Labrador-Level Advanced Analytics: Fraud and Risk Models from Someone Who Never Quite Made It to School

Author: Richard Churchman
Version: 1.0 FINAL
July 2026

I am Richard, the developer of Jube, which is open-source software for real-time Anti Money Laundering and Fraud Detection. “It is just a bunch of if statements” is a statement that comes up with alarming regularity in the discussions of scoring systems and is usually wrong borrowed knowledge from the expert systems days (and even ChatGPT would blush via its biological interface to the outside world). It is a statement I struggle to bite my tongue in the retort, as the inference is that the counterparty to the discussion could not quite be arsed with multiplication of stuff, or summing a collection of things up (at which point “It’s just an [a single] if statement” takes on some truth). The reason I find the statement irksome is that statistical models are the most elegant and consistent tools we have in the bag, while also – especially in the case of Bayesian Networks – have some profound philosophy that anchors my whole consequentialist outlook in life, is the basis for my own human empathy and, moreover, is the secret sauce for escaping the most soul sapping extreme edge case crisis meetings (“don’t know, ask the data…..where is the coffee machine?”).

The whole discipline of advanced analytics is a little bit esoteric, which is a shame, as it has rich academic underpinning, but it is often rather obtuse in its presentation, or in the case of Heuristics and Bias, thunderously dull. Much of the academic text written is perfectly correct, but not always easy to understand. It should not be lost that the genuine thought leaders in the field, specifically Judea Pearl, and a nod to Martin Neil and Bart Baesens, manage however to explain extraordinarily profound and complex topics in a manner that could be understood by a Labrador. To invoke Einstein, "if you can't explain it simply, you don't understand it."

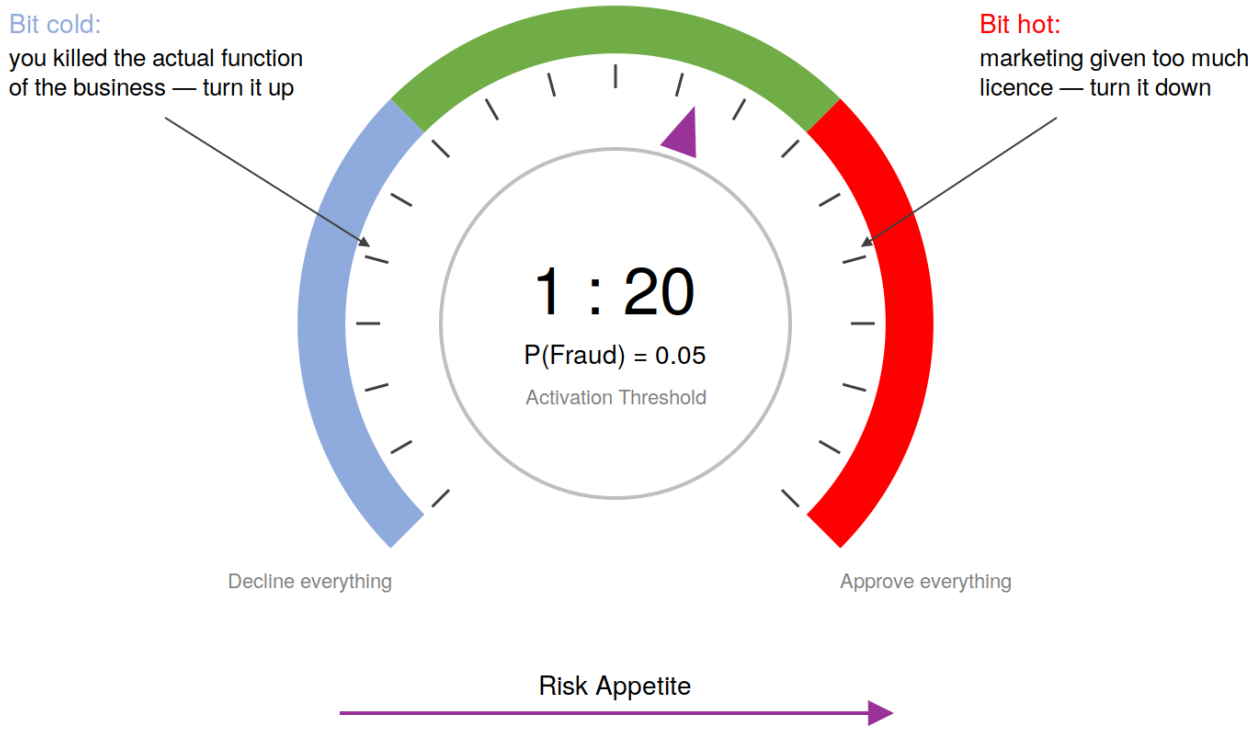
Here as follows is your faithful Labrador taking a swing at explaining it, if only so we never get trapped in a room together with “It is just a bunch of if statements” on the tip of tongue. The whitepaper I present is an abstraction of work I have done to digest this discipline into procedures: https://www.jube.io/AdvancedAnalyticsWithRGuidance.pdf. In time, I will probably spin out dedicated whitepapers exploring each technique in more depth, to bridge between the procedural resources and something passable as content.

