and knowledge barriers to implementation. Third,

[1] As just a few examples: Jingtao Yao, and Chew Lim Tam, "Neural Networks for Technical Analysis: A Study on KLCI," *International Journal of Theoretical and Applied Finance* 2, no. 2 (1999); Mary Malliaris and Linda Saichenberger, "Using Neural Networks to Forecast the S&P 100 Implied Volatility," *Neurocomputing* 10, no. 2 (1996): 183-195; Christian Haefke and Christian Helmenstein, "Forecasting Austrian IPOs: An Application of Linear and Neural Network Error-Correction Models," *Journal of Forecasting* 15, no. 3 (1996): 237-251.

the explosion of big data facilitates ML approaches that involve algorithms with hundreds or thousands of parameters to estimate, and that therefore require enormous quantities of data to train. Finally, there have been algorithmic breakthroughs, including in language processing and the development of better behaved and more computationally efficient neural networks.

In the context of quant investing, interest in ML reflects desire for greater modeling flexibility in two broad respects. The first is non-linear prediction. ML algorithms are designed to infer the relationship between data attributes and a variable of interest. In contrast, ubiquitous linear regressions impose an assumption that the form of the relationship is a straight line. Although linear models have endured for a host of good reasons, including simplicity, transparency, robustness, and modest data requirements, in many investing contexts the assumption of linearity is unfounded, and neither theory nor intuition suggests a particular alternative. Examples include forecasting probabilities of market events or economic regime shifts.[2]

A second motivation for ML is to reveal hidden structure in complex, large data sets. In identifying a company's peer group, for example, ML algorithms may help to illuminate the economic relationships among hundreds of firms, simultaneously accounting for numerous attributes. ML can also help to expose interactions between predictive variables that may vary with environmental conditions. ML-based textual analysis can derive quantitative metrics from qualitative information, sentiment analysis being a prominent example.

For quantitative investment managers, it opens new avenues of research that have genuinely exciting potential. Yet their enduringly successful exploitation will demand considerable effort and firm commitment to a disciplined research process.

ML techniques represent a significant expansion of the quantitative investor's toolkit, but they're not qualitatively distinct from traditional statistical methods. Linear regression, for example, can be expressed as a simple neural network. Elements of ML algorithms to identify the most valuable predictive variables and to avoid overfitting have analogues in conventional statistical methods. As a result, trying to identify a meaningful bright-line demarking traditional approaches form ML would be a fruitless exercise. Fortunately, it's also not important.

Of great consequence, however, are the implications of easy access to more flexible but also more opaque analytical tools. For quantitative investment managers, it opens new avenues of research that have genuinely exciting potential. Yet their enduringly successful exploitation will demand considerable effort and firm commitment to a disciplined research process. And while ML methods require new skill sets, domain knowledge from finance will remain crucial to beneficial research.

For asset owners, the increased modeling flexibility will exacerbate challenges in manager and strategy selection. Finance is already rife with sloppy data mining, and access to a more expansive toolkit only makes it easier. Further, the opacity of some ML approaches will complicate performance attribution even when such techniques are implemented appropriately. As a "canary in the coal mine" regarding such challenges, the data science literature is already replete with claims of unrealistic market forecasting ability. In this evolving environment, a working knowledge of ML techniques and research best practices will help to protect asset owners.

AN ML CASE STUDY

To demonstrate the application of ML to quant investing, we present a case study in the enhancement of a stock-selection signal. In a 2002 paper, Joseph Piotroski proposed an approach to improve valuation factors for bottom-up stock picking.[3] The premise was that conditioning on indicators of a company's fundamental strength can help to distinguish attractively undervalued stocks from value traps. Today, the general concept of employing quality variables to help identify underpriced fundamentals is well established. Nevertheless, Piotroski's framework is worth revisiting as an intuitive, real-world setting in which to highlight motivations for applying ML and to elucidate the ML research process.

To test his hypothesis, Piotroski formed a composite indicator, which he called the "F-score." (Figure 1) This was constructed from nine individual financial statement metrics chosen to reflect 1) profitability, 2) leverage, liquidity, and source of funding, and 3) operating efficiency. To put the data items on an equal footing so that they could be combined, Piotroski took a sensible and robust approach. For each attribute, he created a binary variable, assigning a 1 if the attribute were positive and 0 if negative. For each stock on each date, he then summed the binary variables to create the baseline F-score signal.[4] Exemplifying the signal's intuition, Figure 2 shows that Sears Holdings' F-score captured a protracted deterioration in the company's financial strength in the years prior to its bankruptcy.

[2] Probabilities are bounded by 0 and 1. Linear dependence on a predictor variable that can take on any value would violate those bounds.

[3] Joseph R. Piotroski, Value Investing: The Use of Historical Financial Statement Information to Separate Winners from Losers, January 2002.

