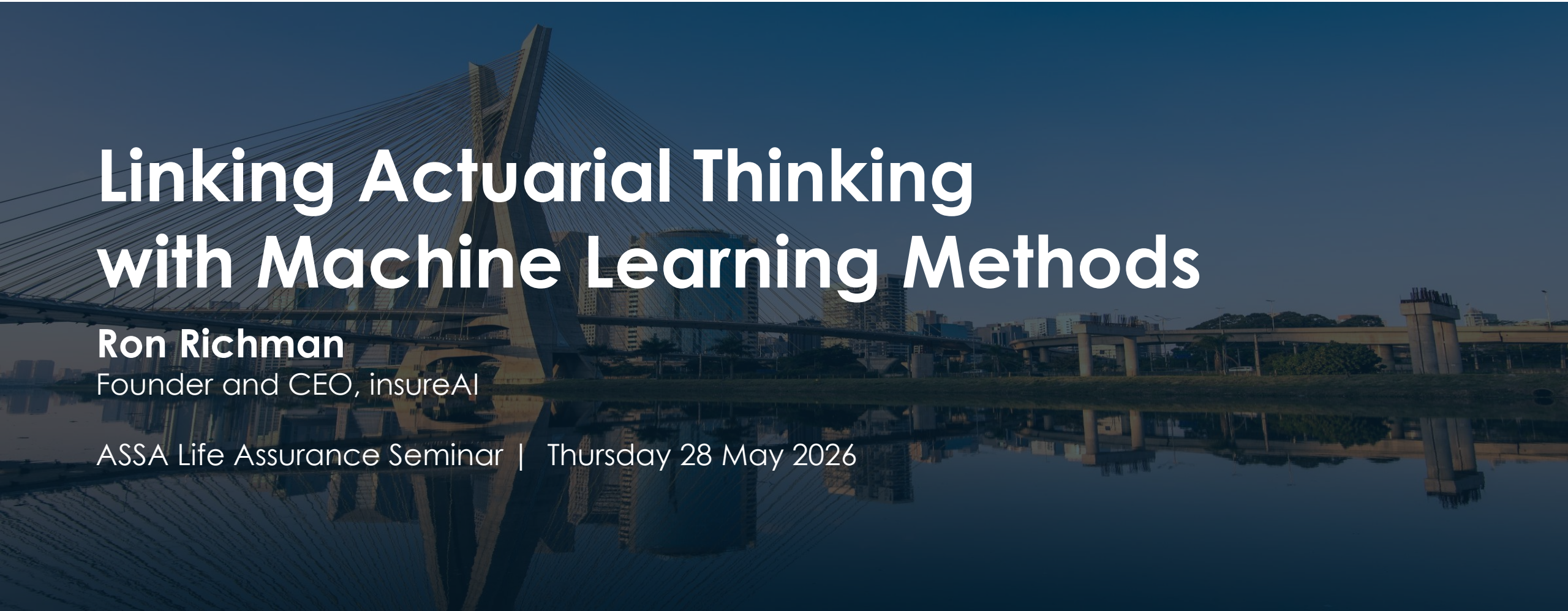
ACTUARIAL SOCIETY OF SOUTH AFRICA
Advancing Actuarial Excellence



Linking Actuarial Thinking with Machine Learning Methods

Ron Richman
Founder and CEO, insureAI

ASSA Life Assurance Seminar | Thursday 28 May 2026

What is this talk about?

- **Building bridges between actuarial science and machine learning**

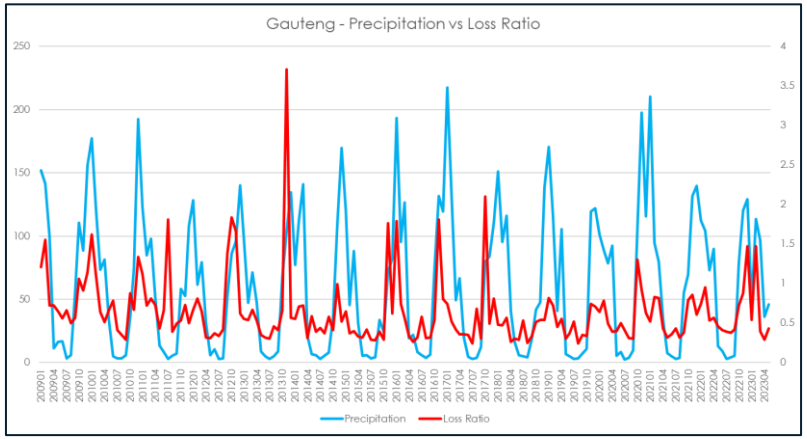
- Bridge 1: both are empirical disciplines

- Bridge 2: scaling the actuarial approach might lead to artificial intelligence

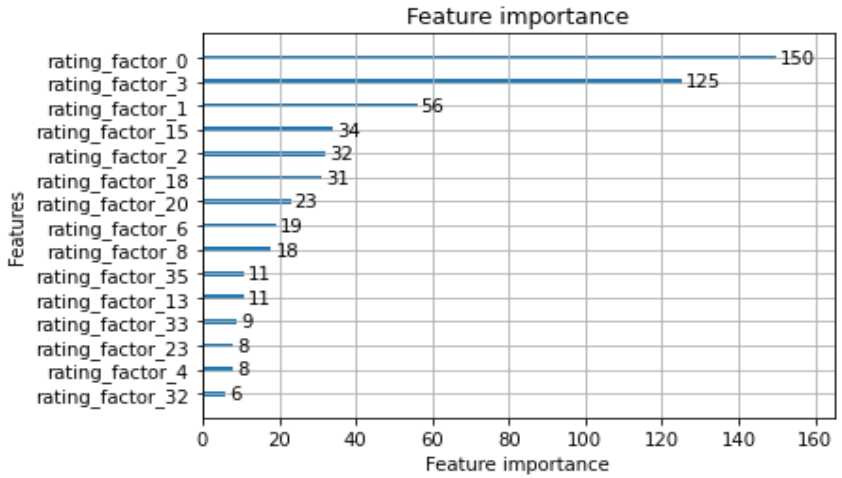
- Bridge 3: actuarial thinking can improve machine learning

Motivating example

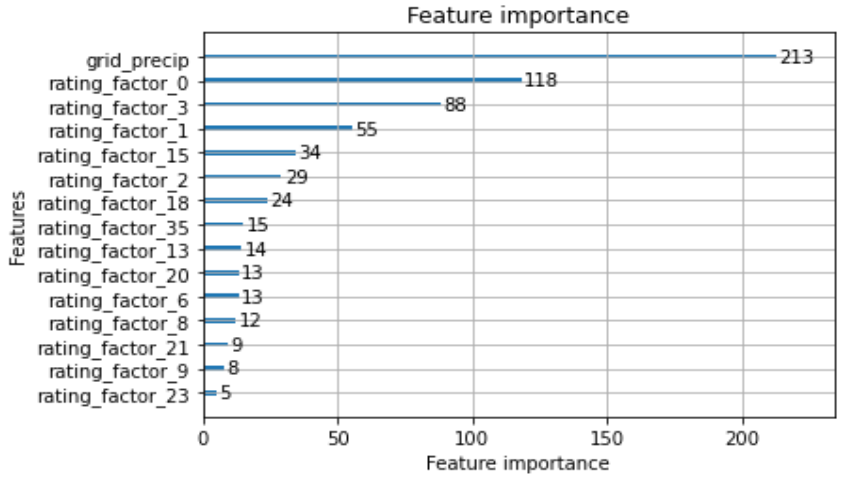
- Motivating example - can we link climate data (external) with claims experience (internal)?
- Allow us to model and project forward climate scenarios at a micro level
- Large property insurance dataset linked with precipitation data over about 10 years
- Model using Gradient Boosting Machine (GBM)
- **Results look great!**



Traditional Pricing Dataset

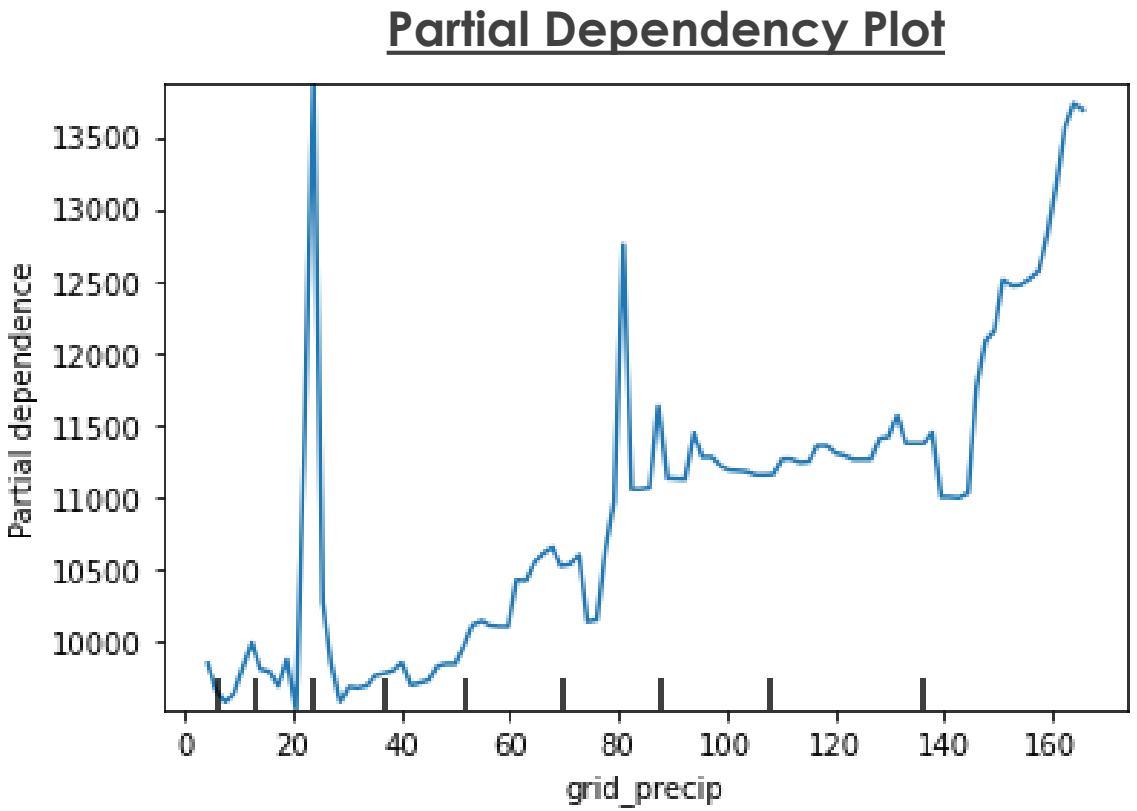


With Actual Precipitation Data



ML gone wrong!