My favouring of R requires some defending I know which I can't go much further than it being the bedrock of most of the Credit Risk writing on the topic, and that I quite like it for what I need it for (models are in R, everything else in C#). Much, if not all, if not more, can be achieved in Python, and if that sits better, happy days. Choice of data frame and statistics libraries hardly matter, if the fundamentals are consistent, and the resulting models can be recalled via HTTP plumbing, where both R and Python have first class support in this respect.

In the age of AI, what follows is somewhat of a departure, but in my opinion, the approach remains the only means to hold out defensible analytical work product in regulated environments. I am hitherto just not quite brave enough for AI, and I don't think it is hard to find anecdotal evidence of AI hallucination as to why (which is sort of the whole point, as this stuff is not, by its nature, anecdotal). The most unfashionable assumption of this paper, in the current climate, is that it directs the reader to books - so, to borrow from the opening of The Lego Movie (cue husky voice): get ready to… READ.

Rules are great, but they are usually akin to a battle dressing, stops the bleeding, but ugly and not sustainable. Rules are subject to a few kinks, the first of which is that they tend to be crafted on the basis of a Subject-Matter Expert's (SME's) judgement (or worse still a decades old battery of best practice rules) and they rarely transpose well between clients. More troubling however is the lack of quantitative evaluation leading up to the creation of rules which tends to yield models – and yes, a rule is a model – which are haphazard and not fit to evolve (we term this adaptation). Over time, the approach leads to hundreds of rules, without quantitative rationale, and becomes a maintenance headache firstly, but above all, sends entirely the wrong signal when the regulator comes knocking; Why? Dunno, Bob left four years ago.

Before venturing further, take a moment to dissect a rule. A velocity rule (count of things meeting certain criteria, triggering on a given threshold) has two component parts. First is the actual count, which we term Abstraction (also known as a feature) and the second is the threshold (the point at which that count tips a balance and fires). Abstraction is the secret sauce of machine learning, and while a threshold is unavoidable, what we term Activation, it needs to be on top of a single quantitative score representing rich trade-offs in the Abstractions. The availability of a single quantitative score, especially in the case of probabilistic techniques representing a well calibrated probability (i.e. one in whatever is fraud), facilitates operational control based on risk appetite in the moment. Imagine if you will the overall risk management mission being akin to a thermostat, as follows.