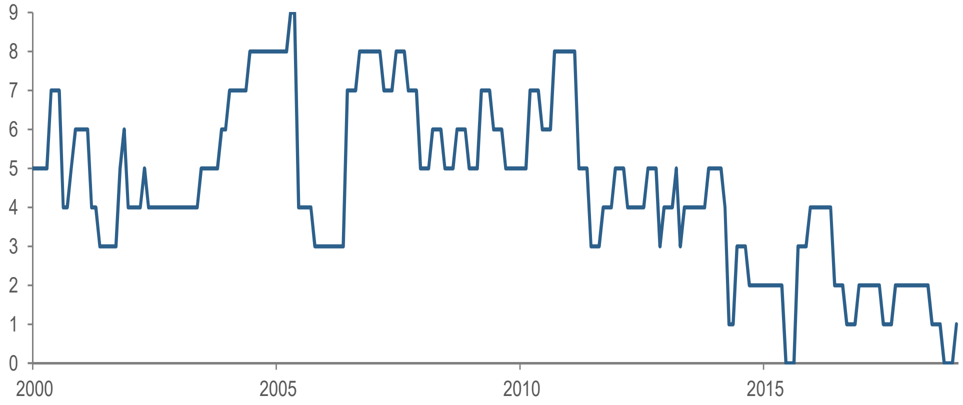
[4] For our analysis, we subject the his baseline F-score implementation and an ML to various proprietary adjustments.

FIGURE 1: CREATING THE “F-SCORE”

CATEGORY	CHARACTERISTIC	ASSIGNED VALUE IF POSITIVE/NEGATIVE
Profitability	RETURN ON ASSETS	1 / 0
	CASHFLOW FROM OPERATIONS	1 / 0
	Δ LEVERAGE	1 / 0
	ACCRUALS (ROA - CFO)	0 / 1
Leverage, Liquidity, Source of Funds	Δ LEVERAGE	0 / 1
	Δ CURRENT RATIO	1 / 0
	NET EQUITY ISSUANCE (PAST YEAR)	0 / 1
Operating Efficiency	Δ OPERATING PROFIT MARGIN	1 / 0
	Δ ASSET TURNOVER	1 / 0
	Composite F-Score:	Sum of binary values

For illustrative purposes only. Not a recommendation to buy or sell a specific security. Source: Piotroski 2002.

FIGURE 2: SEARS HOLDINGS’ F-SCORE



Source: Acadian. For illustrative purposes only. F-Score is calculated based on the methodology described in “An ML Case Study” as applied to Sears Holding Corp. The selected stock is intended as an example of a process and is not a recommendation to buy or sell any specific security. Past results are not indicative of future results. Every investment program has an opportunity for loss as well as profit.

WHY ML?

This is an appealing setting in which to apply ML for several reasons. First, we have intuition that the selected financial characteristics may help return forecasting capabilities; this isn’t an uninformed fishing expedition in a low signal-to-noise-ratio environment. Second, although reducing the individual financial attributes to binary variables and then summing them is intuitive and transparent, we have no reason to think that it fully exploits their predictive value. Third, there is no theory or intuition to suggest how we should combine the individual quality metrics, e.g., whether to accord each equal weight or whether the predictors interact. Fourth, we don’t know what the relationships between F-score components and future returns look like; they may be unequal and/or highly non-linear.

SPECIFYING AN ML APPROACH

To develop and evaluate an ML-based alternative signal, we follow a research process that has three interrelated elements:

- **Data preparation:** We must decide what data inputs to feed into the machine. This involves: 1) selection of candidate predictive variables (“features”), 2) data preprocessing, i.e., cleaning and integration, and 3) transformations of the raw data, e.g., market or peer group adjustments and normalizations. For this case study, we work with Piotroski’s selected attributes.[5] Preprocessing includes treatment of missing and outlier predictor values, exclusion of financials, and a filter for tradability. We also market-adjust stock returns.

[5] Subject to a number of adaptations. Please contact us for details.

- **Algorithm selection:** For this exercise, we choose an approach called "random forest." In data science circles, this method is considered a solid, general purpose algorithm. It's relatively easy to implement, generally effective and robust, and reasonably transparent.

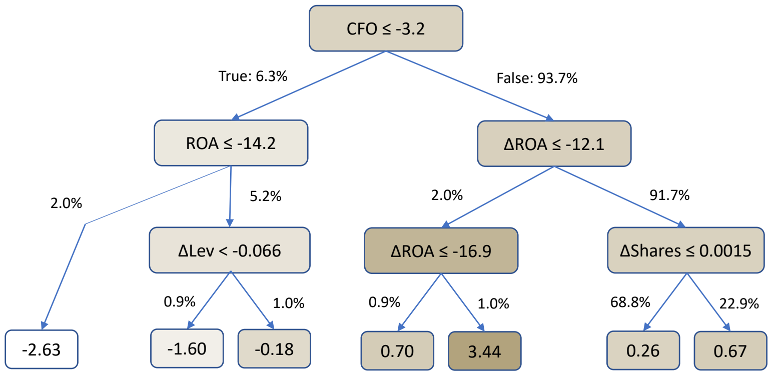
  By way of high-level description, the random forest algorithm is based on generating many "decision trees" fitted to historical data. An individual decision tree, an illustrative example of which is shown in Figure 3, represents a sequence of simple forecasting rules each of which, in the case study, relates an individual accounting variable to future returns. Specifically, at each decision point in the tree, called a node, the algorithm subdivides the training data into two groups of observations based on the value of one attribute. The algorithm selects the attribute on which to split the sample so as to most effectively discriminate between higher and lower future returns in the training sample. Not every variable may appear in a tree; i.e., the algorithm may find that certain attributes don't have predictive power, and a variable may appear several times, associating its different levels with a range of predicted returns. (See the Appendix for further discussion of decision trees.)

