JAYWING

Applying deep learning to credit scoring

Our findings so far

risk.jaywing.com


HELLO.
WE ARE JAYWING.
Martin Smith, Head of Product Development

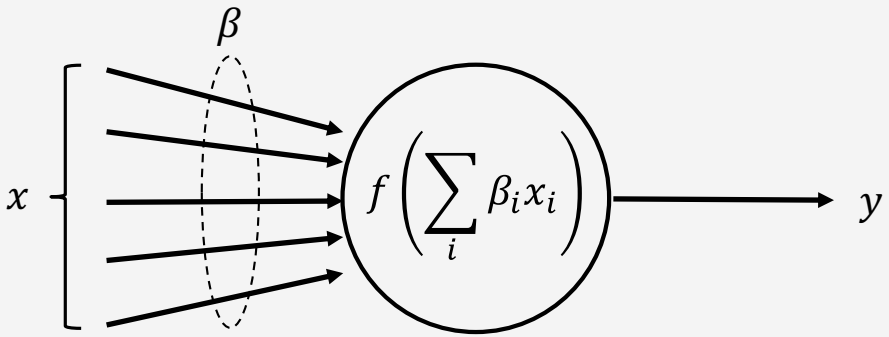
WHY USE DEEP LEARNING?

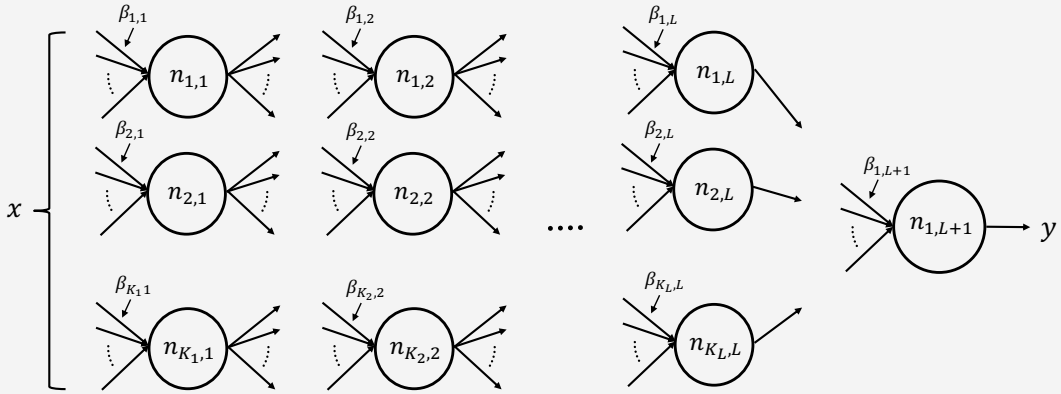
Neural networks are a natural evolution for credit scoring

The familiar linear regression equation takes a variety of inputs, applies a series of weightings, applies a function to give the modelled output.

The neural network is a multi-layered version of this in which the outputs from each neuron form the inputs of a host of other calculations.

Linear models can be considered to be a neural network with just one neuron.





Applying deep learning to credit scoring

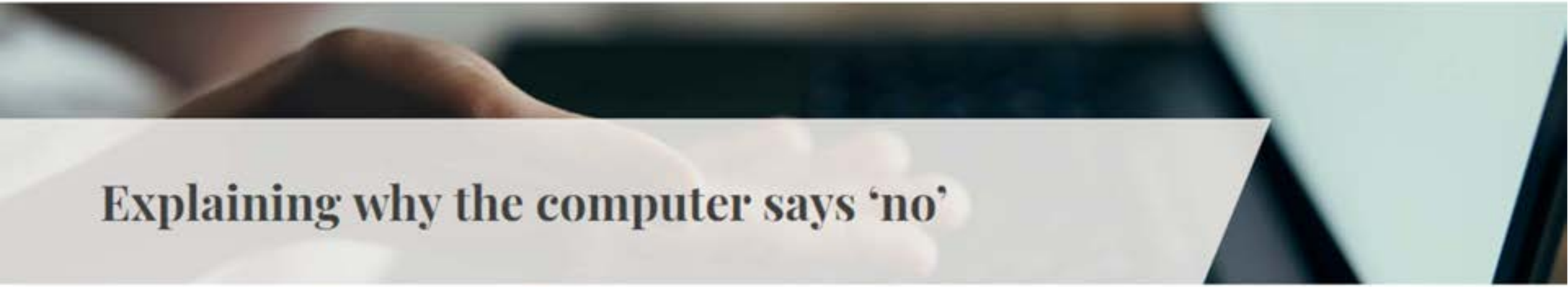
WHAT OUR FINDINGS ARE BASED ON

20+ proof of concept projects, pitching linear regression models against explainable neural networks, using our award-winning Archetype product.

- A range of lender types
- A range of model types – including application risk, fraud risk, behavioural, marketing propensity
- Using data from all 3 of the bureaux
- A full range of product types, from personal loans and credit cards through to buy to let mortgages

Like for like comparisons – same data inputs, same business / governance rules

WHY NOT USE DEEP LEARNING?



Explaining why the computer says 'no'

 Karen Croxson  Philippe Bracke  Carsten Jung

Article 8 mins 4 9

Share

31 May 2019

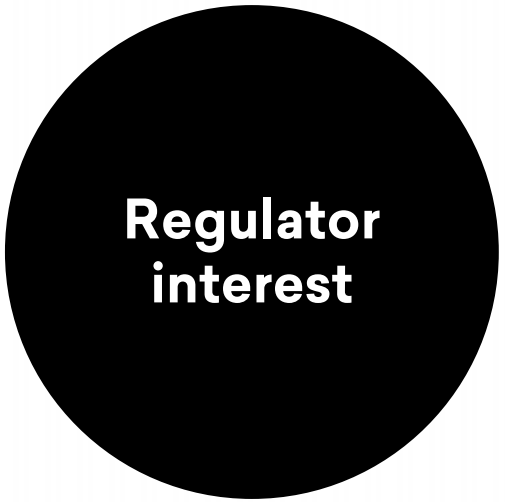
The financial services industry is on the brink of revolution in artificial intelligence. But can the rise of AI decision-making be compatible with the need to explain decisions to consumers?