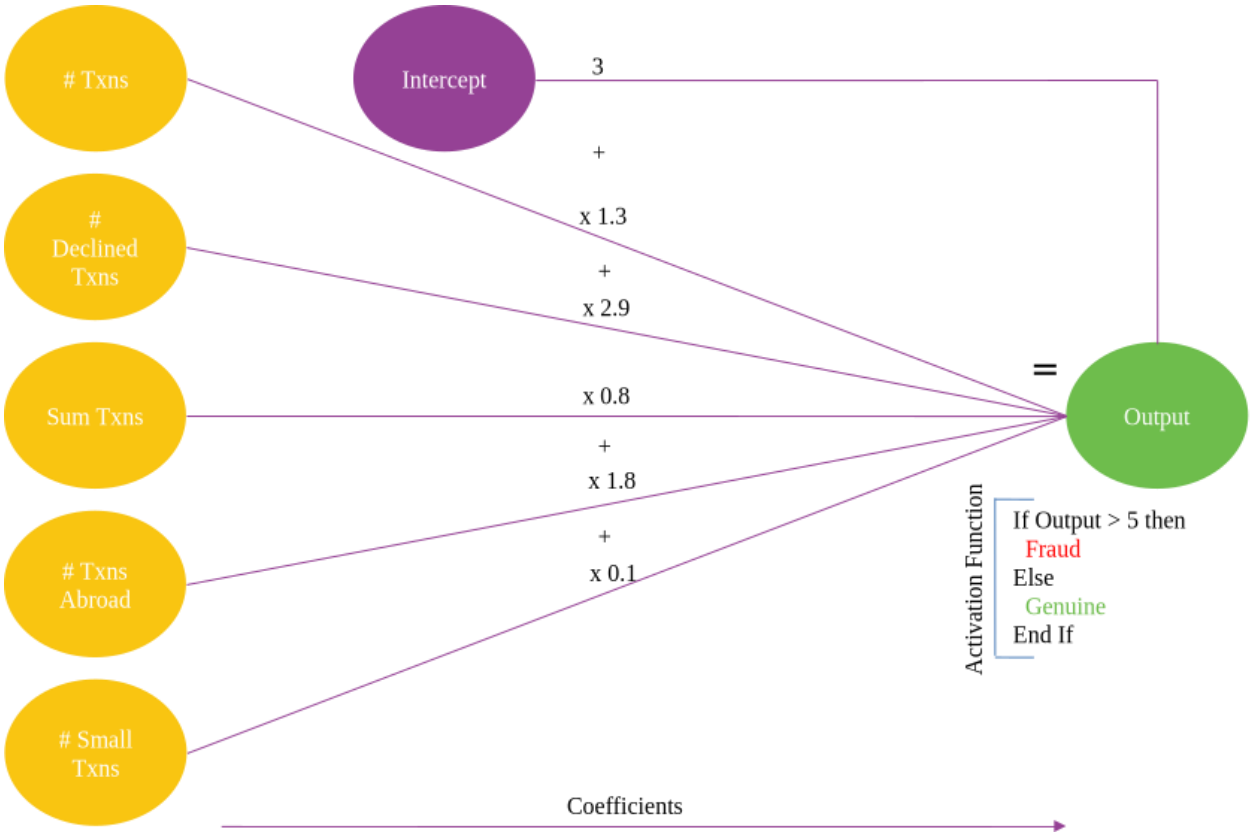
A bit hot, marketing team has been given too much licence, turn down the temperature, a bit cold, you have killed the actual function of the business, turn up the temperature. The whole endeavour should be distilled down into a single quantitative score, with clear, well calibrated, and validated probabilities (does one in five transactions being fraud at a threshold mean that in reality). In reality, models decay at an uncomfortable rate, and in practice the score tends to be blended with expert rules that beat it to the punch: while creating and deploying a new model is a straightforward, near-procedural process, it is nowhere near as quick as an Expert Rule (although, jumping back, separating model topology from weights, and tweaking the weights as new data arrives, is a practical middle ground – and the most reasonable means of holding out a genuinely adaptive system).

Up to this point, expert rules have taken a bit of shade, which is perhaps unfair, if we are to reframe the Expert Rule as being a model, a basic model, one that has been haphazardly derived more often than not, but a model, nonetheless. If we can arrive at clear probability though, then it becomes a model, that can be validated like any other. Here I would introduce the first of three books that I would consider required reading, Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS (ISBN: 978-1119278283), Credit Risk Analytics: The R Companion (ISBN: 1977760864) and Fraud Analytics Using Descriptive, Predictive, and Social Network Techniques: A Guide to Data Science for Fraud Detection (ISBN: 978-1119146834), all written by Bart Baesens of KU Leuven. While hopelessly gauche in the AI age, please heed the clarion call that whatever work product produced must pass regulatory scrutiny, and there is no more tightly regulated sector than BASEL, and even though not absolutely required, try and anchor to the explanatory requirements of BASEL, as it is awkward regulator conversations wholly avoided such is the acceptance of the generally accepted practices in model creation and validation. Bart's work is enormously valuable and has allowed practitioners who don't have a rigorous academic

background, such as me, to participate in what would otherwise be a quite obtuse discipline. Keep in mind that nobody has the money for SAS set off against an open-source landscape, which is why Jube is an R shop (although you could be Python if that is more to taste).

With the validity of Expert Rules at least partially restored, the first tool to reach for in the dusty bag is Logistic Regression, part of the Generalized Linear Model (GLM) family, and frankly, you could probably stop reading after this section, having arrived at a state of more than good enough. The goal of Logistic Regression, as is the case with all models, is to bring the abstractions together in a fashion that allows trade-offs to be made, digesting the decision into a threshold on the single quantitative score.