  In a "random forest" implementation, we generate many trees, fitting each one from a random subset of the historical data. The purpose is to reduce the risk of overfitting associated with estimating just a single tree, which might be driven by spurious relationships in the full sample. We generate the final return forecast for new, out-of-sample observations by averaging return forecasts across all the fitted trees. For the case study, we estimated between 300 and 3,000 trees per forest in various candidate implementations.

- **Data sample management:** Implementing an ML algorithm typically involves subdividing the historical dataset into three segments: 1) a "training" sample to estimate algorithm parameters, 2) a "validation" sample to tune the algorithm and/or select among several variants, and 3) "out-of-sample" historical data to evaluate the approach.

  For this case study, we used the validation step to choose the maximum number of nodes in the decision tree (tree "depth") and other criteria to limit the branching process. The purpose of these restrictions is to avoid overfitting and reduce sensitivity to outliers. We trained and validated the random forest algorithm based on data from 2000-2012, and we ran it out-of-sample from 2013-2018. We restricted tree depth to 3 levels.

FIGURE 3: SAMPLE DECISION TREE



Source: Acadian. Diagram illustrates the results of fitting a single tree in a random forest implementation as described in "An ML Case Study." For illustrative purposes only.

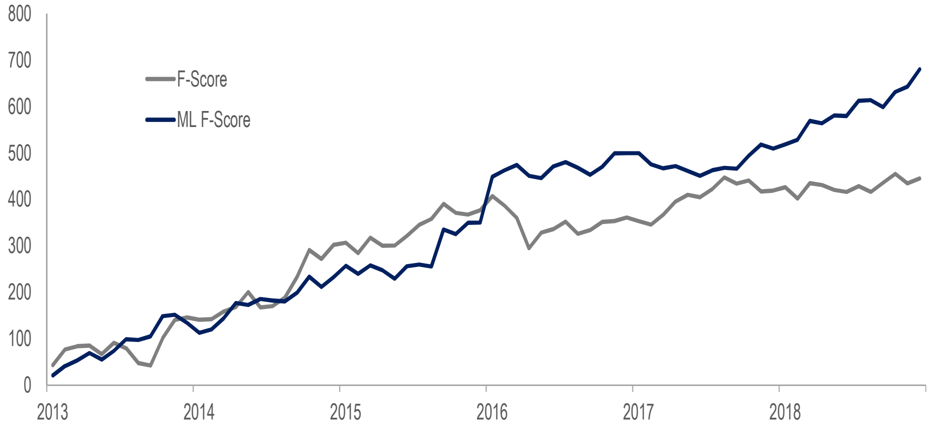
RESULTS

Figure 4 compares the predictive efficacy of the baseline and ML-based F-score signals in a broad universe of developed market stocks from 2013-2018. The table shows that the ML-based version displays greater efficacy in predicting stocks' 1-month-ahead returns, as evidenced by a higher slope coefficient (9.4bp/month versus 6.2bp/month). Its predictive ability is highly statistically significant (t-stat of 3.7 versus 2.3 for the baseline version).[6]

[6] Each coefficient represents a time series average of cross-sectional regression slopes, i.e., a Fama-Macbeth analysis. The ML coefficient of 9.4 indicates that an ML_F-score that is one standard deviation above (below) the cross-sectional average is associated with an increase (decrease) in 1-month ahead returns of 9.4 bp. These results are not investible. The analysis does not include transaction costs or trading frictions.

FIGURE 4: EFFICACY COMPARISON – BASELINE F-SCORE VS. ML F-SCORE

Developed market companies, Jan. 2013 – Dec. 2018



Source: Acadian. For illustrative purposes only. Chart shows sum of slope coefficients from monthly cross sectional regressions of one month ahead stock returns on vanilla and machine learning-based F-score formulations using Acadian's developed market universe restricted to non-financial companies market capitalizations above $100mm USD. Regression results do not reflect trading, borrow costs, and other implementation frictions. For these and other reasons, they do not reflect performance of an investible strategy nor are they indicative of actual future results. Every investment program has the opportunity for loss as well as profit.

Analysis of individual trees within the random forest reveals that the forecasts are primarily driven by cash flow from operations followed by the change in return on assets, whereas change in gross margin, current ratio and leverage have limited influence. This variation contrasts meaningfully with the baseline F-score's implicit equal-weighting scheme across the nine attributes. Interestingly, even though the ML formulation is materially driven by cashflow from operations, its direct correlation is modest. This likely reflects the non-linear nature of decision trees and, consequently, the random forest itself. As well, the ML F-score signal has a fairly long workout horizon and high serial correlation, which suggest that it wouldn't induce excessive turnover.

OPPORTUNITIES AND CHALLENGES

ML's added flexibility has application throughout the investment process. Motivations for the case study help to explain why certain contexts make for promising use-cases, and special challenges in applying ML to a market context help to explain why some uses-cases hold less appeal.

Generation of new quant data inputs: ML is already extensively used to create alternative data sets, with language processing of news, filings, and social media among prominent examples. Contemporary applications include sentiment measurement, event detection, and evaluation of qualitative relationships and risks. Inferring context in unstructured, long-form text remains a challenge, and this use-case is helping to drive development of deep learning methods. Data imputation is another area well-suited to ML. Examples in the context of company fundamentals would include detecting errors and estimating missing or late-reported data items. ML techniques may be able to exploit underlying accounting logic or management behaviors that generate subtle, predictable patterns in the data.