Opening the black box of machine learning

The financial services industry is facing increasing pressure to explain its decisions to consumers. When faced with possibly life-changing outcomes the customer quite reasonably may have questions: 'Why have I been

Ensuring intuitive responses to input data is critical in credit scoring

Why have I been rejected?



Are customers being treated fairly?
How robust are your decisions?

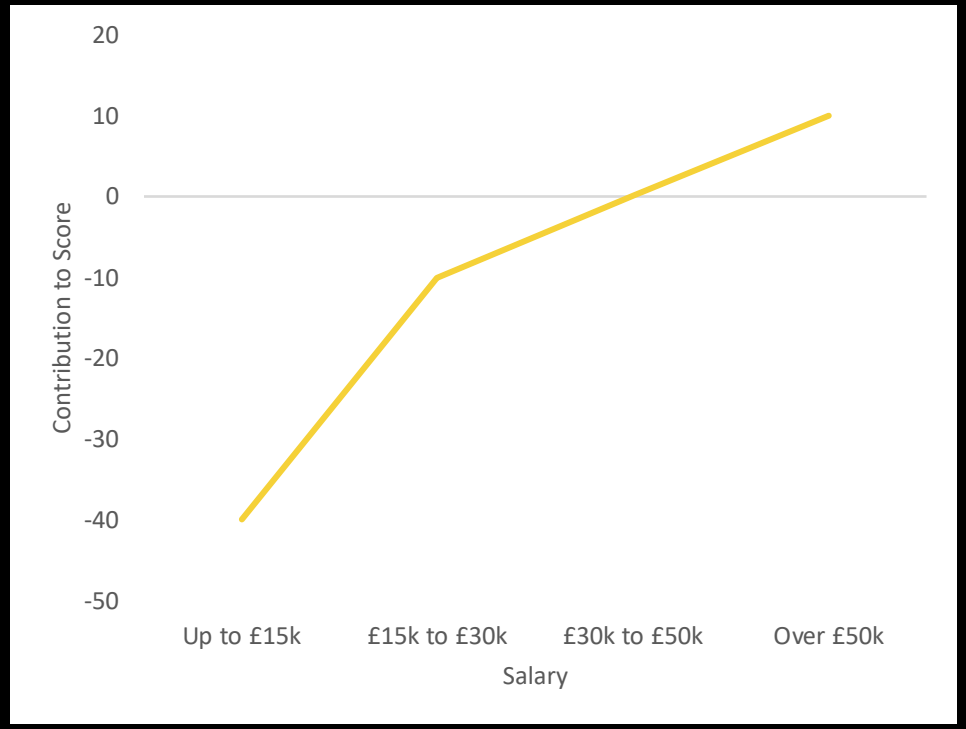
The Senior Managers' Regime:
What is not working and how do I fix it?

The recipe for building a credit scorecard

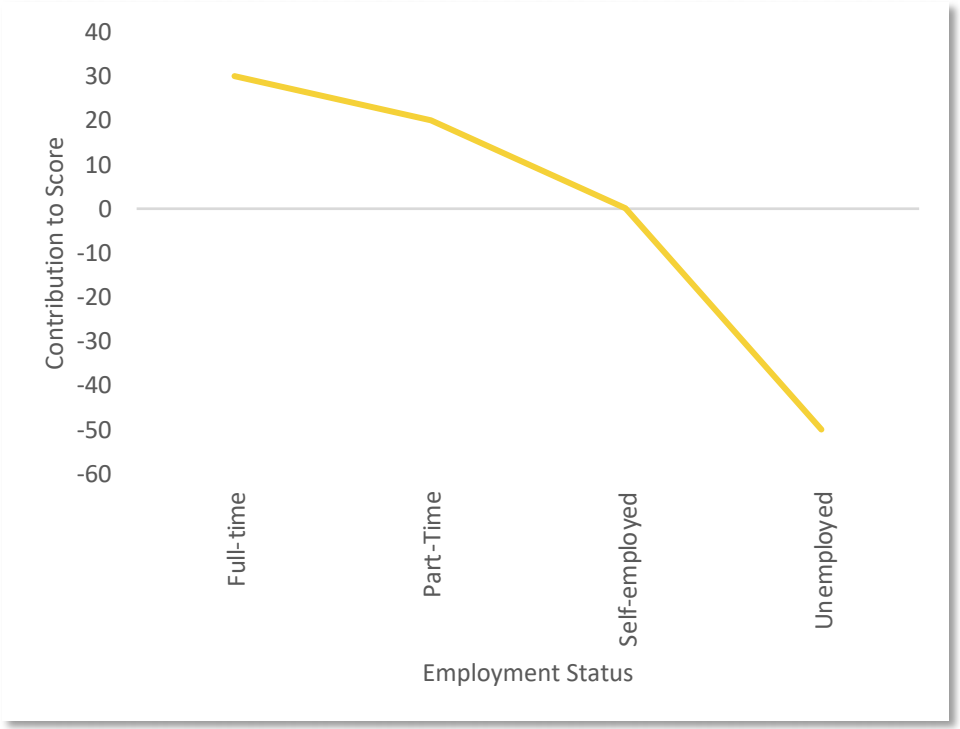
p-Values
Characteristic Reports
Coarse Classing
Scrutinise β's

Intuitive behaviour examples

Increasing salary should always mean increasing score.