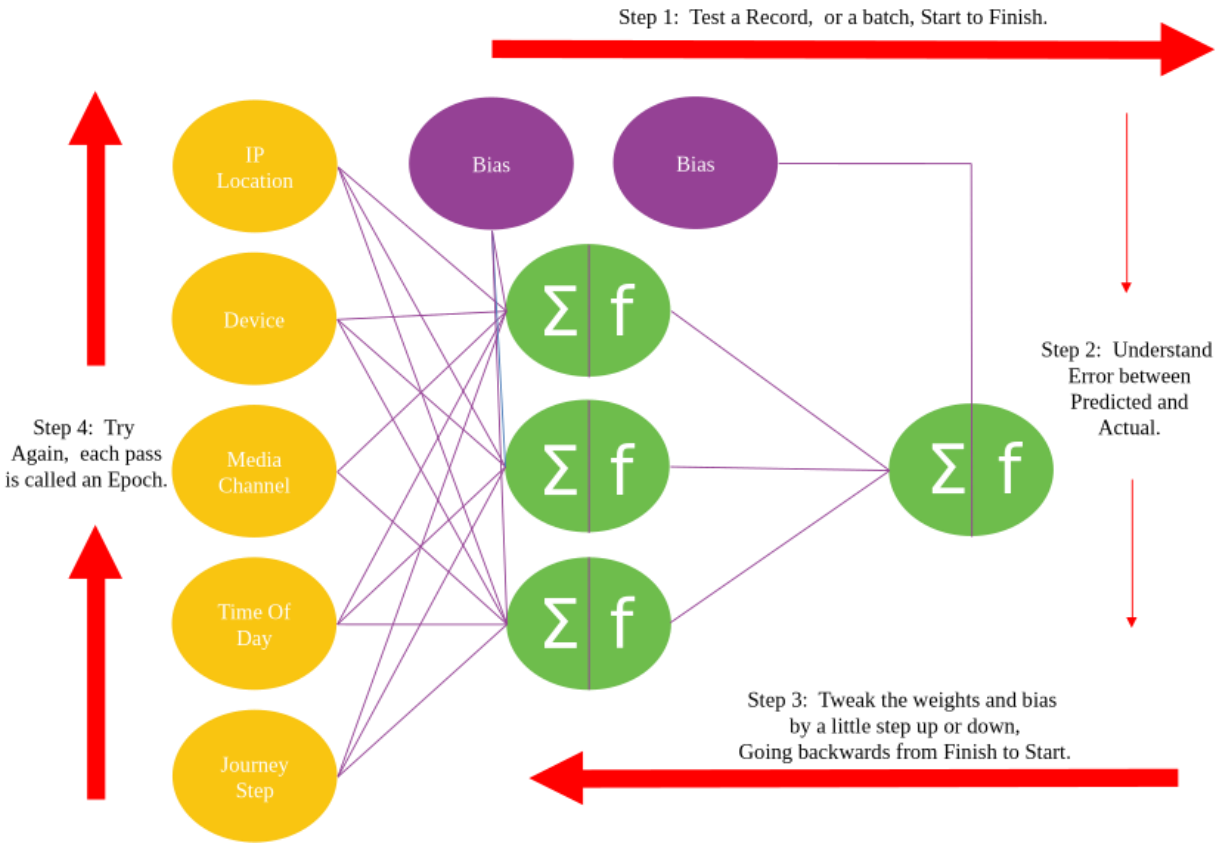
Abstraction deserves, and will in time receive, a whitepaper of its own – it is the secret sauce, after all, and a couple of paragraphs here would be an injustice. For the present purpose, the intuition is this: a raw transaction says almost nothing; an abstraction says what that transaction means in context. The count of transactions in a day, the sum against the customer's own weekly average, the variance that says whether that average can be trusted, the flag that spend has strayed outside the product's typical merchant categories or countries – each is a small act of expert judgement frozen into the data pipeline, computed over windows of a day, a week, a month, a lifetime, against the customer's own history and against their peers. Get these right and even a modest model sings; get them wrong and no amount of topology will save you – which is why the models in this paper are the shorter half of the story. The full catalogue of abstractions, set out prescriptively by risk factor and grouping, lives in the Jube guidance notes: https://jube.io/JubeAMLMonitoringComplianceGuidance.pdf. With the assumption made that our pipelines – probably Jube – have created that rich corpus:



The model, going left to right, takes the inputs, multiplies the value of the inputs by coefficients, adds that value to a starting point, the intercept, and produces a log-odds value, typically landing in the region of ±5 in practice, which is transformed into a probability, which

can be probabilistically interpreted. The residual task is to decide where to set the threshold, or the turning of the thermostat, which is something I refer to as activation in Jube, for reasons that will become clear as we evolve the Logistic Regression topology out into Neural Networks. Never let anyone tell you that Logistic Regression is not a Neural Network, it is just basic, and the way the weights are created differs, but topology wise it conforms in all but a little variation on the language (a coefficient is called a weight, an intercept is called a bias). The whitepaper does not explore the process of creating the coefficients or intercepts, but this is comprehensively documented in Jube's procedures: https://jube.io/AdvancedAnalyticsWithRGuidance.pdf.