- Clear increasing relationship between precipitation and claims severity…
- … but unreasonable from an actuarial perspective
- GBMs are one of the easiest ways to get accurate models on tabular data
- Could impose a monotonic penalty BUT would still be quite spiky
- **We need actuarial ML!**
- **How can we do machine learning while staying true to our actuarial thinking?**



DNA of Actuarial Modelling

- **Actuarial thinking is…**

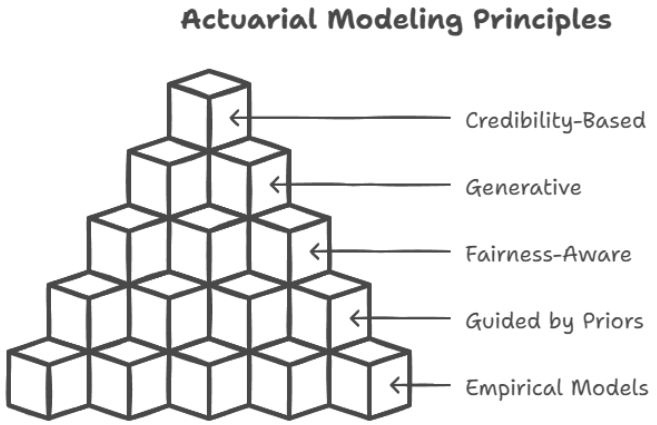
- **Empirical models**
  - We build models based on observed experience… not based on abstract theory
  - Actuarial modelling in a nutshell: **study of observed regularities**

- **Guided by priors (inductive biases)**
  - Impose order on data through domain knowledge: monotonic and smooth effects
  - Use judgement to set assumptions when we cannot rely on data

- **Fairness-aware**
  - Allocation of costs - premium equals expected loss + risk load, no hidden cross-subsidy
  - Prudence - recognise downside skew; hold capital for tail risk

- **Generative**
  - Produce future loss scenarios, capital projections and stress tests
  - Ask "what if?" as often as "what is?"

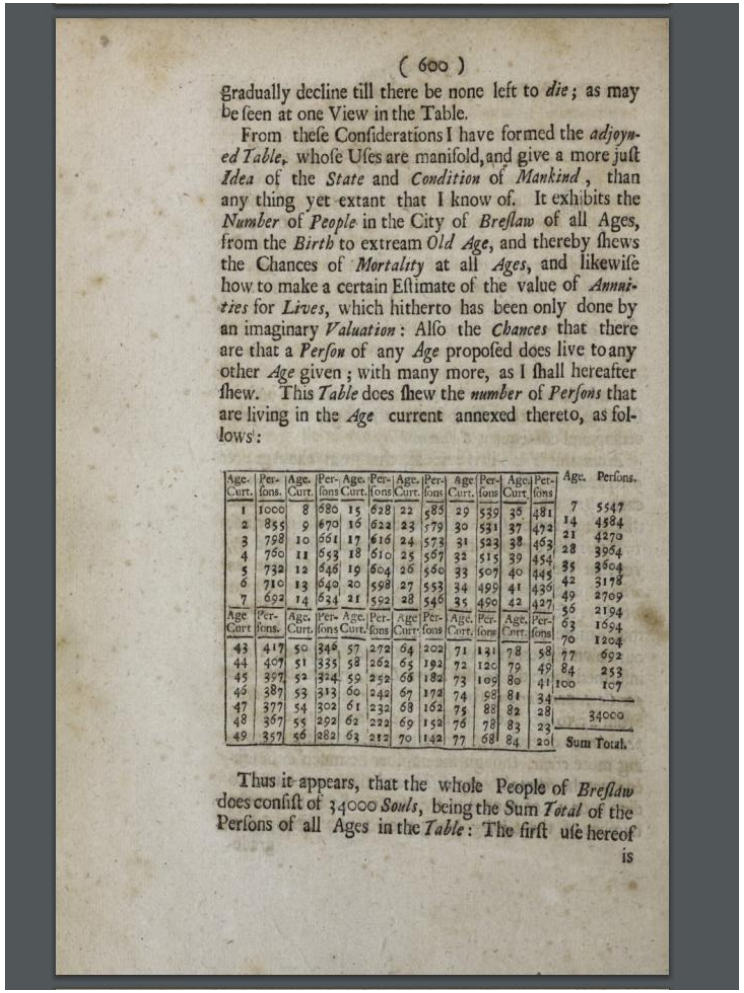
- **Credibility-based**
  - Blend individual and collective experience with formal weights
  - "Big data speaks loudly, small data gets a whisper" - Bühlmann credibility in practice



The first actuarial report?

- Halley's 1693 life table was a pure data experiment - count births and funerals, fit survival curves, then price annuities

- **That "let the experience speak" philosophy - actuarial empiricism - is identical to the principles we apply today!**



(600)

gradually decline till there be none left to *die*; as may be seen at one View in the Table.

From these Considerations I have formed the *adjoyned Table*, whose Uses are manifold, and give a more just *Idea* of the *State* and *Condition* of *Mankind*, than any thing yet extant that I know of. It exhibits the *Number* of *People* in the City of *Breslaw* of all Ages, from the *Birth* to extream *Old Age*, and thereby shews the Chances of *Mortality* at all *Ages*, and likewise how to make a certain Estimate of the value of *Annuities* for *Lives*, which hitherto has been only done by an imaginary *Valuation*: Also the *Chances* that there are that a *Person* of any *Age* proposed does live to any other *Age* given; with many more, as I shall hereafter shew. This *Table* does shew the *number* of *Persons* that are living in the *Age* current annexed thereto, as follows:

Age. Curt.	Persons.	Age. Curt.	Persons	Age. Curt.	Persons	Age. Curt.	Persons	Age. Curt.	Persons	Age. Curt.	Persons
1	1000	8	680	15	628	22	585	29	539	36	481
2	855	9	670	16	622	23	579	30	531	37	472
3	798	10	661	17	616	24	573	31	523	38	463
4	760	11	653	18	610	25	567	32	515	39	454
5	732	12	646	19	604	26	560	33	507	40	445
6	710	13	640	20	598	27	553	34	498	41	436
7	692	14	634	21	592	28	546	35	490	42	427
Age Curt	Persons.	Age. Curt.	Persons	Age. Curt.	Persons	Age Curt	Persons	Age. Curt.	Persons	Age. Curt.	Persons
43	417	50	346	57	272	64	202	71	131	78	58
44	407	51	335	58	262	65	192	72	120	79	49
45	397	52	324	59	252	66	182	73	109	80	41
46	387	53	313	60	242	67	172	74	98	81	34
47	377	54	302	61	232	68	162	75	88	82	28
48	367	55	292	62	222	69	152	76	78	83	23
49	357	56	282	63	212	70	142	77	68	84	20

Age.	Persons.
7	5547
14	4584
21	4270
28	3964
35	3604
42	3178
49	2709
56	2194
63	1694
70	1204
77	692
84	253
100	107
	34000
	Sum Total.

Thus it appears, that the whole People of *Breslaw* does consist of 34000 *Souls*, being the Sum *Total* of the Persons of all Ages in the *Table*: The first use hereof is

Exemplar 1: Lee-Carter

- Forecast mortality rates = key inputs into demographic forecasting, life insurance and pensions models

- Foundational model for mortality forecasting is the Lee-Carter model (Lee and Carter 1992) (LC model)

- Mortality over time modelled using:

$$\log\left(u_{x,t}\right) = a_x + b_x k_t$$

- Relies on latent variables that must be estimated from data and then multiplied
  - Use non-linear/PCA regression to estimate the latent terms (Brouhns, Denuit and Vermunt 2002; Currie 2016; Lee and Carter 1992)

- Time index *k* extrapolated using time series methods

- **Purely empirical - describes how mortality evolves over time, not why!**

Exemplar 2: Cairns-Blake-Dowd model

- Different approach = CBD model (2008)
- Rather than estimate latent factors underlying mortality rate evolution over time…
- … fit Gompertz curves in each observed period and model how these curves vary:

$$\text{logit}\,(q_{x,t}) = \kappa_t^{(1)} + \kappa_t^{(2)}(x - \bar{x})$$

Where :

- $\kappa_t^{(1)}$= overall level of old-age mortality in year t
- $\kappa_t^{(2)}$= steepness of the age gradient in year t
- $x - \bar{x}$= centred age, so the model is stable and interpretable

- Can easily draw analogies to fitting models to experience across mortality/morbidity/lapse
- AvE adjustments to standard tables are also a simpler form of empirical modelling

Exemplar 3: Chain-ladder

- Basic starting point for IBNR analysis is the chain-ladder method
- Totally empirical: assumes the same pattern of claims development observed in the past will be observed in the future
- Application: if R10 became R20 after several years of development, then R15 will become R30 after the same development

	lag																			
AY	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	
201003	1.58	1.06	0.96	0.98	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	
201006	1.32	0.95	0.99	1.00	1.01	0.97	1.01	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00		
201009	1.20	0.95	1.00	1.00	0.99	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00			
201012	1.36	1.00	0.99	0.98	1.00	1.00	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00				
201103	1.27	0.98	0.96	1.01	0.99	0.99	0.99	0.99	1.01	0.99	1.00	0.99	1.00	1.00	1.00					
201106	1.37	0.98	0.97	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00						
201109	1.34	0.97	0.99	0.99	0.98	0.99	1.00	1.01	1.00	0.99	1.00	1.00	1.00							
201112	1.53	1.00	1.00	0.99	1.00	1.00	1.00	1.00	1.00	1.00	1.01	1.00								
201203	1.64	1.01	0.99	1.01	1.01	0.99	1.00	0.99	1.00	1.00	1.00									
201206	1.37	0.94	1.00	1.00	0.99	0.99	0.99	1.00	0.99	1.00										
201209	1.43	1.01	0.97	0.99	0.98	0.98	1.00	0.97	1.01											
201212	1.53	1.02	0.99	1.00	0.98	1.00	0.99	1.01												
201303	1.51	0.99	1.00	1.00	0.97	0.99	1.00													
201306	1.32	0.98	1.00	1.00	1.00	1.00														
201309	1.47	0.97	0.99	0.99	1.00															
201312	1.61	1.05	1.01	0.99																
201403	1.81	1.07	1.02																	
201406	1.59	1.09																		
201409	1.54																			
0	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	***10***	***11***	***12***	***13***	***14***	***15***	***16***	***17***	***18***	***19***	
x(t+1)	589 739	558 111	514 636	473 693	424 571	394 007	366 064	326 707	285 135	252 371	225 237	195 089	162 469	136 443	114 291	88 274	64 444	46 689	24 097	
x	401 732	554 920	519 425	476 257	428 222	397 168	367 697	327 558	285 304	252 853	225 187	195 482	162 615	136 529	114 381	88 291	64 444	46 689	24 097	
LDF	1.468	1.006	0.991	0.995	0.991	0.992	0.996	0.997	0.999	0.998	1.000	0.998	0.999	0.999	0.999	1.000	1.000	1.000	1.000	
% to Ult	71%	104%	105%	104%	103%	102%	101%	101%	101%	101%	100%	100%	100%	100%	100%	100%	100%	100%	100%	100%

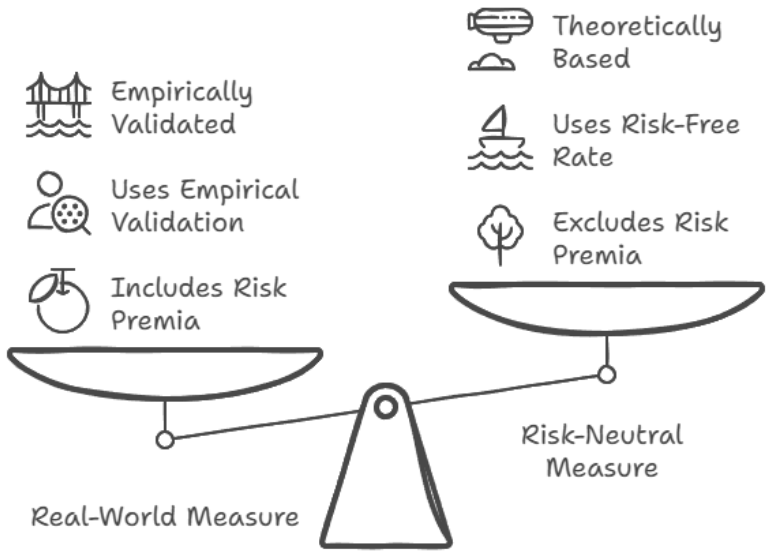
Mind your P's and Q's

- **Actuarial work classically is in the real-world measure (P)**
  - We allow for risk premia in our projections…
  - … and do not necessarily calibrate to market prices (usually don't exist)
  - Real-world models are built and validated empirically!