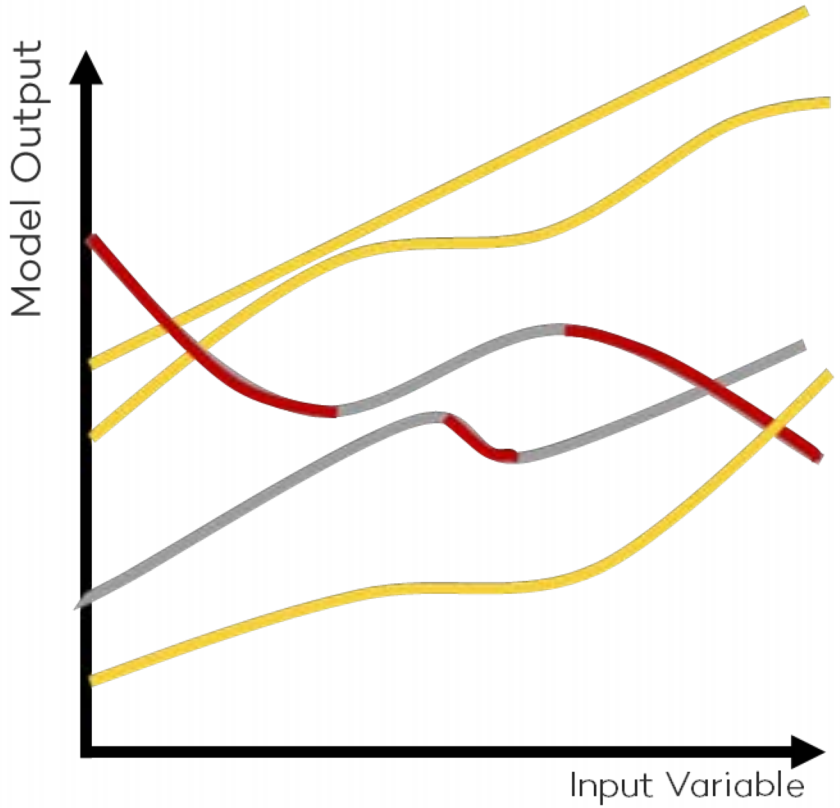
Being in full-time employment should always produce a higher score than being unemployed.



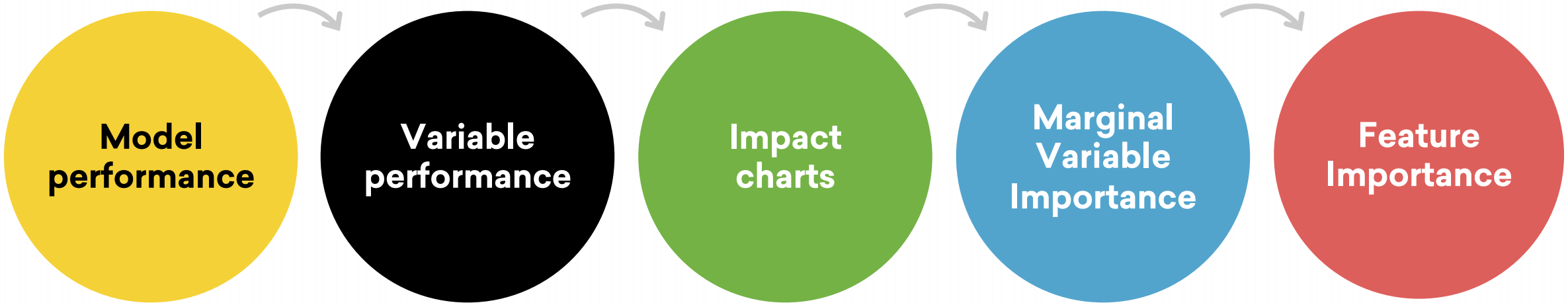
Non-linear models are more complex

The response to a given variable can vary from case to case

- The response can increase for one case and decrease for another.
- It can even vary within a case: increasing for some of its range, but decreasing in other sections.
- This behaviour is almost certain to happen in a non-linear model (albeit infrequently), and preventing it is difficult in general.
- But the model can still be interrogated to understand how it responds to changes in inputs – even though those relationships may be more complicated.



Five essential charts to explain a neural network model



Gini / R-squared curve or scatter plot

How does each variable perform on average, in relation to the outcome?

How does each variable contribute to the model, assuming all else is equal?

Which variables could add more to the model?

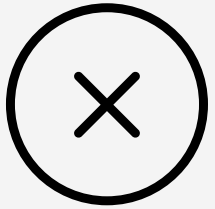
A rank ordered chart showing the relative importance of each variable

‘explainable AI’ is insufficient for credit scoring:
you need control

Just being explainable is insufficient

You need **control**

Being able to explain a model's behaviour doesn't stop it doing the wrong thing in the first place.



But you also need a means of preventing unwanted behaviour.

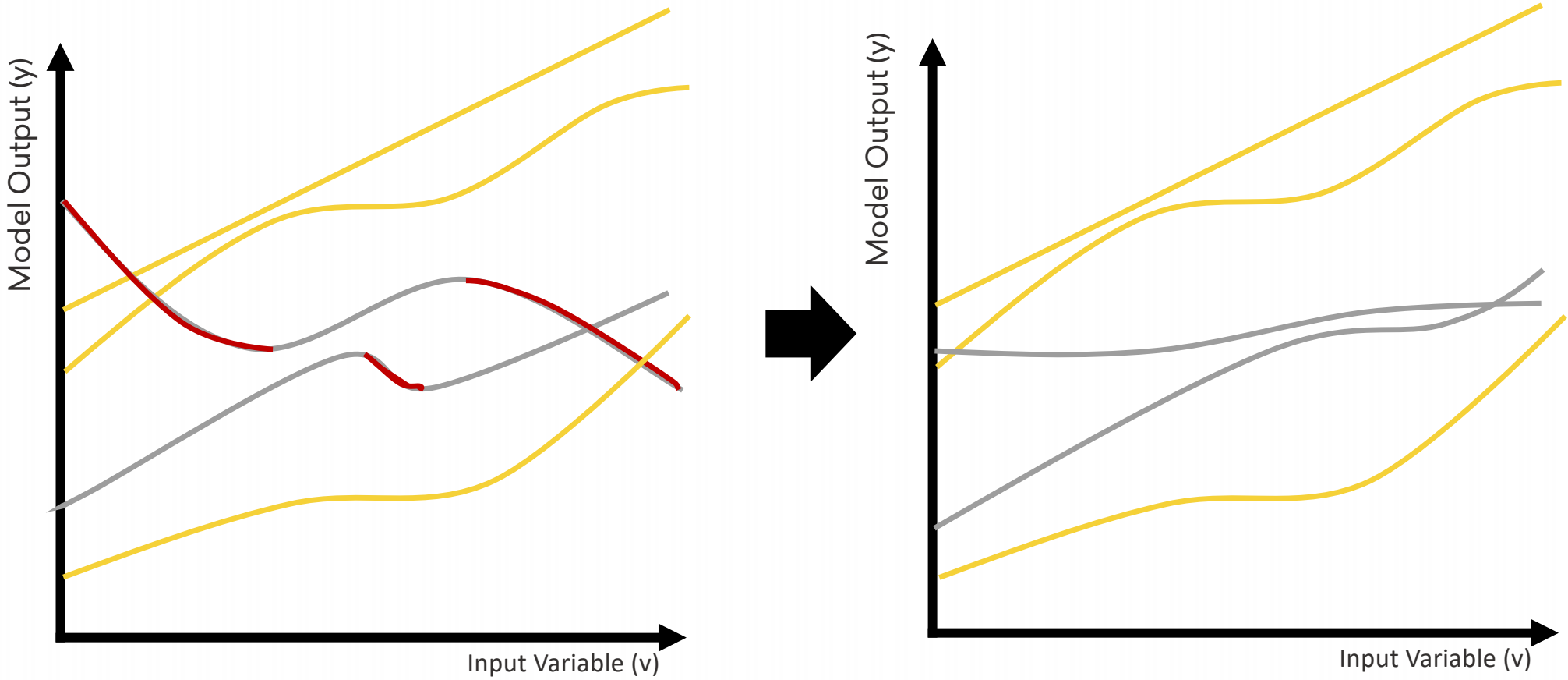
You need human involvement in the modelling process to prevent inadvertent inclusion of bias.