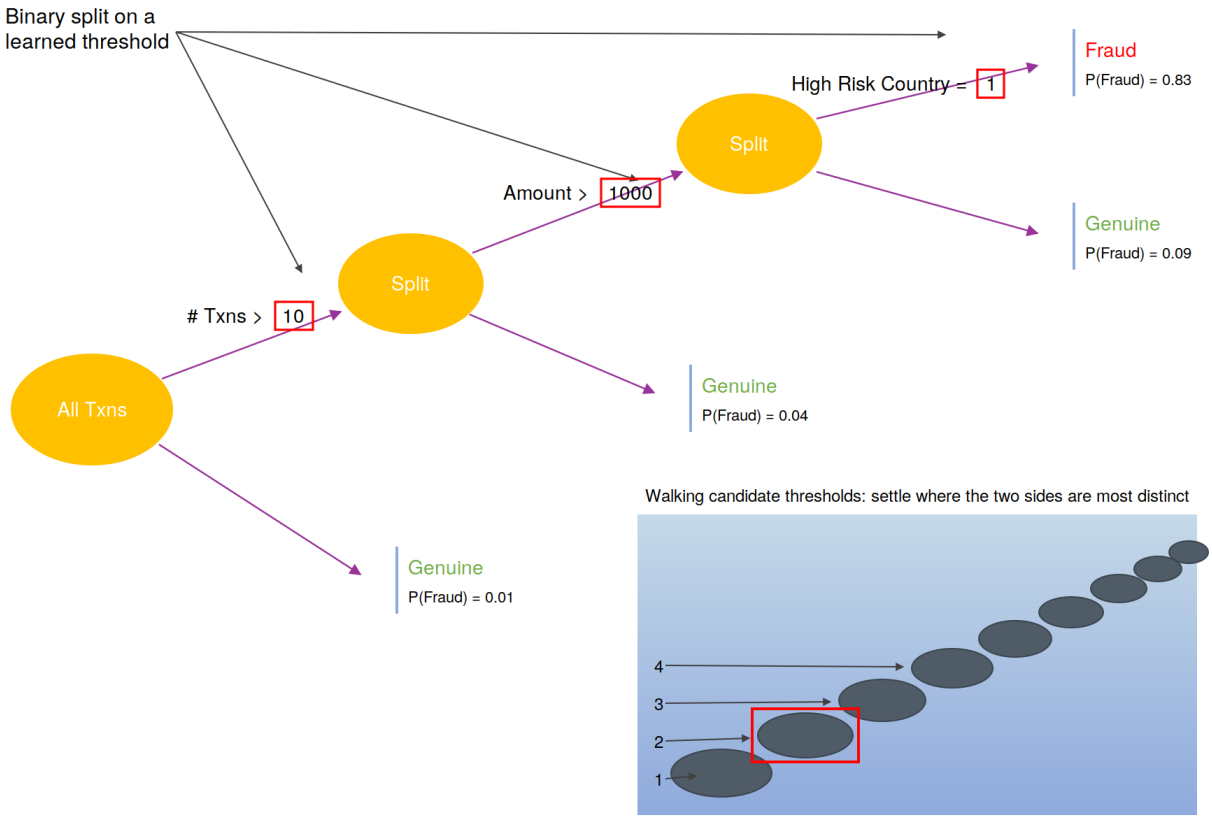
There are some limits to Logistic Regression however, and in my practical experience, any more than 12 variables, you will struggle to get it to converge well (which I take to mean here the P values being anything like defensible). A Neural Network meanwhile earns its keep by shrugging at very large abstractions being passed, albeit at the loss of explainability, which is a departure from the maxim of explainability that is generally accepted by a regulator. A Neural Network looks a lot like a Logistic Regression model, except for including the concept of hidden layers and processing elements, which allow the model to perform its own abstractions which hitherto have relied on expert judgement. Offloading Abstractions sounds tantalising at first pass, until the realisation that it can only abstract on what it has been given, and can't interrogate historical repositories to identify volumes or velocities, and expert judgement in the creation of the core features is largely unavoidable.