- **Modern actuarial valuation takes place in the risk-neutral measure (Q)**
  - We do not allow for risk premia (even though we know these exist)
  - We will rather discount using the risk-free rate

- A quick theoretical argument:
  - In the Q-measure, we believe or assume that we can find assets to hedge away risk…
  - … therefore we should not expect to get any returns above risk-free

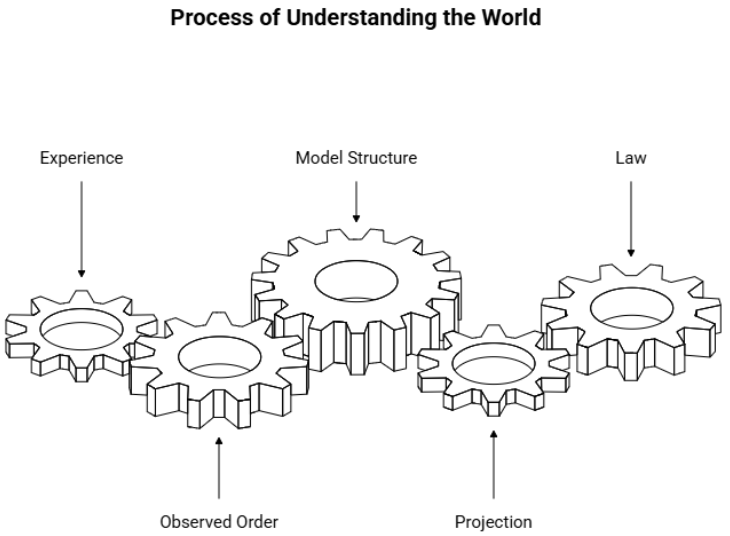
- **Q-measure / risk-neutral models are based on theoretical assumptions, not empiricism!**



Comparing Actuarial Valuation Measures

Order and Law

- **Actuarial work utilizes regular patterns observed empirically in data..**
- **... and extrapolates these into the near or distant future**
- **If the regularity is stable enough, we start calling it a "law":**
  - "an observed regularity or pattern" (Thatcher 1999)
  - "a fundamental law of nature" (Kirkwood 2015)
- **In mortality, a law may be an observed pattern, or possibly something deeper about ageing itself.**
- **Do we need laws for actuarial work?**
  - Most of the time, we can rely on disciplined empiricism.
  - But laws help when we need to extrapolate beyond directly observed experience.
- **A modern analogue is emerging in AI: scaling laws.**



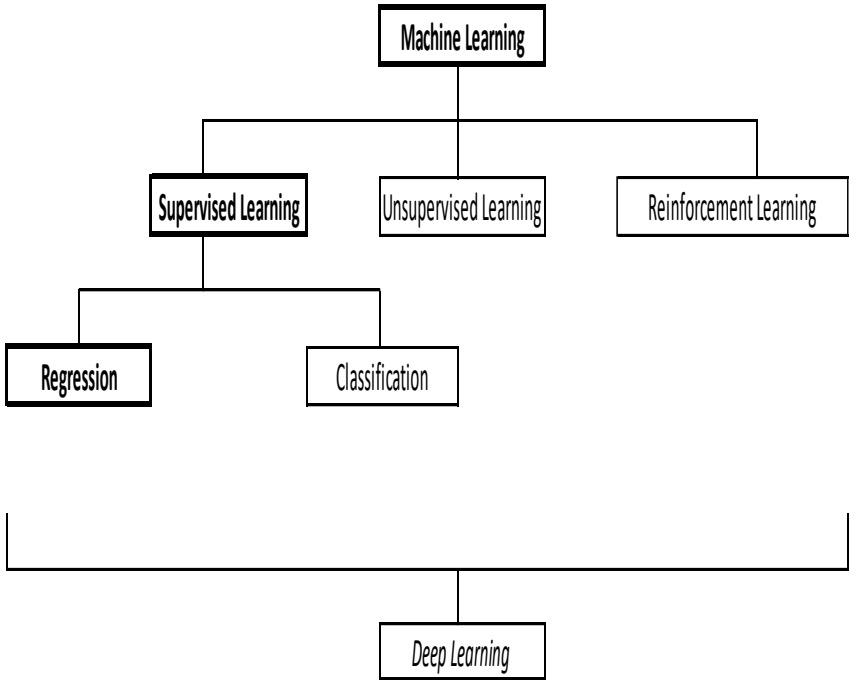
ML is another empirical discipline!

- Machine Learning: "the study of algorithms that allow computer programs to automatically improve through experience" (Mitchell 1997)

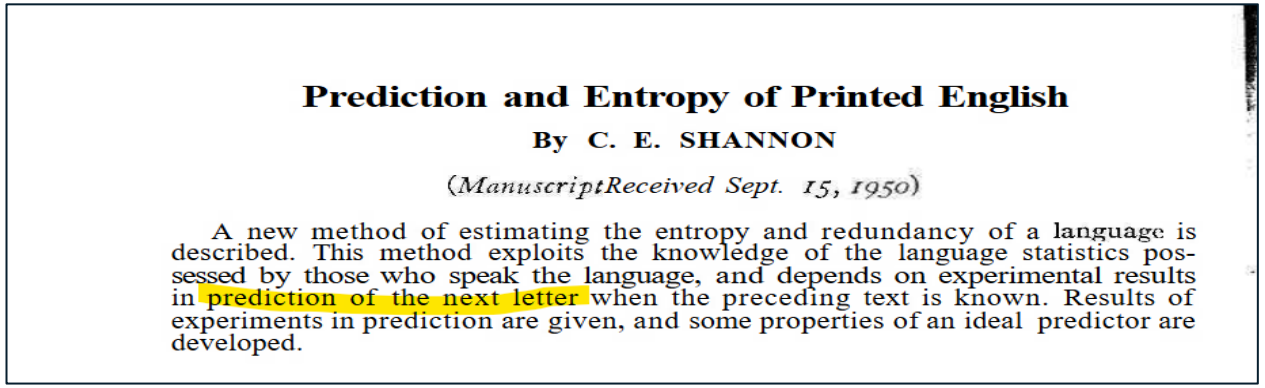
- Machine learning is an approach to the field of Artificial Intelligence
  - Systems trained to recognise patterns within data to acquire knowledge (Goodfellow, Bengio and Courville 2016)

- **ML Paradigm - feed data to the machine and let it figure out what is important from the data!**

- **ML mirrors the actuarial style of thinking**



Can we project the next word?

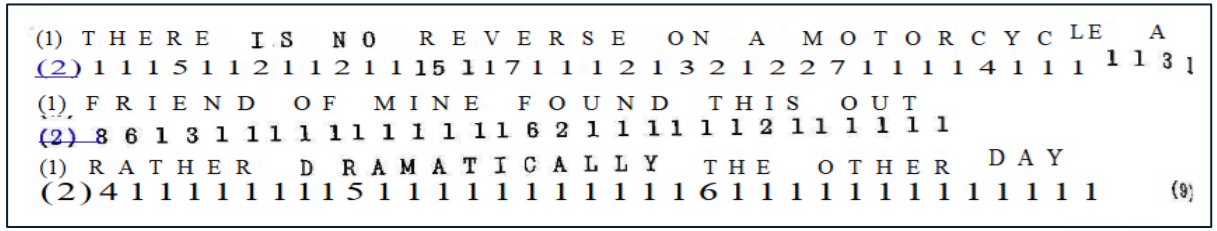


Prediction and Entropy of Printed English

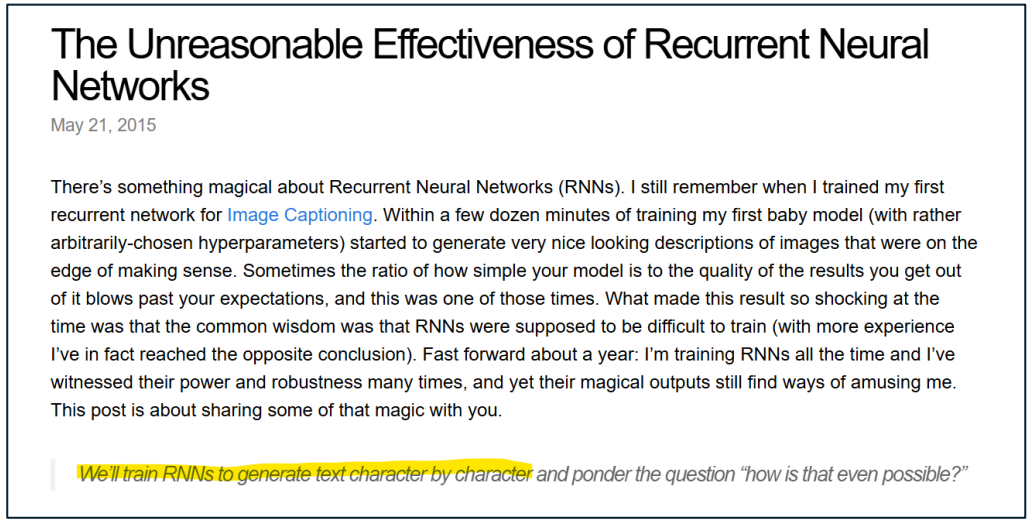
By C. E. SHANNON

(*Manuscript Received Sept. 15, 1950*)

A new method of estimating the entropy and redundancy of a language is described. This method exploits the knowledge of the language statistics possessed by those who speak the language, and depends on experimental results in prediction of the next letter when the preceding text is known. Results of experiments in prediction are given, and some properties of an ideal predictor are developed.