Returns forecasting: ML has broad potential in forecasting. Examples include enhancement of technical signals (i.e., signals related to historical patterns in returns), where flexible ML-based prediction can better exploit richness in the data than prior generations of indicators that traded off informational nuance for robustness. As well, ML is already incorporated into sophisticated cross-asset technical signals to help identify subtle linkages among securities.

But history is littered with forgotten academic articles and dead weblinks that chronicle persistently unrealistic expectations among methodologically sophisticated but market-naive data miners. When working in financial markets, ML researchers must recognize that the signal to noise ratio is lower and the data-generating environment more unstable than in many other contexts in which such methods are being applied.

Risk forecasting: There is no consensus as to the form of the underlying process driving financial market volatility despite decades of empirical research. Innumerable complex statistical models have been proposed to describe and predict its dynamics, including "jumps" and "regime shifts." In this context, the gap between past statistical approaches and ML seems particularly indistinct. At the portfolio level, evaluating and forecasting risk depends on capturing the comovement among the component instruments. The task usually involves some form of "dimensional reduction," i.e., identifying common drivers of returns, a natural context for the application of "unsupervised" learning algorithms.[7] ML also has application in predicting risk events, a setting that motivates non-linearity and where training datasets of past events usually are easy to construct.

Factor weighting: Shifting factor exposures as macroeconomic and risk conditions evolve is oft-cited as an area of potential ML benefit. The premise being that flexible analytical approaches may be able to detect market regime shifts and adapt weightings appropriately. This use-case presents challenges, however. Among them, we may not have observed enough relevant historical transitions between states to train data-hungry algorithms, and investors may be particularly uncomfortable with reduced transparency during periods of market stress.

Implementation: ML has achieved comparatively wide use in high frequency trading in part because there is an enormous amount of intraday data that can be used to train execution algorithms. As well, in an era of market fragmentation, optimal choice of venue, method, and timing likely depends on many conditioning factors, including expected volumes and order flow, upcoming events, and sentiment. Optimizing over this large and complex set of inputs is a task well-suited to inductive ML processes.

BEST PRACTICES

In identifying promising ML applications, the nature of the variable being forecasted and availability of historical data emerge as consistent themes. In many investing contexts, the low signal-to-noise ratio and relative scarcity of training data imply that blindly unleashed ML algorithms are prone to overfitting, i.e., picking up on spurious relationships in training data that are of no use in forecasting out-of-sample.

While finance has long been rife with sloppy and abusive data mining, we expect that ML's flexibility and opacity will increase its prevalence. For asset owners this will exacerbate challenges in distinguishing value-added approaches from artifacts of backtesting and slick presentation.

This concern informs a set of best practices to watch for in ML's application to investing. It should be guided by understanding of a specific problem, and algorithms must be carefully controlled and validated to have out-of-sample efficacy. ML is generally applied for data mining rather than hypothesis testing, i.e., with the objective of maximizing predictive power rather than providing insight. With that frame of reference in mind, the research process outlined in the case study is expressly designed to provide "guardrails" that reduce the risk of overfitting.

Investment expertise will remain crucial to beneficial ML research. Decisions regarding data preparation, algorithm selection, and data sample management are interdependent, and investment domain knowledge informs them. As one example, judicious data preparation reduces the amount of data required for training and validation. Relying on ML to derive a peer-relative transformation likely would require an extremely flexible and data intensive algorithm and increase the risk of overfitting.

Because overfitting is such a prominent concern, overreaction to that risk may lead to underfitting. Techniques designed to reduce fitting noise can result in losing a signal altogether. While need for transparency should influence the decision to apply ML as well as algorithm selection, concern about complexity and opacity may lead to excessive caution in ML's application.

While understanding the drivers of algorithm behavior can be a challenge, even in the context of complex ML methods we may nevertheless have means to gain insight. For a decision tree, inspecting early branches reveals which features the algorithm deems most predictively valuable, and, as we demonstrated in the case study, we can derive summary statistics of relative feature importance. For a neural network, the first "hidden layer" of nodes may serve as a feature-identification step, filtering and combining raw inputs into more refined items that are then passed deeper into the system. We may also be able to assess an algorithm's likely response to circumstances of special interest by feeding it contrived data.

As well, many longstanding research best practices remain applicable to ML contexts. These include comparing an ML-based forecaster's behavior to simpler, more transparent signals, analyzing whether it loads on known risk factors, and examining whether its serial correlation and workout horizon suggest that it will induce portfolio instability and high turnover. Integration of ML doesn't necessarily turn the investment process into an impenetrable black box.

CONCLUSION: THE OUTLOOK

ML is already a part of sophisticated quants' investing toolkit. It offers valuable modeling flexibility and has application throughout the investment process. We view the integration of ML into investing as an evolution rather than a revolution. The techniques aren't qualitatively distinct from traditional methods, and we expect that the quantitative investment process will remain recognizable in the foreseeable future, even as the research tools that inform its components evolve. Markets make for an especially challenging environment in which to apply ML, and beneficial ML-based research calls for process discipline and finance domain knowledge. For asset owners, more flexible but more opaque analytical tools will exacerbate challenges in distinguishing strategies that add value from data artifact. Working knowledge of ML approaches and ML research best practices offer a valuable defense.