You want to be able to 'design out' the unwanted outcomes so that your model always behaves in a way which is acceptable, not just understandable.

We have developed patent-pending new mathematics that provides certainty regarding how neural networks behave, solving the black box issue.

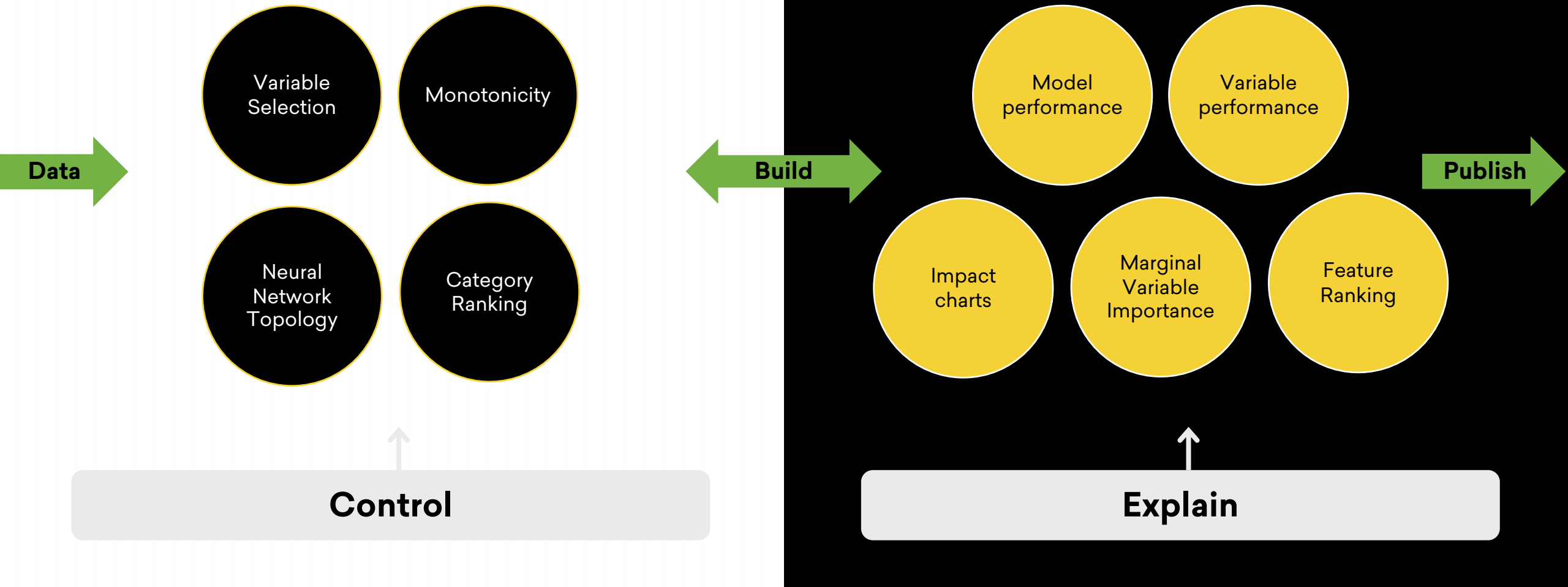
This is a critical enabler for use of AI in regulated industries such as Credit Scoring.

The black box problem - monotonicity



Our solution ensures that the model always behaves as expected

Our approach within Archetype: **control <u>then</u> explain**



Which means...

Governance is front and centre

- Agreeing how the model should function is a design step, not an approval process.
- Re-configured models are guaranteed to adhere to the original model rules and will use the same data.
- Models can be re-developed or recalibrated as often as you wish with limited resource requirement, reducing project costs and keeping the model performance optimal as your population changes.
- New data sources can be introduced with confidence, as you know you can define how they will behave within the model.