(1) THERE IS NO REVERSE ON A MOTORCYCLE A
(2) 1 1 1 5 1 1 2 1 1 2 1 1 15 1 1 7 1 1 1 2 1 3 2 1 2 2 7 1 1 1 1 4 1 1 1 1 1 3 1
(1) FRIEND OF MINE FOUND THIS OUT
(2) 8 6 1 3 1 1 1 1 1 1 1 1 1 1 1 1 6 2 1 1 1 1 1 1 2 1 1 1 1 1 1
(1) RATHER DRAMATICALLY THE OTHER DAY
(2) 4 1 1 1 1 1 1 1 1 5 1 1 1 1 1 1 1 1 1 1 1 1 1 6 1 1 1 1 1 1 1 1 1 1 1 1 1 1 (9)

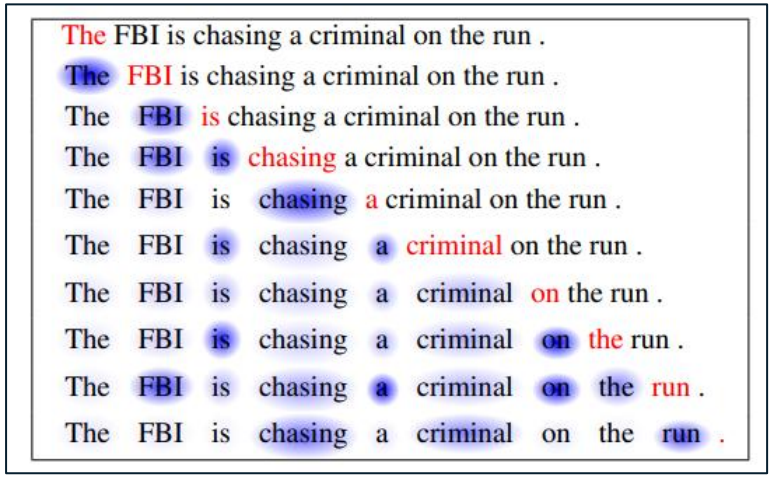


The Unreasonable Effectiveness of Recurrent Neural Networks

May 21, 2015

There's something magical about Recurrent Neural Networks (RNNs). I still remember when I trained my first recurrent network for Image Captioning. Within a few dozen minutes of training my first baby model (with rather arbitrarily-chosen hyperparameters) started to generate very nice looking descriptions of images that were on the edge of making sense. Sometimes the ratio of how simple your model is to the quality of the results you get out of it blows past your expectations, and this was one of those times. What made this result so shocking at the time was that the common wisdom was that RNNs were supposed to be difficult to train (with more experience I've in fact reached the opposite conclusion). Fast forward about a year: I'm training RNNs all the time and I've witnessed their power and robustness many times, and yet their magical outputs still find ways of amusing me. This post is about sharing some of that magic with you.

> *We'll train RNNs to generate text character by character and ponder the question "how is that even possible?"*



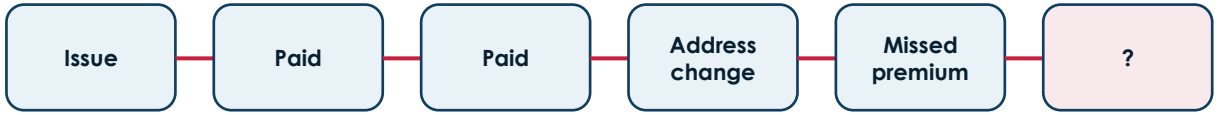
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .
The FBI is chasing a criminal on the run .

Karpathy: "The Unreasonable Effectiveness of Recurrent Neural Networks"

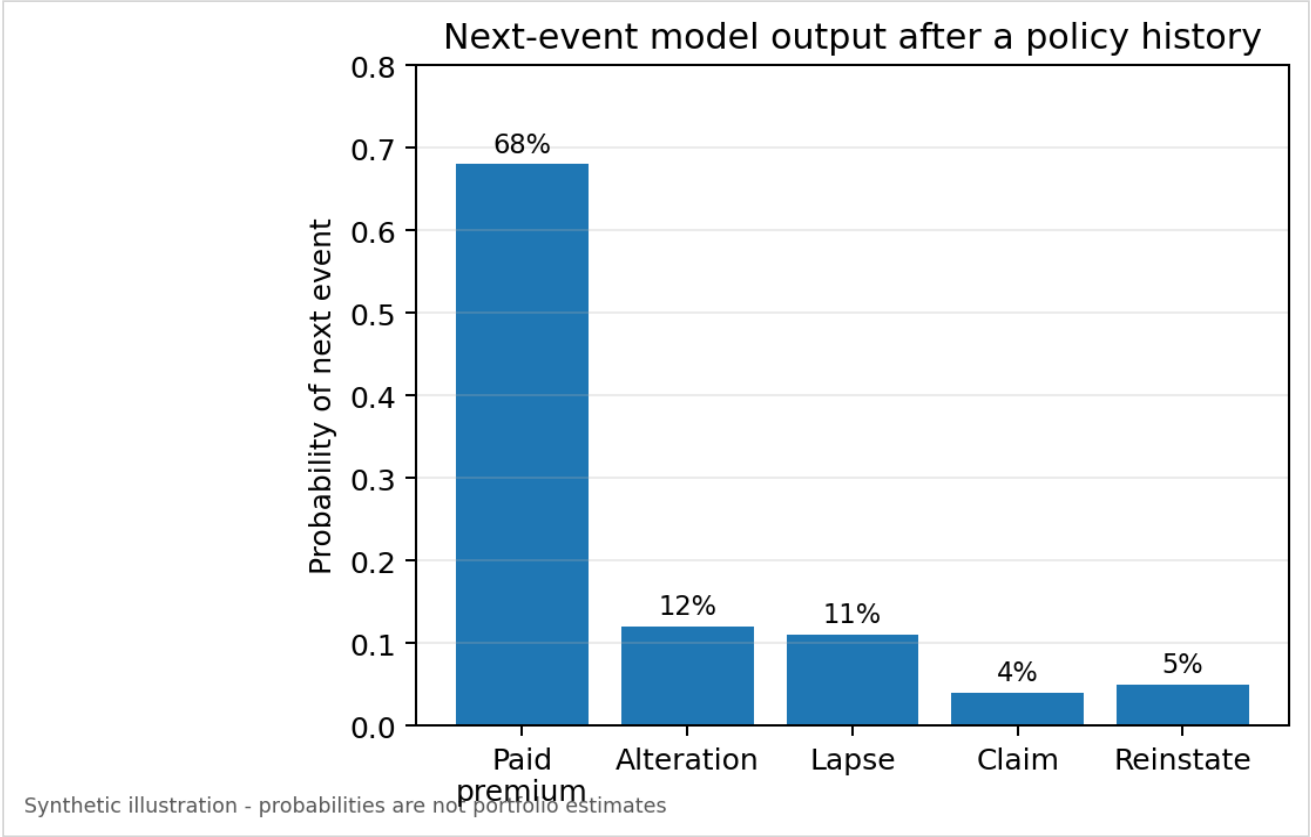
Character-level RNN learning what to predict next

Can we project the next policy event?

- A policy history is a sequence, just like text.



- Given previous events, predict the next event:
  - Paid premium
  - Alteration
  - Lapse
  - Claim
  - Reinstatement

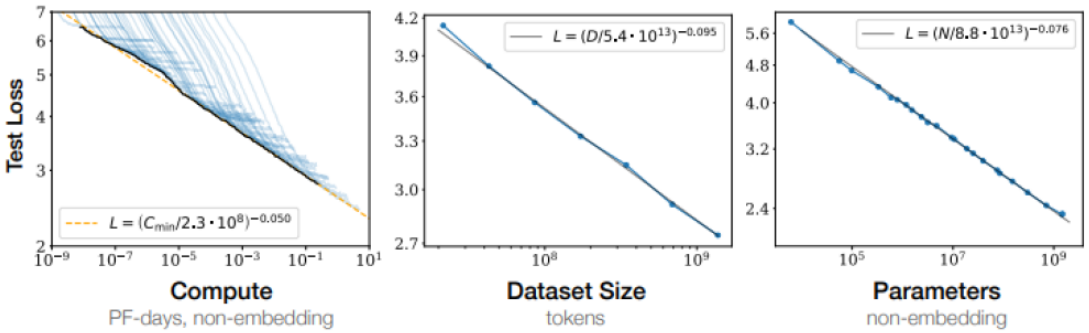


- This is the life assurance version of next-token prediction. Instead of asking:
  "What word comes next?"
  we ask:
  "What policy event comes next?"
- Life = sequence of decisions, behaviours and risk events - exactly the kind of structure modern ML is built to learn.

Scaling towards AI

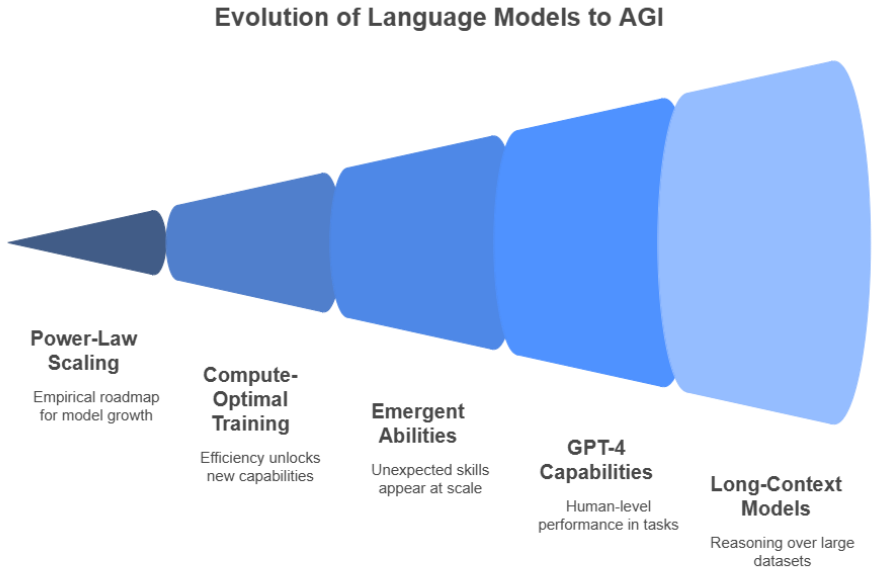
- **As we scale up empirical modelling of the next word, we get highly intelligent models**

- One can find "laws" that explain how models will likely perform as we increase data, compute or parameters