[7] In "unsupervised learning" no specific target variable is predicted. Often the goal is to group or infer patterns in input data.

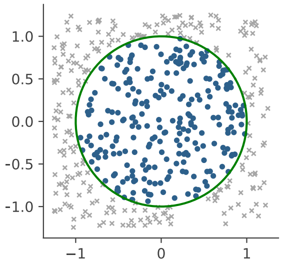
APPENDIX

ML METHODS: A PRIMER FOR INVESTING PRACTITIONERS

This Appendix provides insight into three of the most prevalent machine learning approaches, decision trees, support vector machines, and neural networks. We make use of an illustrative "binary classification problem" where the goal is to categorize data observations into two groups. In medicine, problems of this form would include identifying tissue samples as cancerous or non-cancerous; in investing, examples would include seeking to predict whether a market index may rise or fall.

Specifically, we've selected the problem of distinguishing datapoints that are inside and outside of a circle that has a radius of 1, as depicted in Figure A1. The dataset consists of observations that are defined by random x and y values. Points within the circle are blue, and points outside of it are grey. In ML jargon, the points' x and y coordinates are "features," which simply means that they're interesting, measurable characteristics of the data under consideration. In a stock picking context, features might include financial ratios.

FIGURE A1: THE CIRCLE DATA SET



Source: Acadian. For illustrative purposes only.

DECISION TREES

Decision trees ought to seem familiar: the game of 20 questions represents a common type of decision tree. Figure A2 is a fun illustrative example. Participants ask a sequence of yes/no questions, each of which should depend on all the prior answers. Participants continue to ask questions until they have enough information to make an educated guess at (i.e., to estimate) the solution or until they run out of questions. This is precisely how a decision tree works for classification: it is a flow chart of questions to "ask" a data point in order to determine to which class it belongs.

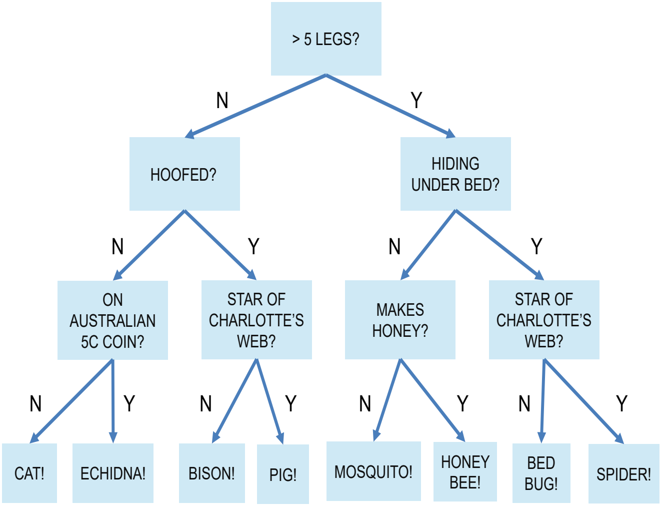
Figure A3 illustrates the workings of a decision tree in the context of the circle problem. If we could only pose one question, the best we could do in guessing whether an observation is inside or outside the circle, i.e., blue or grey, might be to check whether its x coordinate is larger than some number close to 1. That's because any datapoint whose x coordinate is greater than 1 must lie outside the circle. This is evident in Figure A1: no points to the right of 1 are blue.

The leftmost plot in Figure A3 illustrates the result of applying this rule. Points guessed to be inside the circle are represented by the yellow shaded area. Points guessed to lie outside of the circle lie in the unshaded region.

When allowed to ask only one question, the algorithm's predictive accuracy is terrible: the yellow shaded area looks nothing like a circle.

But by asking successively more questions about the data, looking from left to right, the yellow-shaded approximation improves. At a depth of 10, on the far right, it looks quite good.