OUR FINDINGS AND OBSERVATIONS

Fact
01

Like-for-like Gini uplifts of circa
10% are usually achievable

Deep Learning can improve on optimised linear models

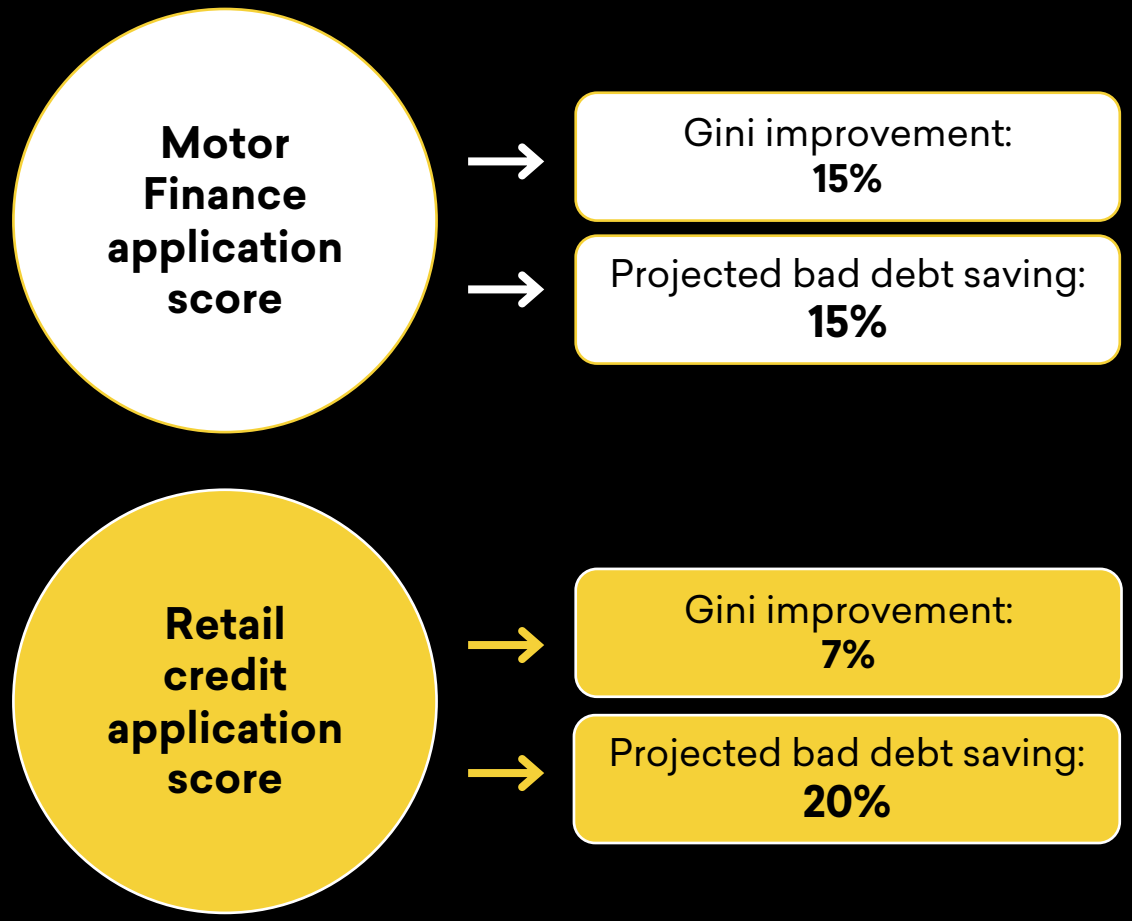
10% uplift is pretty standard.

Even where an organisation has well-performing models in place, Deep Learning is able to squeeze additional insight from the underlying data.

Our like for like tests show that Archetype can generate additional uplifts of circa 10% given the **same data set.** All of the uplifts quoted compare an optimised linear model to a DL model built using the same data. All of the DL models have been fully constrained such that they're fit for use as a credit score.

These uplifts in predictive power translate to highly compelling benefits cases.

Lender type	Model type	Performance improvement
Subprime personal loan	Application score	**11%**
Current account	Application fraud	**10%**
Revolving credit	Behavioural score	**8%**
Credit Card	Application score	**12%**
Subprime retail	Application score	**4.5%**
Residential mortgage	Application score	**5%**
DCA	Payment Probability	**19%**
Peer to Peer	Application score	**2.5%**
Buy To Let	Application score	**19%**
Motor Finance	Application score	**15%**
Retail Credit	Application score	**7%**
Credit Card	Marketing churn	**16%**
Residential mortgage	Behavioural score	**11%**
Prime personal loan	Application score	**8%**



Relative gini uplifts can translate to huge savings

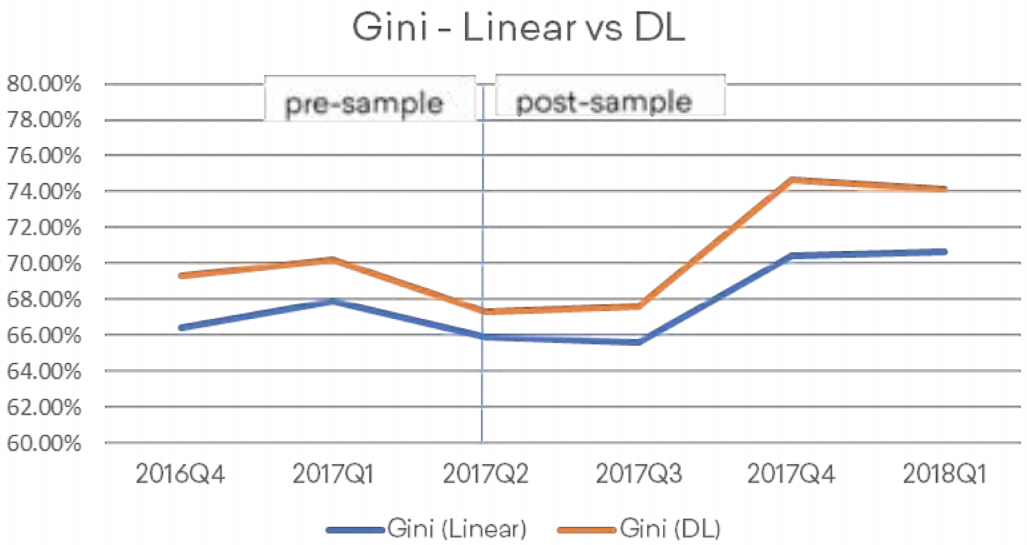
Neural Networks consistently outperform linear models.

Neural Networks consistently outperform linear models. At their very simplest implementation they would collapse to a linear model, and so in general they *can't* underperform based on the same data.

The benefits seen from using DL-based modelling exceed the improvements seen 10-15 years ago by introducing multi-bureau data: it's a further step change in predictive power, and can represent multi-million pound savings with no additional outlay in data costs.

Fact
02

Deep Learning models are **more** stable



Deep Learning models are more stable

But the approach supports continuous review

Somewhat counterintuitively, DL models degrade more slowly than linear models, despite significant extra complexity.

Because of the way the models are built, they don't depend as heavily on a small subset of inputs, making them more stable over time.

The regularisation approach avoids small numbers of characteristics claiming most of the benefit.

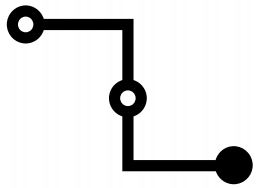
The consequence is that the investment in a new model build lasts longer but – paradoxically – via our approach it is also easier and faster to rebuild more frequently.