- Power-law scaling gives us an empirical roadmap from "toy" LLMs to AI models

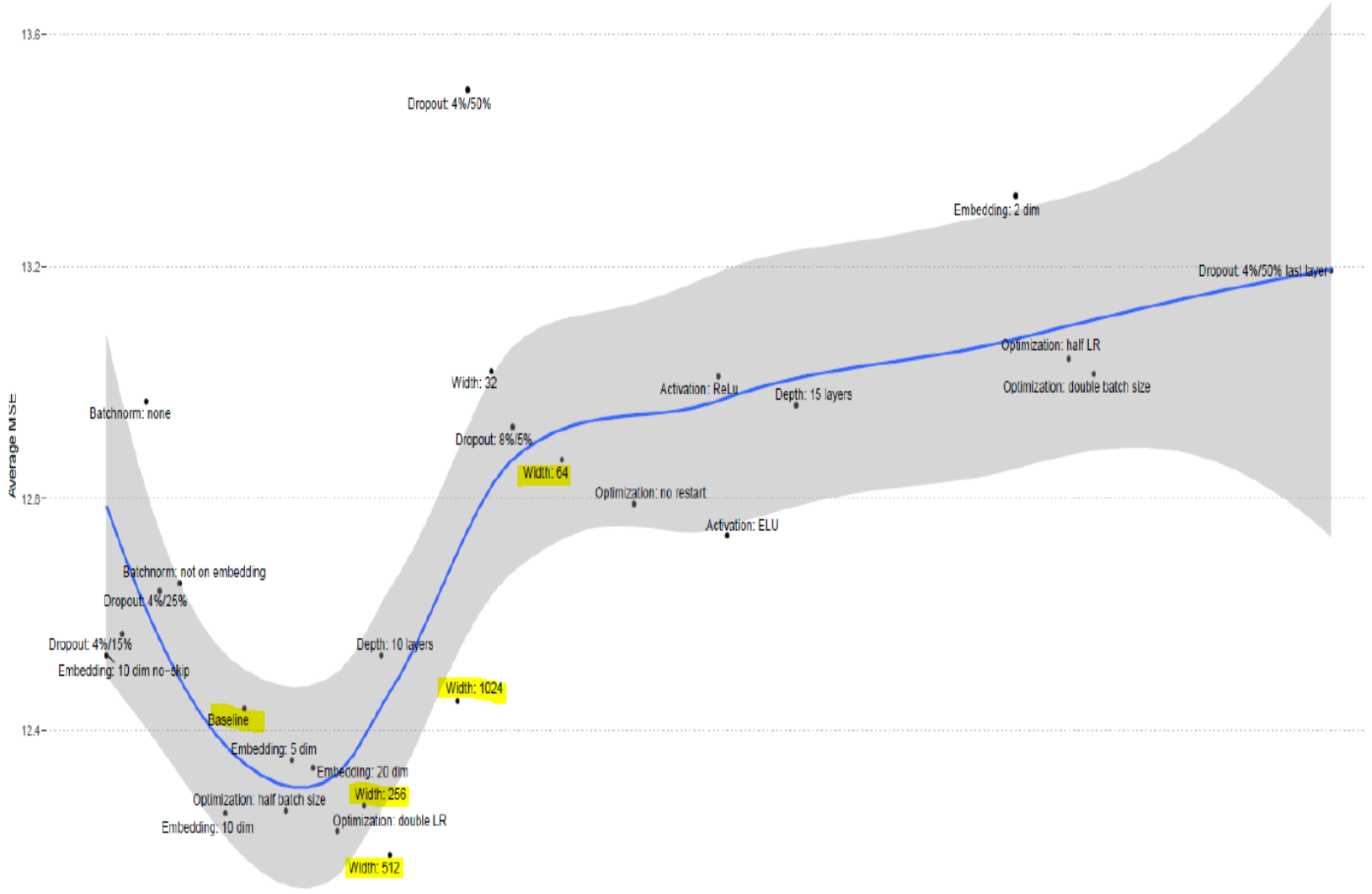
- Above certain scale thresholds, emergent abilities appear unexpectedly:
  - Zero- or few-shot inference
  - Reasoning chains of thought
  - Ability to use tools



- Our current long-context, reasoning models push the horizon from "predict the next sentence" to "ingest or rewrite entire codebases"

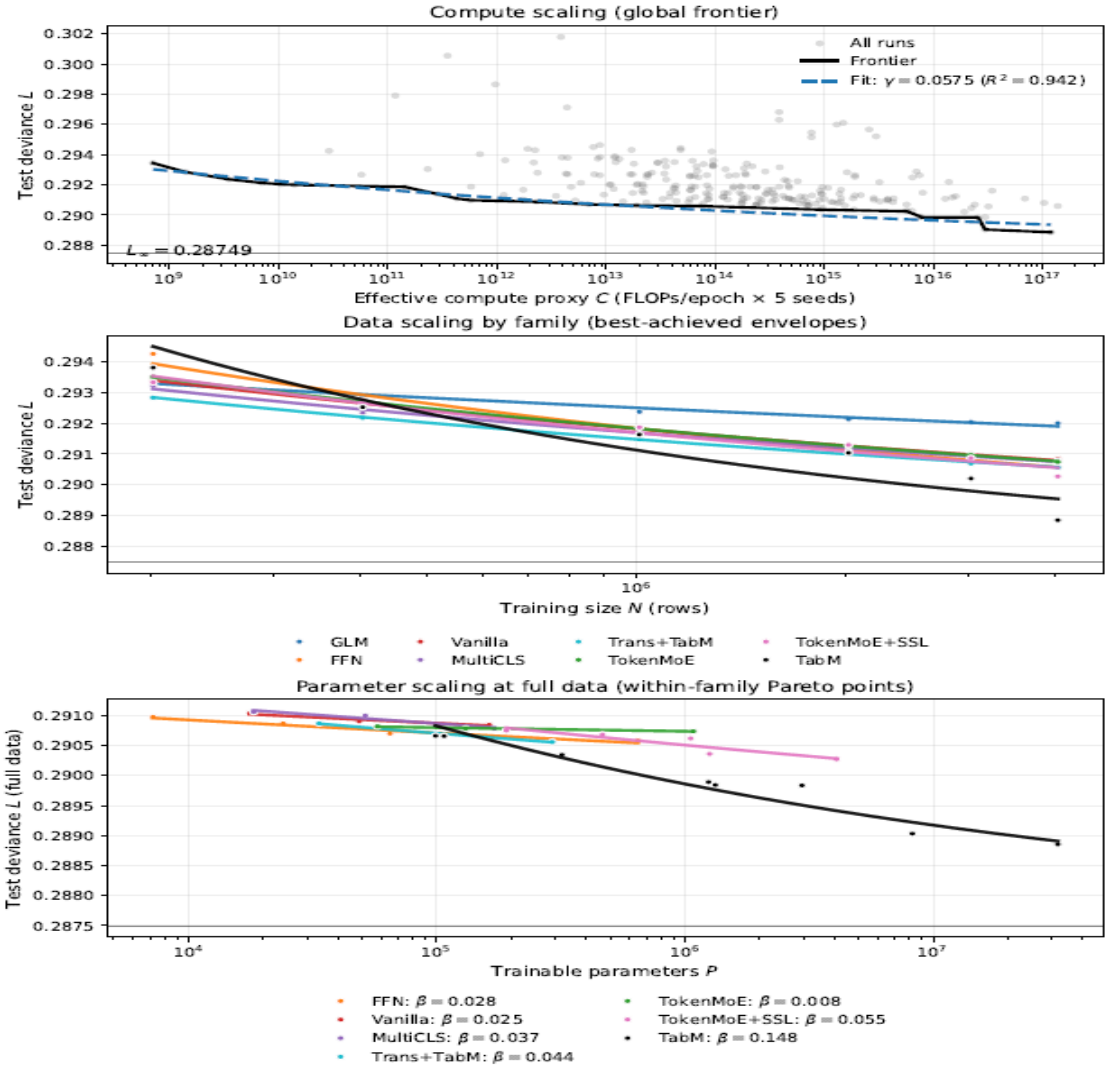
Do scaling laws apply in actuarial science?

- **Actuarial modelling often is done at much smaller sizes than language modelling…**
- **… so it is unclear whether similar scaling could occur in actuarial modelling**
- **Other differences:**
  - much lower signal-to-noise ratio
  - time trends
  - different types of predictors
- **A first hint of scaling – as one increases parameters in a neural network mortality forecasting model, the loss decreases up to a point!**



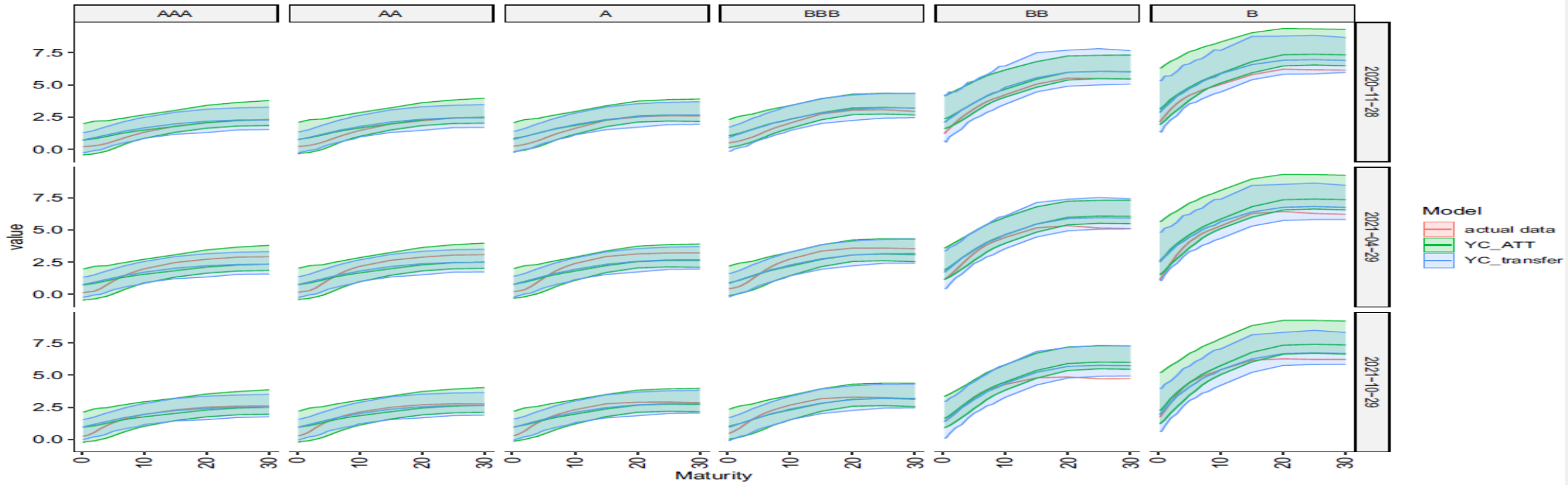
Actuarial models can scale!