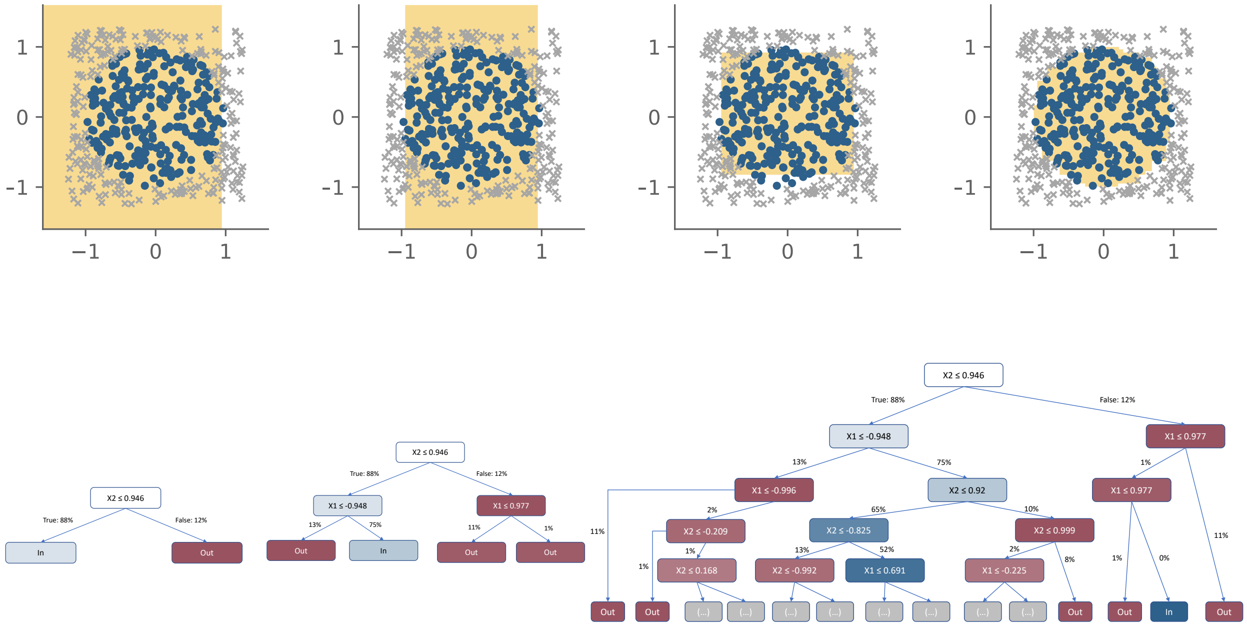
FIGURE A2: A HYPOTHETICAL "20-QUESTIONS" DECISION TREE FOR GUESSING AN ANIMAL



Source: Acadian. For illustrative purposes only.

FIGURE A3: DECISION TREE ACCURACY AS A FUNCTION OF TREE "DEPTH"

From left to right, depth = 1, 2, 4, 10



Charts in top row reflect decision boundary for decision trees of various depths applied to the circle dataset. Bottom row displays the first several levels of the respective decision tree. Source: Acadian. For illustrative purposes only.

In general, decision trees are constructed in this step-by-step manner. The algorithm starts by identifying an individual feature and an associated value that, in isolation from all the other features, most accurately classifies the datapoints. The process is then repeated within each of the two resulting subsamples, and so on, until the tree reaches a preset limit on its depth or some other stopping criterion set by the researcher.

This stepwise approach to tree construction is termed a "greedy" algorithm. Greedy is technical jargon indicating that at each stage of the tree the algorithm selects a feature and a value based only on the immediate improvement in classification without considering implications further down the tree. We would expect superior classification ability from a holistic approach that simultaneously considers the combined effect of all subsequent feature and value selections. Unfortunately, such algorithms are often computationally infeasible.

This step-by-step tree construction approach has a visibly interesting consequence in the context of the circle problem. Because the algorithm only considers one feature at a time, either x or y, every line drawn must be parallel to either the y or the x-axis. This is easy to see in the three left-most panels of Figure A3. As we allow the tree to have greater depth, i.e., as we ask more and more questions, the boundary appears more and more circular. But squinting at the image reveals that the circle approximation still consists only of lines parallel to either the x- and y-axis. The broader point is that understanding how an algorithm works can help to provide a sense of the complexity required to accomplish a particular task.

SUPPORT VECTOR MACHINES

While Support Vector Machines (SVM) have a complex name, they are an intuitive method of classifying data. The circle problem is ideally suited to illustrating how they work.

In this context, the key question is whether we can distinguish points inside and outside of a circle with a straight line. (This property is known as "linear separability.") Unfortunately, this doesn't look possible even from just a cursory glance at Figure A1. Why do we care? Because it's easier to work with and to fit straight lines than curves. This concept becomes increasingly important as the complexity of the data increases, i.e., as we have more and more features.

It turns out, though, that by transforming the information in each data point's x and y values, we might be able to do it. Specifically, we could create two new features, $x^2$and y^2. If we then re-chart the datapoints on the basis of those transformed (squared) values we get Figure A4. It's clear in this picture that the blue and grey points are now separable by the green straight line. [8]

[8] Readers may recognize $r^2 = x^2 + y^2$ as the equation of a circle centered at (0,0) with a radius of r. The green line then represents $x^2 + y^2=1$, precisely capturing the circle dataset.

FIGURE A4: THE CIRCLE DATA SET TRANSFORMED

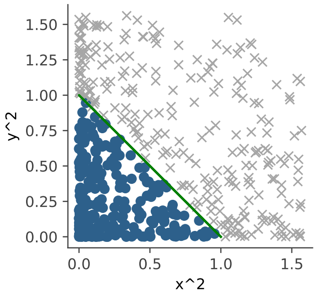


Chart shows the data in a transformed coordinate system in which the data is readily separated by a straight line. Source: Acadian. For illustrative purposes only.

To generalize from the circle example, SVMs find transformations of data, specifically new "coordinate systems," that make classifying the data easier. Through resource efficient computer implementations, they can try hundreds of alternatives, starting with small powers of the original features, squares and cubes, and progressively larger and larger ones. The algorithm will then choose the transformation and classification rule that best separates the two groups of data.

NEURAL NETWORKS

Artificial neural networks (commonly just "neural networks") are the most flexible of the three methods. They can approximate intricate functional relationships in data. The cost of this flexibility is algorithmic and computational complexity and opacity.