Fact
03

Constraints don’t adversely affect performance

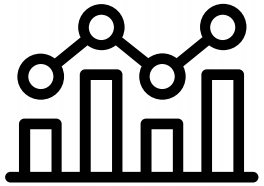
And sometimes they improve it

Constraints don’t adversely affect performance



Neural networks have a tendency to find an alternative route to the right answer.

Fully constraining all of the fields within a typical model development can have a cost as little as **0.2%** on the gini uplift you would otherwise get.



In some cases, such as where the development data is very noisy, constraining the behavior of fields can improve the outcome.

A recent example

Number of Variables in model	Constraints?	Model validation gini
96	Yes	**87.2%**
96	No	**86.9%**
20	Yes	**85.2%**
20	No	**85.4%**

A recent application score comparison

This example is based on:

- One data set offering 500+ variables for modelling, including reject inference
- Different model iterations undertaken with and without data constraints on every field
- Both examples show a minor difference of around 0.2% depending on whether constraints are used or not
- With the larger model, adding data constraints actually improves performance slightly.

Fact
04

50 - 100
variables is the sweet spot

50-100 variables is the sweet spot

Correlation is less of an issue within NNs because of the use of drop-out – so you can safely include more fields.

For bureau-based models, around 50 variables gives the optimum result

Fewer variables can give a similar result, but the marginal gains are worth having.

At 100+ variables you generally hit diminishing returns.

Monitoring is exactly the same as for traditional models, prioritised based on the biggest contributing variables.

As before, control and explainability are essential.

20-30	50-100	100+
Good performance	Near optimal performance	Diminishing returns
	Low risk of over-fitting	Risk of over -fitting
	Rapid model execution	More extensive monitoring
	Manageable monitoring	
	Limited marginal gain from additional fields	

How to get from **3000+ to 50+**

Quality / policy	Remove fields that will never make it through Governance.
Build example model	Build an example model - all remaining columns, shallow sample of data, no constraints.
Univariate exclusion	Exclude based on correlation / similar measures; Drop suspiciously-predictive characteristics.
Build target model	300+ remaining contenders, using all development data, fully constrained where needed.
Non-linear exclusion	Remove fields based on marginal impact and non-linear measures: keep the important contributors; remove those with low marginal value.
Tweak and refine	Optimise the choice of variables as well as the neural network topology. Calculate the marginal information value between actual and expected to find fields that will add more value.

Fact
05

Optimal topology is trial and error

Optimal topology

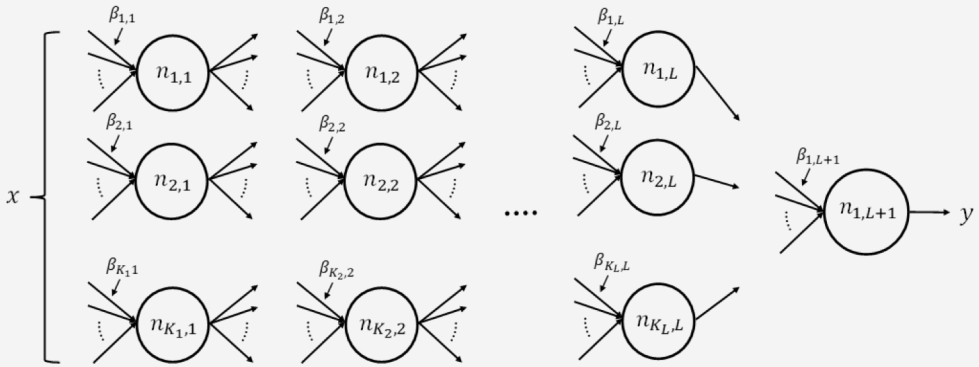
2-3 layers; **200 - 400** neurons works well for application models

We've found, through experimentation, that the above topology works well for binary models.

The more feature engineering you do, the fewer layers you need, e.g.

- By splitting out default values and capturing interactions there's less benefit in going deep.
- We got the same performance from 2 layers and fewer neurons with default values than a non-engineered data set using 4 layers.

For a model predicting repayment amount rather than risk, 5 layers were optimal.



Fact
06

Reject Inference gets even trickier

Reject inference gets even trickier

Using Reject Inference based on a linear model (or an incorrect non-linear model) can generate some unwanted results.