- **Experiment on a large claim modelling problem with claim frequency as a target**
- **~= 4.5m rows of experience**
- **Fit different:**
  - families of models
  - sizes of models
  - proportions of the data
- **One can predict model performance quite well => scaling laws apply to actuarial models too**
- **Not all families of models scale as well**



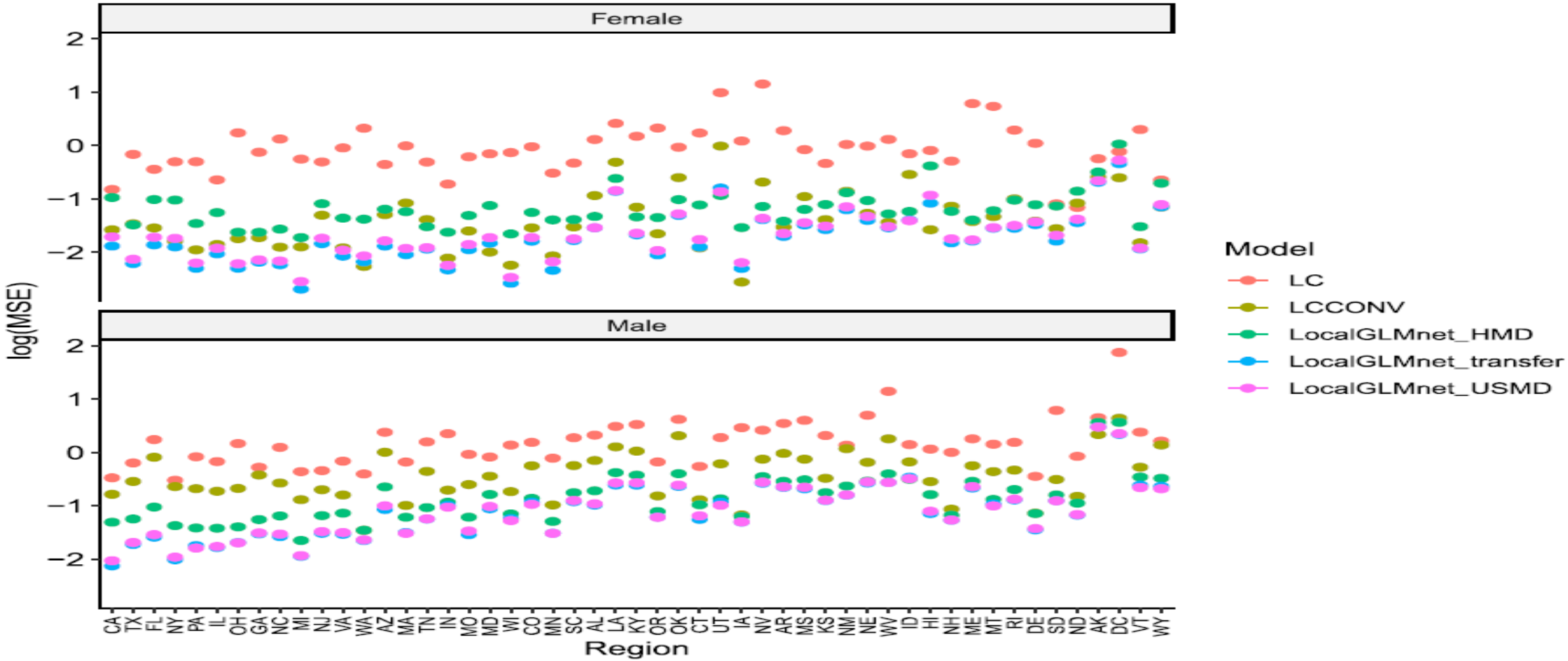
Emergent capabilities of large actuarial models

- As we scale up actuarial models, these models become good at transfer learning…
- … i.e. the ability to perform well on tasks the models have not seen before!
- Example 1 – train models on govi yield curves then refit to corporate curves



Transfer learning in mortality forecasting

- **Example 2 – train models on national mortality data then finetune on sub-national data**



Can actuarial thinking improve ML?

- Discuss a few applications of actuarial thinking within ML

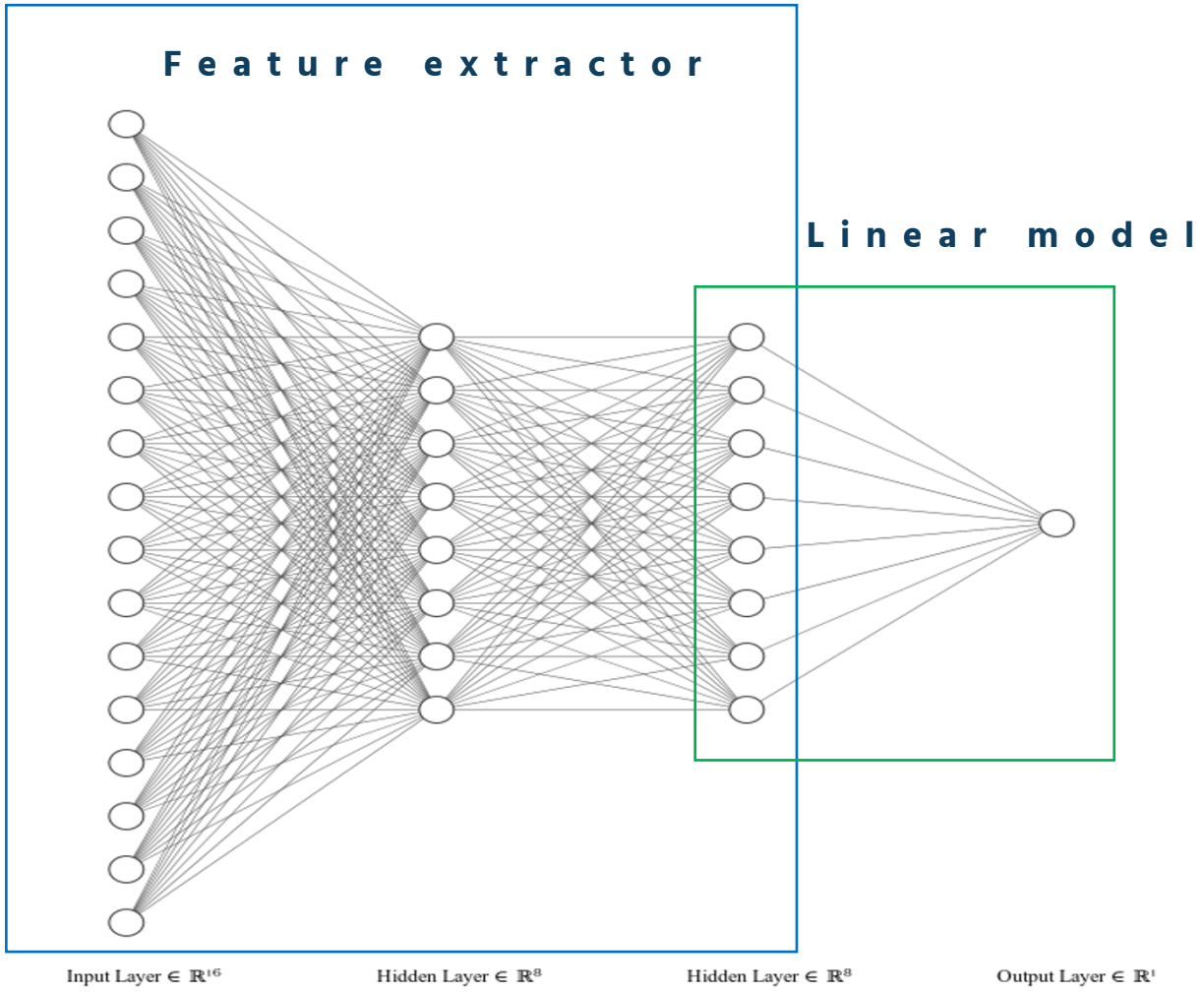
- Can we impose a prior view of "order" on models to improve them?

- **Surprisingly (or not!) these improve the ML models themselves:**
  - Smoothing and monotonicity constraints
  - Credibility modelling of relativities (embeddings)
  - Credibility improvements to Transformers

- **But first, the briefest introduction!**

Neural networks generalise GLMs

- GLMs are the classical actuarial workhorse model
- Can express many actuarial models within the GLM framework
- **The whole neural network framework powering AI is nothing more than a "Generalised GLM"**
  - Intermediate layers = representation learning, guided by supervised objective
  - Last layer = (generalised) linear model, where input variables = new representation of data
- No need to use GLM — strip off last layer and use learned features in, for example, XGBoost
- Or mix with traditional method of fitting GLM



Let's solve our opening problem

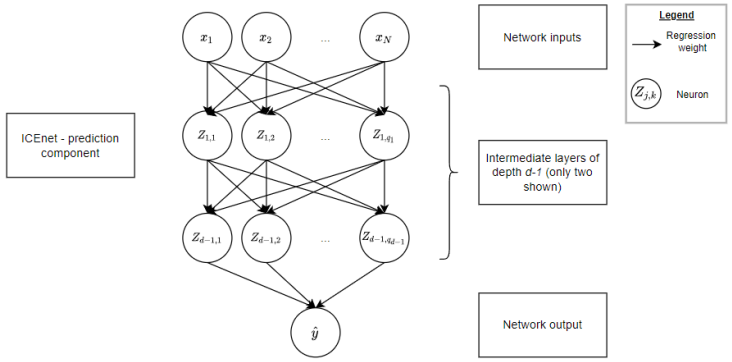
- We know what a good actuarial model should look like…
  - … ideally, predictions should be smooth (unless good reason to expect roughness)
  - … and we can often intuit where relationships should be monotonic
- Take a flexible class of models (like neural networks) and constrain them to meet these criteria
- **The trick — work in the output / prediction space of the networks (ICEnet)**
- **Adding actuarial thinking improves model performance and interpretation**

Description	Learn		Test	
FCN	0.2381	(0.000211)	0.2387	(0.000351)
FCN (nagging)	0.2376		0.2383	
ICEnet	0.2386	(0.000180)	0.2388	(0.000236)
ICEnet (nagging)	0.2384		0.2385	

Description	Constraints	Learn		Test	
ICEnet	Smoothing + Monotonicity	0.2386	(0.000180)	0.2388	(0.000236)
ICEnet	Smoothing	0.2385	(0.000423)	0.2388	(0.000385)
ICEnet	Montonicity	0.2384	(0.000214)	0.2385	(0.000140)
ICEnet (nagging)	Smoothing + Monotonicity	0.2384		0.2385	
ICEnet (nagging)	Smoothing	0.2382		0.2385	
ICEnet (nagging)	Monotonicity	0.2381		0.2382	