The above diagram gives a nod to the radically different approach to arriving at the coefficients, although we now call them weights, and the intercept, which is now called the bias. In the explanation of sum of product and threshold introduced in Logistic Regression, we have swapped in a more succinct representation in the form of ∑ and f, where the ∑ is the summing up of the variables multiplied by the weights, added to the bias, and the f is how it is activated (i.e. exceeds a threshold). The number of hidden layers and processing elements

are described as topology, or part of topology at least. Neural Networks, as Bayesian Networks to come, are especially powerful in that topology is separate from the weights, or probabilities, and can facilitate a truly adaptive system. In practice, these models rarely activate on a threshold, to trigger, or feed forward, their value, and it is almost always a Sigmoid transformation, but conceptually, hardly matters, subject to iterative improvement of the topology, with what is a received wisdom or trial and error process seeking continual improvement. Through experience, there is no alternative to doing the hard yards of exhaustive topology exploration seeking constant improvement, which is how Exhaustive came to be: https://jube-home.github.io/aml-fraud-transaction-monitoring/Configuration/ExhaustiveAdaptation/. All taken together, Neural Networks will likely give you a three to five percent improvement in classification accuracy over Logistic Regression done right.

By way of an honourable mention, though neither my first nor second choice, are decision trees, in the flavour of C5 or Random Forests. Trees depart from everything discussed so far in that there is no sum of products at all; rather the algorithm recursively partitions the data, greedily hunting for the split that best separates the classes, and the result is – delicious irony fully intended – genuinely a bunch of if statements. This makes a single tree supremely explainable, to the point it can be read aloud to a regulator, and better still, deployed directly as a coder rule in Jube, sidestepping HTTP serialisation and transport cost altogether (of which more later). The probabilistic slicing comes from the leaves: each terminal node carries the proportion of training cases in that leaf belonging to the target class (or in a Random Forest, the share of trees voting for it), which serves as a raw probability. The caution is that these raw probabilities are notoriously badly calibrated – single trees are lumpy, forests herd towards the middle – and so a pass through Platt scaling or isotonic regression is needed before the score can honestly claim that one in five at a threshold means one in five in reality, at which point it slots onto the same thermostat as everything else.



The above diagram is simplistic – please don't deploy it – with the stepping stones intended to illustrate how splits are selected: the algorithm walks through candidate thresholds, settling on the split where one side and the other are as distinct as they can be made. Even on a good day this feels inelegant, not least when it comes to wrestling the result back into a single quantitative score. It follows that I tend to reach for decision trees when the work product is destined to be an expert rule deployed to Jube, rather than recalled into a single quantitative score.

Despite being the author of Exhaustive, I revisit my clarion call to Bart Baesens' work; Given the lack of explainability at the point the regulator comes knocking, I would ask why would you do it to yourself? The juice is just not worth the squeeze. Rather like nuclear weapons, once Neural Networks exist, everyone feels obliged to have them, and – in the context of my stressors, which is to say regulators – I rather wish we could disinvent the lot.

Which brings me to my weapon of choice, Bayesian Networks – though not without introducing some more required reading. Judea Pearl, of UCLA, is the godfather of probabilistic causality, and living proof of Einstein's quote on depth of understanding. The foundational work is his Probabilistic Reasoning in Intelligent Systems (ISBN: 978-1558604797), which gave Bayesian Networks to the world, but my favourite – and the one I would press into your hands – is Causality: Models, Reasoning and Inference (ISBN: 978-0521895606), which is as profound a book as I have read, and written in a remarkably accessible manner. The adjacent work – one for the sunbed – is The Book of Why: The New Science of Cause and Effect (ISBN: 978-0465097616). In work I have taken to be an abstraction of Pearl's, would also encourage a peruse of Risk Assessment and Decision Analysis with Bayesian Networks (ISBN: 978-1351978972) by Martin Neil, of Queen Mary University of London and, as we will shortly need to build, Bayesian Networks with Examples