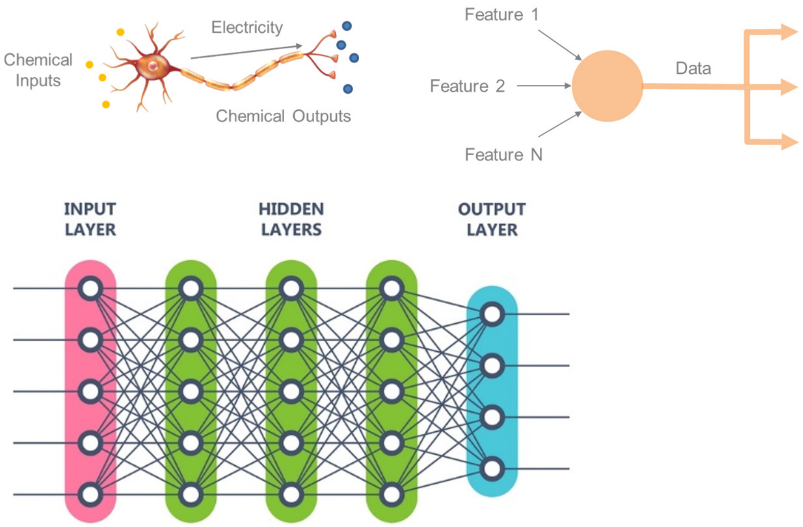
Neural networks are based on the behavior of neurons in the human brain. Roughly speaking, neurons receive and transmit chemical stimuli. They respond if the input stimulus exceeds a threshold level, at which point they transmit other chemicals to the next neuron in the chain. Artificial neurons seek to replicate this behavior as shown in the top row of Figure A5.

Neural networks are comprised of artificial neurons much like brains are comprised of biological neurons. Figure A5 displays a generic schematic in which each white circle is such an artificial neuron. The researcher specifies what the nodes do and how they are arranged based on the nature of the problem at hand.

Instead of chemicals, artificial neural networks filter, process, and transform data. Features from the input dataset are passed into the first layer of nodes. If the data item received at a particular input node exceeds a certain threshold value, then the node will pass that information along to a node or several nodes in the next layer. A node may transform the data that it receives. It may combine information from several nodes in the prior layer, applying different weights to different sources. A neural network composed of several layers and a large number of nodes may extensively transform the data, abstracting it considerably away from the input information. This is how "deep learning" neural networks make complex decisions.

To implement a neural network, the researcher must specify its architecture, which is an engineering problem. The researcher sets 1) the number of layers of nodes, 2) which nodes are connected to each other, and 3) what transformation function is embedded in each node. In principle, a linear regression could be implemented as a simple neural network: it would consist of one layer, one node, and a regression line as the function (intercept + slope*input feature). In the training stage, the algorithm would estimate the regression intercept and slope to minimize the prediction error.

FIGURE A5: NEURAL NETWORK



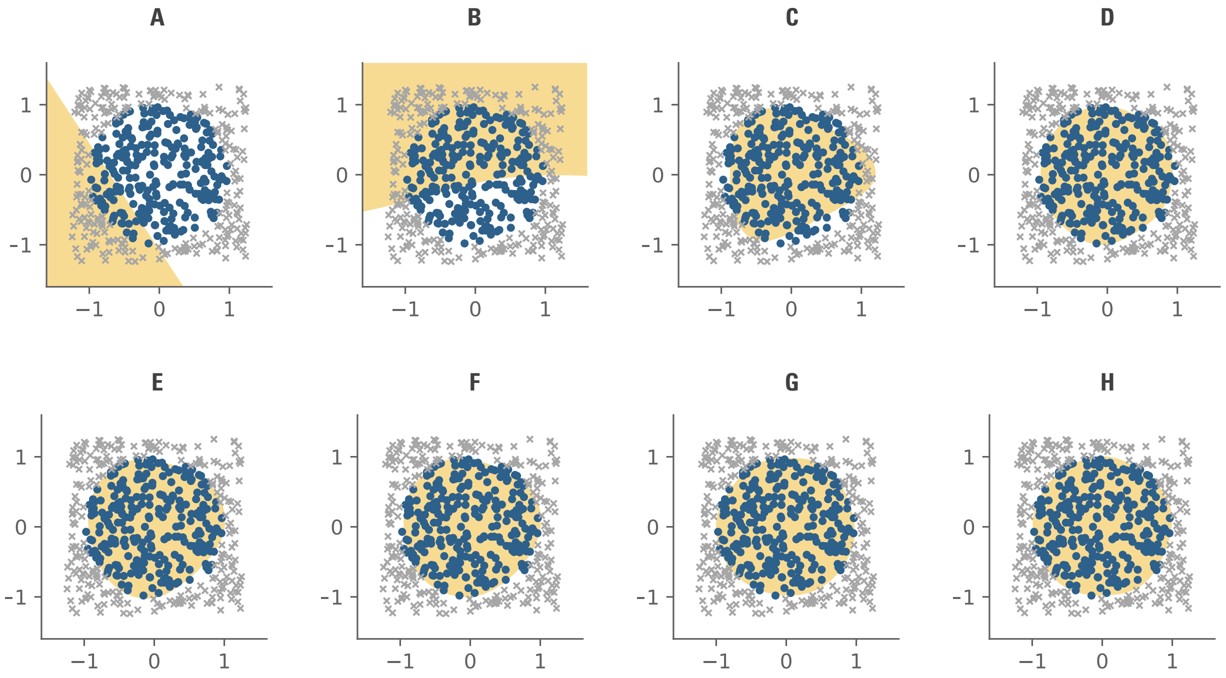
For illustrative purposes only. Biological neuron and neural network diagram copyright Dreamstime.