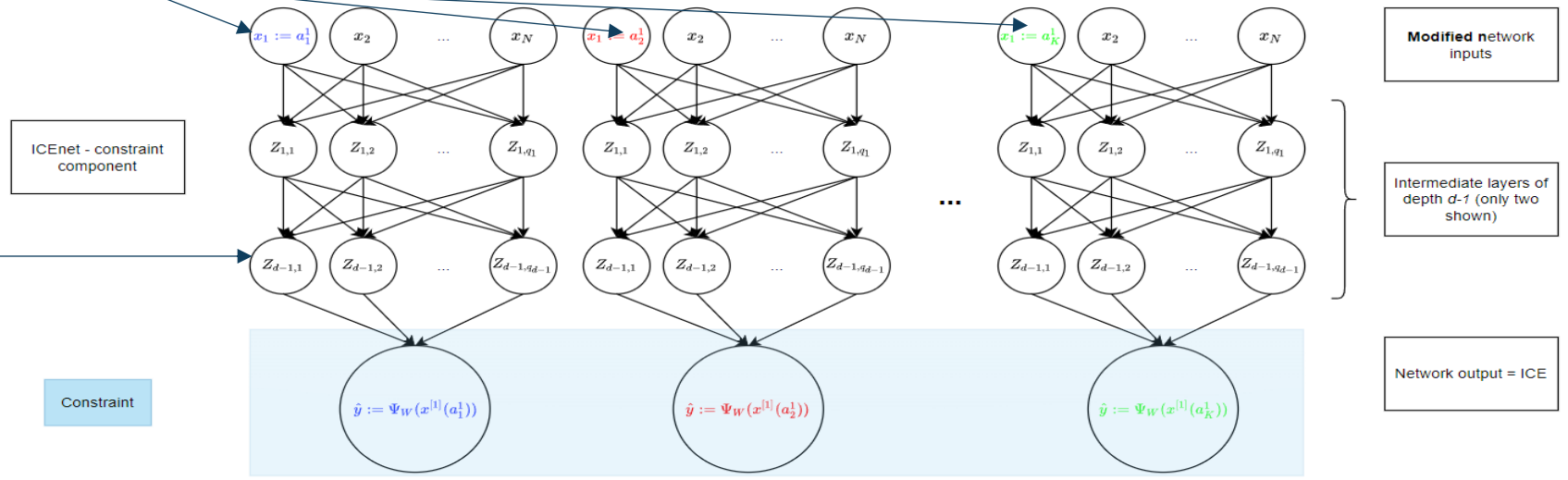
ICEnet

Prediction network

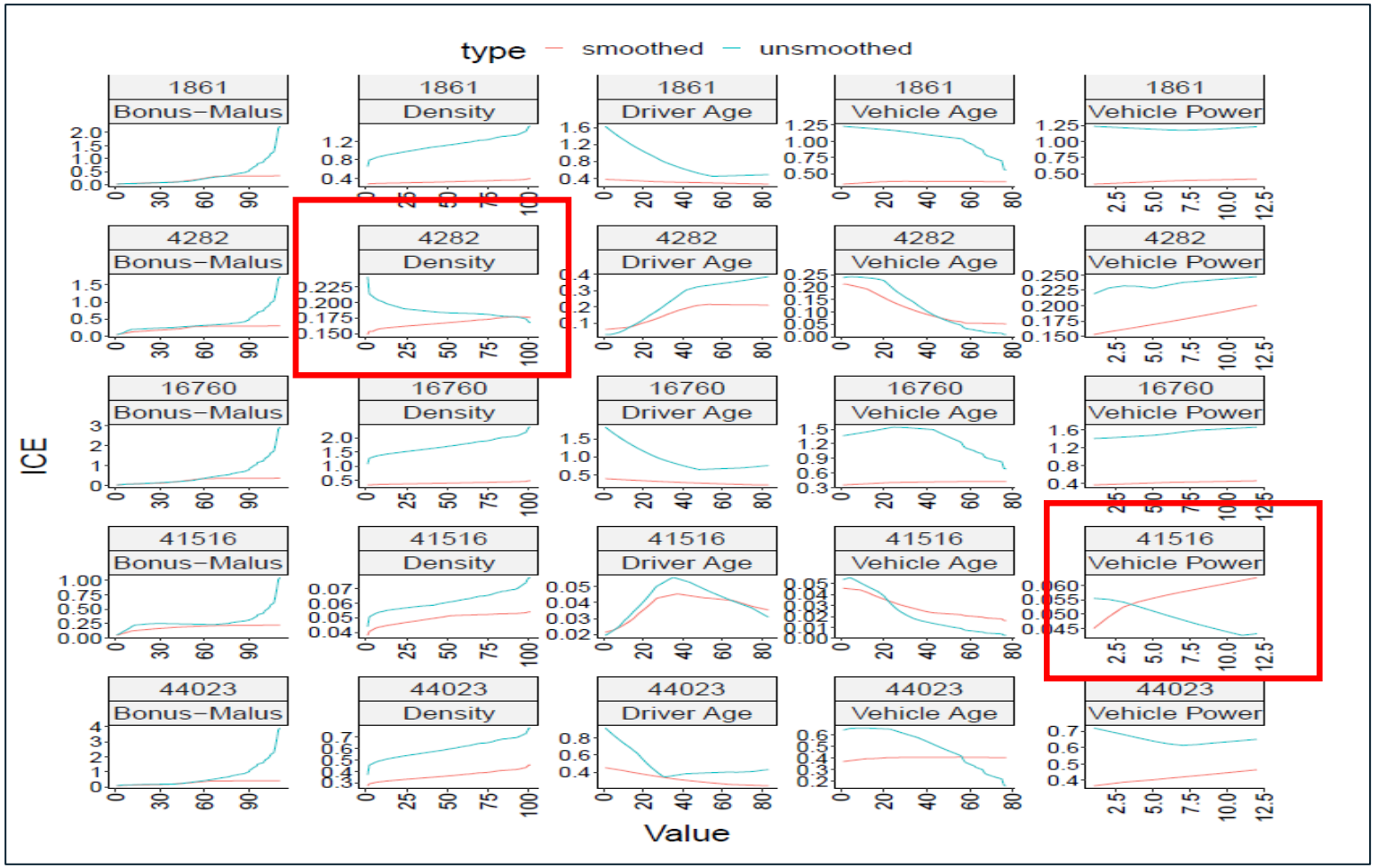


Change value of one variable

Constraint network

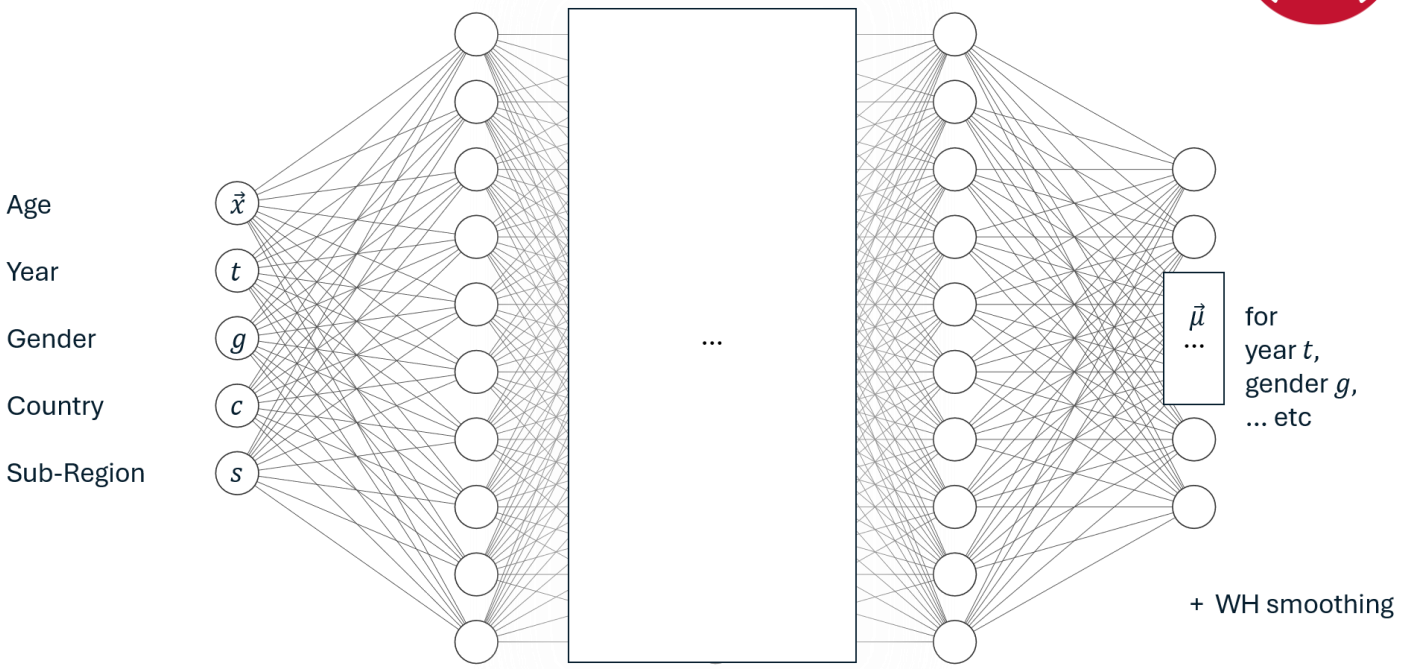


Moving in the right direction



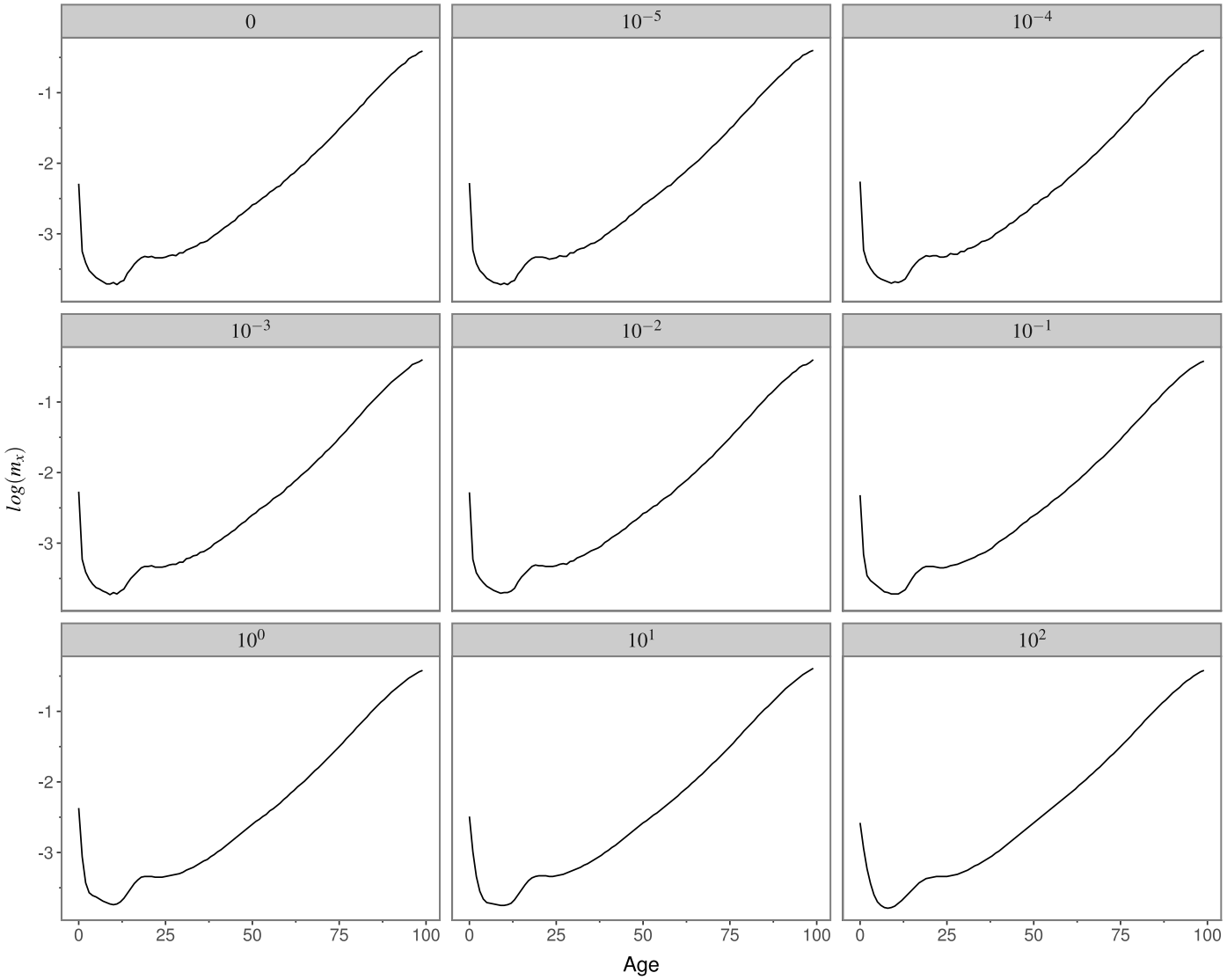
Deep learning with Whittaker-Henderson

- In the life insurance space – can we impose smoothness within neural networks themselves?
- **Output an entire mortality table in one shot and smooth that!**
- **Vector Lee Carter Network**
- **Improves results on national population data!**