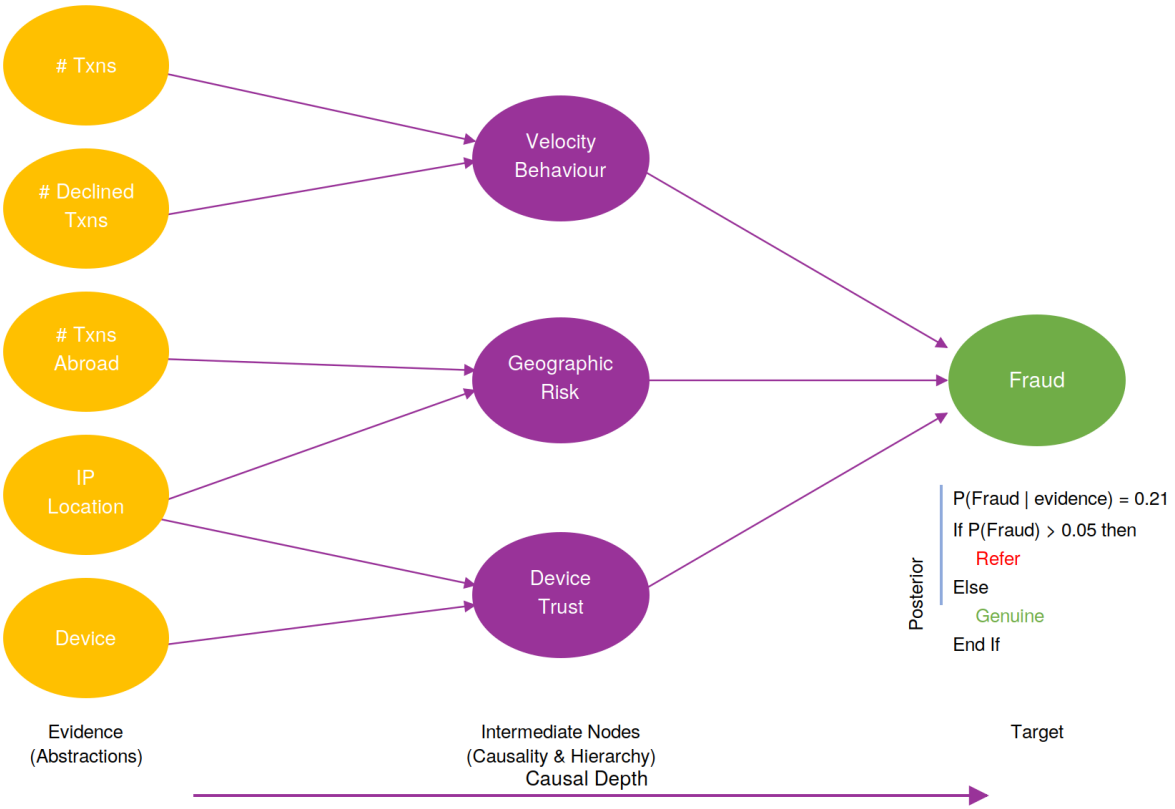
in R by Marco Scutari, of the Dalle Molle Institute for Artificial Intelligence in Switzerland and the author of bnlearn (ISBN: 978-1000410389).

I permit myself one brief departure from the technical, as is a whitepaper author's occasional privilege. The two thinkers who have most shaped my outlook - Pearl, and in another sphere entirely, Gabor Maté - carry their heritage into a moment of profound sorrow, and I will leave the politics exactly there. What I will not leave unsaid is this: Pearl lost his son Daniel to extremist violence in 2002, and answered the worst thing that can happen to a father not with bitterness but by founding, in Daniel's name, an institution devoted to cross-cultural understanding and dialogue. It is one thing to write the mathematics of updating beliefs on evidence; it is quite another to refuse, under the cruellest evidence imaginable, to abandon the belief that people of every creed and culture are fundamentally the same. He wrote the first and lived the second, and that, more than any theorem, is why his name anchors this paper.

Bayesian Networks are, at their core, probabilistic. Suppose you watch the weather channel, and it says that Cyprus in winter carries a 33.3% probability of rain (the Cyprus meteorology department prefers to issue such predictions three or four days after the drainage systems have long given up, mind you). What this means to say is: Wait three days and, on average, one of them will rain. If that prediction holds, it is judged to be a well calibrated probability. This language, in the sphere of fraud analytics and credit risk, is enormously valuable, as it allows for the unambiguous communication of how a model will perform in the wild – one in five transactions above this threshold will be fraud, and we staff the queue accordingly – and, via model validation techniques, unambiguously measures expectation against reality, and the model's drift over time. It is the difference between a score that merely ranks and a score that means something; the thermostat only works if the temperature reading is true.