Generally speaking, though, most networks employ functions that have an "on/off" property, resembling the firing of biological neuron. Perhaps for the largest input values the function would pass along a 1, for most other values a 0, and for a small range of values some number(s) between 0 and 1. Image recognition might provide an intuitive functional context: a neural network could take in a set of input features, e.g., pixels of from photos, and through the training process discriminate shading or coloration in which pixel locations, or combinations thereof, are predictively valuable in recognizing image types.

Returning to the circle problem, Figure A6 compares classification ability of multiple neural network architectures where the functions at each node have the on/off property. Several architectures with modest numbers of parameters, i.e., only a few layers and nodes, do quite well at approximating the circle.

FIGURE A6: CLASSIFICATION ABILITY OF SEVERAL NEURAL NETWORK ARCHITECTURES

A: 1 hidden layer with 1 neuron; B: 1 hidden layer with 2 neurons; C: 1 hidden layer with 4 neurons; D: 1 hidden layer with 8 neurons; E: 2 hidden layers with 4 neurons each; F: 2 hidden layers with 8 neurons each; G: 3 hidden layers with 4 neurons each; H: 3 hidden layers with 8 neurons each



Charts display the decision boundary for various neural network architectures applied to the circle datset. Source: Acadian. For illustrative purposes only.

But even in this simple context, the opacity of these neural networks is problematic in trying to understand what they're doing. Knowing the function of an individual node won't provide much insight. The transparent flow of a decision tree or the nature of the data transformation applied by an SVM doesn't have a clear analogue. What's more, the number of parameters to estimate will grow rapidly as network architecture becomes richer, potentially consuming substantial data and computational expense in the training process.

GENERAL LEGAL DISCLAIMER

Acadian provides this material as a general overview of the firm, our processes and our investment capabilities. It has been provided for informational purposes only. It does not constitute or form part of any offer to issue or sell, or any solicitation of any offer to subscribe or to purchase, shares, units or other interests in investments that may be referred to herein and must not be construed as investment or financial product advice. Acadian has not considered any reader's financial situation, objective or needs in providing the relevant information.

The value of investments may fall as well as rise and you may not get back your original investment. Past performance is not necessarily a guide to future performance or returns. Acadian has taken all reasonable care to ensure that the information contained in this material is accurate at the time of its distribution, no representation or warranty, express or implied, is made as to the accuracy, reliability or completeness of such information.

This material contains privileged and confidential information and is intended only for the recipient/s. Any distribution, reproduction or other use of this presentation by recipients is strictly prohibited. If you are not the intended recipient and this presentation has been sent or passed on to you in error, please contact us immediately. Confidentiality and privilege are not lost by this presentation having been sent or passed on to you in error.

Acadian's quantitative investment process is supported by extensive proprietary computer code. Acadian's researchers, software developers, and IT teams follow a structured design, development, testing, change control, and review processes during the development of its systems and the implementation within our investment process. These controls and their effectiveness are subject to regular internal reviews, at least annual independent review by our SOC1 auditor. However, despite these extensive controls it is possible that errors may occur in coding and within the investment process, as is the case with any complex software or data-driven model, and no guarantee or warranty can be provided that any quantitative investment model is completely free of errors. Any such errors could have a negative impact on investment results. We have in place control systems and processes which are intended to identify in a timely manner any such errors which would have a material impact on the investment process.

Acadian Asset Management LLC has wholly owned affiliates located in London, Singapore, Sydney, and Tokyo. Pursuant to the terms of service level agreements with each affiliate, employees of Acadian Asset Management LLC may provide certain services on behalf of each affiliate and employees of each affiliate may provide certain administrative services, including marketing and client service, on behalf of Acadian Asset Management LLC.

Acadian Asset Management LLC is registered as an investment adviser with the U.S. Securities and Exchange Commission. Registration of an investment adviser does not imply any level of skill or training.

Acadian Asset Management (Japan) is a Financial Instrument Operator (Discretionary Investment Management Business). Register Number Director-General Kanto Local Financial Bureau (Kinsho) Number 2814. Member of Japan Investment Advisers Association.

Acadian Asset Management (Singapore) Pte Ltd, (Registration Number: 199902125D) is licensed by the Monetary Authority of Singapore.

Acadian Asset Management (Australia) Limited (ABN 41 114 200 127) is the holder of Australian financial services license number 291872 ("AFSL"). Under the terms of its AFSL, Acadian Asset Management (Australia) Limited is limited to providing the financial services under its license to wholesale clients only. This marketing material is not to be provided to retail clients.

Acadian Asset Management (UK) Limited is authorized and regulated by the Financial Conduct Authority ('the FCA') and is a limited liability company incorporated in England and Wales with company number 05644066. Acadian Asset Management (UK) Limited will only make this material available to Professional Clients and Eligible Counterparties as defined by the FCA under the Markets in Financial Instruments Directive.

ACADIAN

GLOBAL AFFILIATES

Boston London Singapore Sydney Tokyo

ACADIAN-ASSET.COM