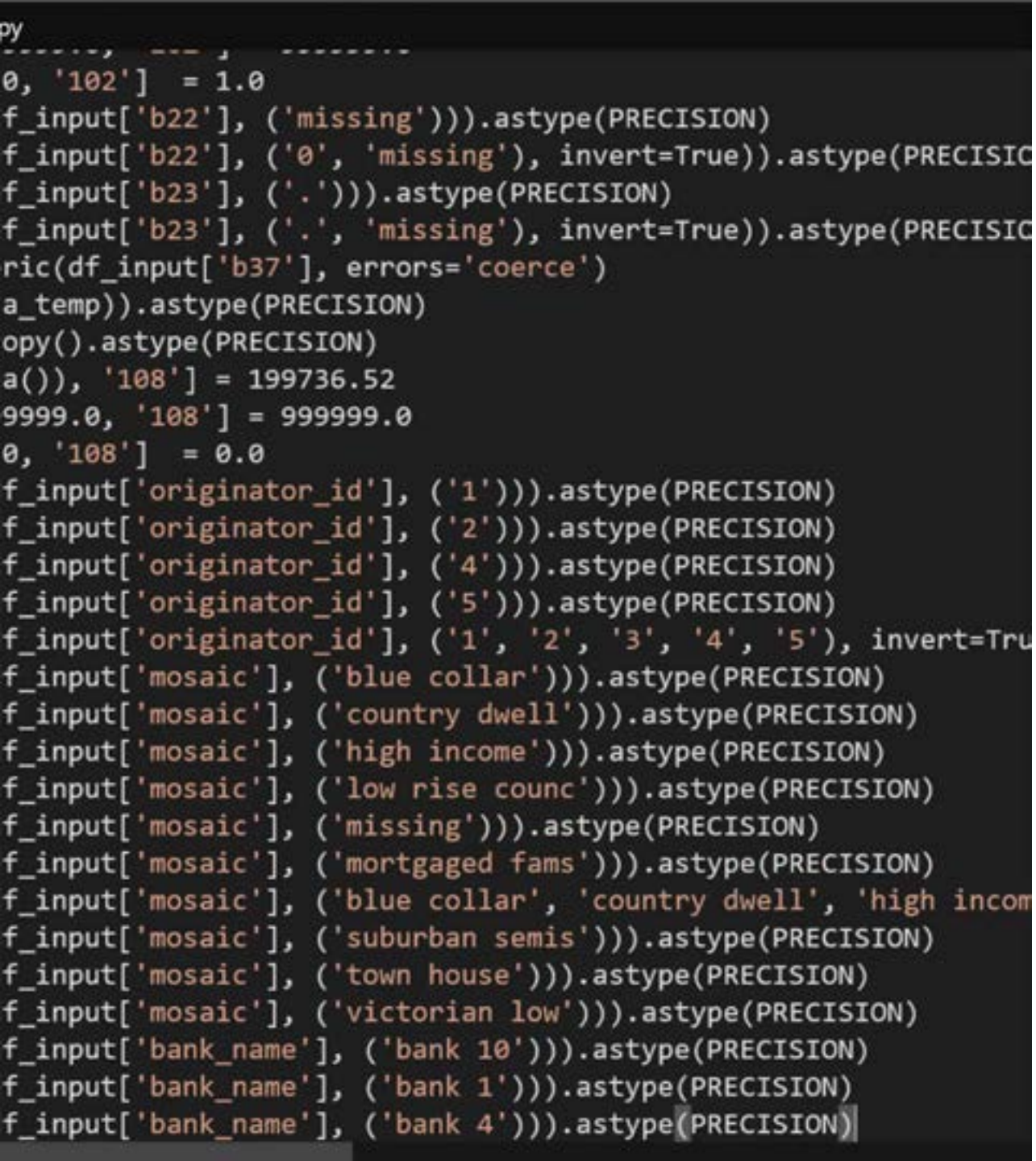
Reject records need careful weighting to avoid dominating the model, and the approach can be tested by checking for correlation between the predicted outcome and the weighting of the rejected records.

In extreme cases you end up predicting the original scorecard outcome and end up with very limited uplift.

Reject Inference needs a different approach – which was the subject of Nick Priestley's talk yesterday

Fact
07

Modelling is complex; deployment is straightforward



Modelling is complex

Deployment is straightforward

- The complexity of neural networks requires significant expertise and heavyweight processing capability during the creation step
- This is required to exploit millions of interactions across thousands of data points
- However the resulting model code, whilst lengthy, is not mathematically complex.
- Neural network models can be deployed in any modern decision system with a scripting capability, or in analytical tools such as SAS
- Our own tool generates this code automatically – in SAS, SQL, Lua, Powercurve or a range of other scripts, which can run efficiently wherever required without any upgrade requirements

Fact
08

AI techniques are comparable.
But...

AI techniques are comparable

Our tests suggest that most AI-based techniques, such as random forests and Gradient-boosted models, generated a similar level of uplift over linear regression models that neural networks do.

Any of these approaches could be used to generate predictions.

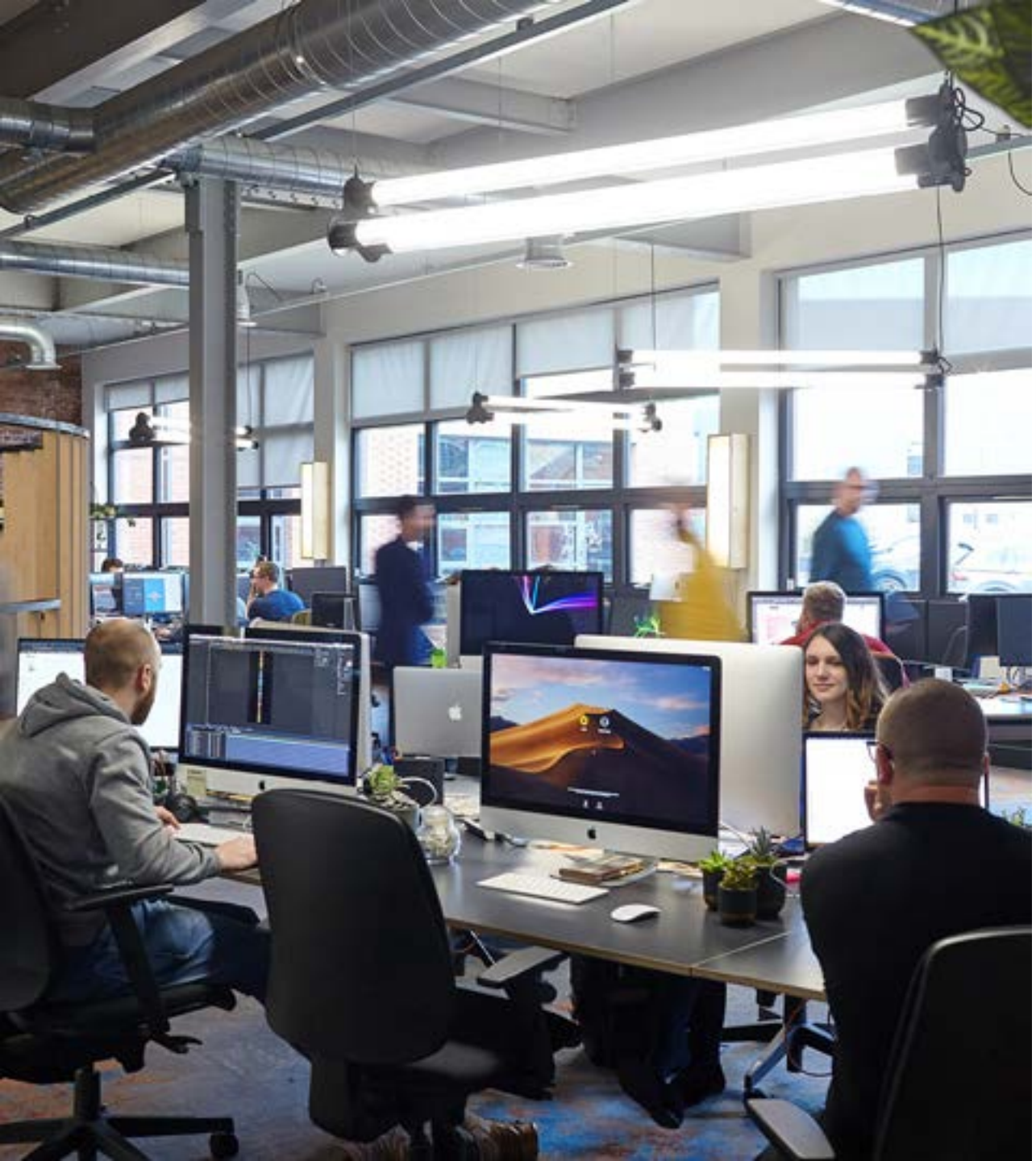
But...