A brief historical footnote that never fails to please: the formula underpinning all of this came not from a computer scientist but from the Reverend Thomas Bayes, a Presbyterian minister, whose theorem was published posthumously in 1763. It is a pleasing thought that the mathematics keeping the regulator at bay was conceived by a man of the cloth, some two centuries before anyone had a transaction to monitor.

A Bayesian Network is, structurally, a directed acyclic graph: nodes representing our abstractions, arcs representing the dependencies between them, with probability flowing through the structure. My own heritage is Netica, where networks are lovingly handcrafted from expert judgement, and there is nothing wrong with that – expert knowledge is encoded exactly where it belongs. The step change in bnlearn is structure learning: the data proposes the graph. Point a Hill-Climbing or MMHC algorithm at the transaction corpus and it will surface dependencies – device fingerprint to geolocation to fraud – that the expert in the room never articulated, and probably never saw. Expert judgement is not discarded but demoted to constraint, in the form of arc whitelists and blacklists that enforce what is known while the data fills in the rest. It is the sanest reconciliation of expert rules and machine learning that I have encountered.



The explanatory value is where Bayesian Networks earn their keep, and where the earlier grumbles about Neural Networks are answered. Every node in the network has a story: evidence enters, probability propagates through the intermediate nodes – causality and hierarchy made visible – and the posterior at the target node is the single quantitative score, with its journey through the graph laid out for inspection. Better still, inference runs in both directions; the same network that predicts fraud given the evidence will, on request, explain which evidence most moved the needle. When the regulator comes knocking, the answer is not a shrug at a gazillion weights, it is a walk through the graph.

Validation, too, arrives with unusual grace. Bootstrap resampling of the structure yields arc strength – the proportion of resamples in which a given dependency appears – and "this arc is present in 94% of bootstrap samples" is precisely the sentence model governance wants to hear, being a statement about the stability of the model rather than an appeal to its author's judgement. Set against the generally accepted practices of BASEL model validation, the whole apparatus holds together as defensible analytical work product, which, as should be apparent by now, is the entire point.

Whatever model we end up with – in my orbit, almost always a Bayesian Network via bnlearn in R – the practical question of deployment follows. I am not the world's biggest fan of microservice architecture; software architects greatly underestimate the overhead of serialisation and transport, and micro-services have a well-observed tendency to sprawl. Here, however, the separation of concerns is genuine – analytics on one side, the pipelines responsible for creating the abstractions on the other – and the pattern earns its place. Once a model is created in R, Plumber wraps it for consumption as an HTTP request, hosted wherever you like, though often it becomes its own Docker image running in wider orchestration. A pleasing side effect of a well-specified HTTP protocol is that the language behind it ceases to matter: honour the protocol, and whether the model was built in R,

Python, or crayon is nobody's business but the author's – which retires the "why not Python" conversation before it starts. The material benefit follows: you gain access to the brightest minds in the discipline, producing quantitatively verifiable output, in the tools they are comfortable with – and if you are bringing in PhDs to help, that is almost always R.

Summing up: the work I enjoy the most – Exhaustive Neural Network Exploration – is the work I am not quite brave enough to deploy. I anchor instead in the most I can get away with under BASEL regulations, applied to fraud problems – which is to say Logistic Regression when twelve variables will do, Bayesian Networks when I want the causality and the explanatory walk through the graph, trees when the work product is destined to be an expert rule, and Neural Networks admired politely from a distance. Not sexy, but I go to great lengths to keep my life stress-free, and the entire promise of this discipline – "don't know, ask the data… where is the coffee machine?" – is a promise entirely broken in the context of models with million-token context windows, 32 layers, and a gazillion weights. When the regulator comes knocking, "the arc is present in 94% of bootstrap samples" is a sentence; a gazillion weights is a shrug. AI, in this regard, is simply a load of stress I don't need – after all, I already have "it's just a bunch of if statements" to deal with.

And on that charge, a closing concession: the accusation is, in the end, half true. The whole apparatus – the abstractions, the coefficients, the posteriors, the calibration – does finally collapse into a single if statement on a single quantitative score. The difference, and it is the entire difference, is that everything above that if statement was measured, validated, and can be defended in a room. It is fitting that this is written in the month a World Cup crowned the pre-tournament favourite while serving up Germany losing to Paraguay on penalties along the way – a well calibrated probability does not promise the absence of upsets, merely that they arrive at the advertised rate.

Perhaps not a faithful Labrador’s rendition on reflection – more a snippy dachshund – but the contents of this whitepaper, in practice, serve to keep complex and highly material risk problems framed inside a common methodology and lexicon, and the thermostat within comfortable reach.