It's only within neural networks that the ability to explain and control behaviour has been solved.

We can demonstrate that this can be done without any significant loss of predictive power (and sometimes delivering an uplift).

Fact
09

Getting the best models still needs good analysts



The best models still involve analysts

You can get good models by throwing data into an AI process.

You get a better model by giving oversight of the process to a skilled analyst.

Analysts bring domain expertise, and data knowledge.

AI doesn't replace the analyst role, it assists and improves it – although you may ultimately need fewer analysts.

AI tools improve the outcome, speed up the process, process more data and undertake more complex modelling than can otherwise be achieved.

Fact
10

Lenders are already benefitting

Revolutionising Secure Trust Bank's credit scoring models using our market-leading AI approach

Secure Trust Bank

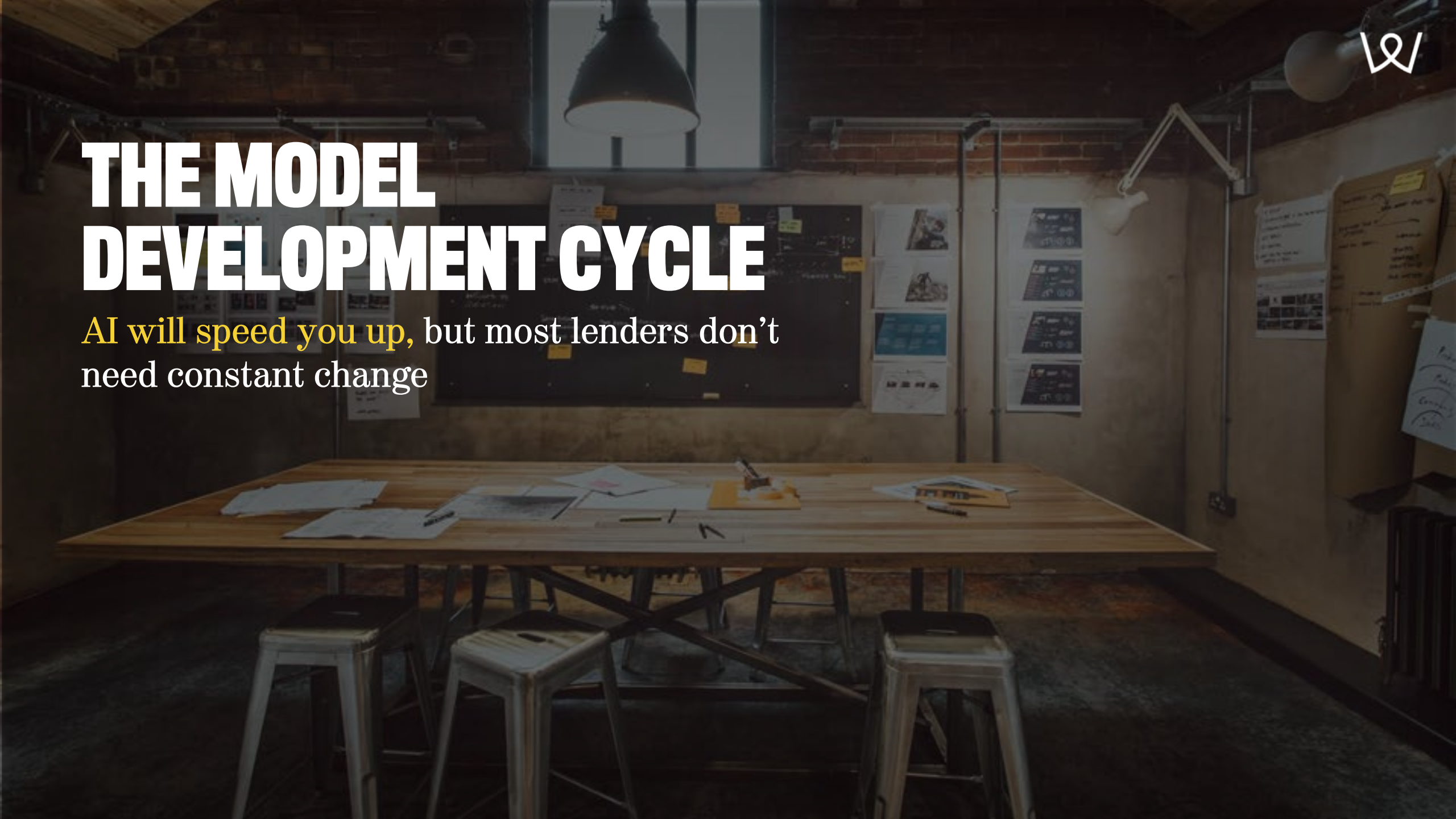
FURTHER THOUGHTS


AVOID BIAS


USE ENOUGH
DATA

SUFFICIENT INTERPRETABILITY

Lower hurdles in areas like fraud or marketing propensity



THE MODEL DEVELOPMENT CYCLE

AI will speed you up, but most lenders don't need constant change

SUMMARY

SUMMARY

Deep learning models almost always achieve uplift over a linear model.

Sufficient interpretability: control then explain.

Use 50 – 100 variables for a credit risk model.

Constrain your variables to avoid unwanted behaviour.

Deployment is relatively straightforward.

Deep learning models are more stable.

The benefits of DL are now available, and lenders are deploying it.


QUESTIONS?

Thank you

Come see us for a demo of Archetype on our stand

Martin Smith
Head of Product Development – martin.smith@jaywing.com

JAYWING