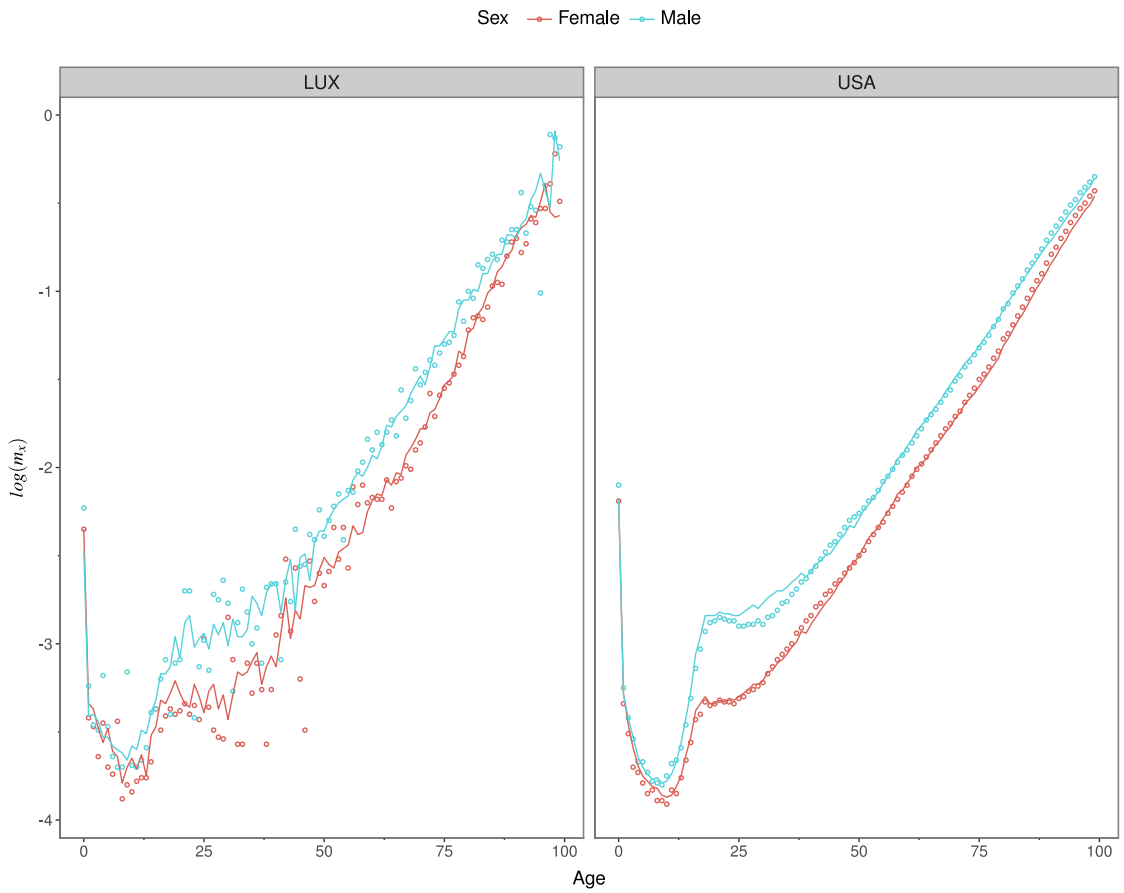


Model	Type[1]	α	Average MSE	Median MSE
Lee-Carter (LC)	National		5.56	2.47
LC	Sub-national		22.10	1.48
LCNN	National		3.05	**1.53**
LCNN	Sub-national		**20.43**	**0.86**
V-LCNN	National	1	**2.91**	1.80
V-LCNN	Sub-national	1	20.66	1.34

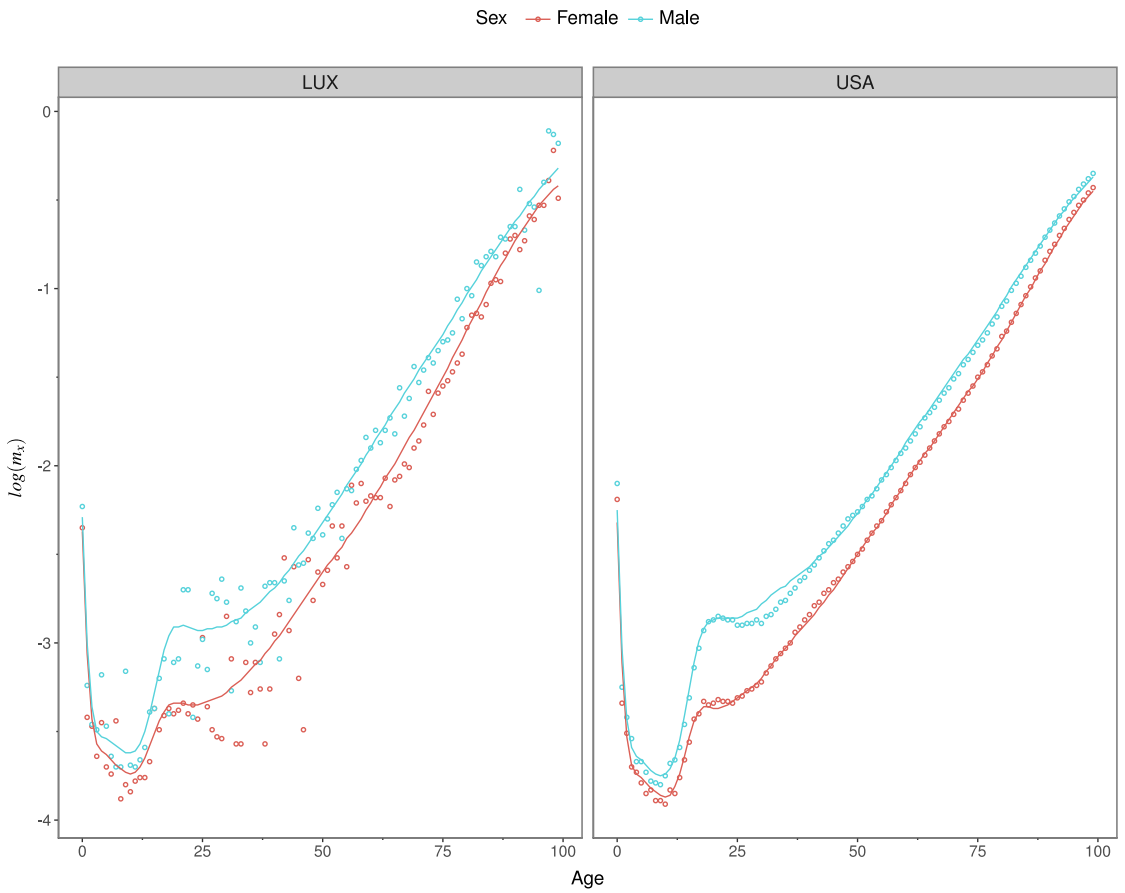
What does WH do?



Comparing models – unconstrained vs constrained

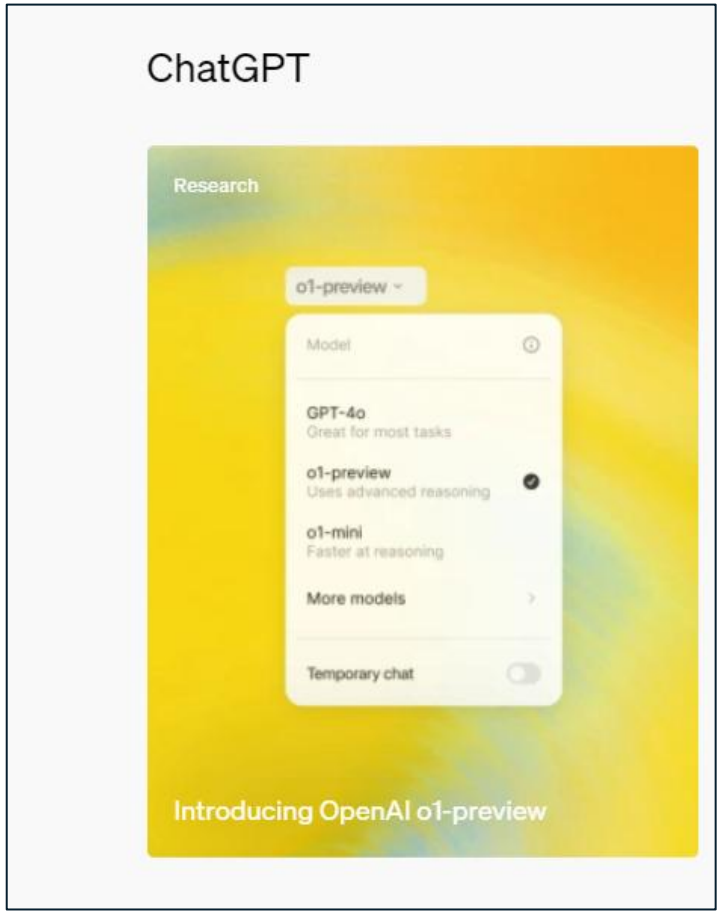


LCNN



V-LCNN ($\alpha = 1$)

Wider field of ensuring “rational results”



Deep Reinforcement Learning from Human Preferences

Paul F Christiano
OpenAI
paul@openai.com

Jan Leike
DeepMind
leike@google.com

Tom B Brown
nottombrown@gmail.com

Miljan Martic
DeepMind
miljanm@google.com

Shane Legg
DeepMind
legg@google.com

Dario Amodei
OpenAI
damodei@openai.com

Abstract

For sophisticated reinforcement learning (RL) systems to interact usefully with real-world environments, we need to communicate complex goals to these systems. In this work, we explore goals defined in terms of (non-expert) human preferences between pairs of trajectory segments. We show that this approach can effectively solve complex RL tasks without access to the reward function, including Atari games and simulated robot locomotion, while providing feedback on less than 1% of our agent's interactions with the environment. This reduces the cost of human oversight far enough that it can be practically applied to state-of-the-art RL systems. To demonstrate the flexibility of our approach, we show that we can successfully train complex novel behaviors with about an hour of human time. These behaviors and environments are considerably more complex than any which have been previously learned from human feedback.

Training language models to follow instructions with human feedback

Long Ouyang* **Jeff Wu*** **Xu Jiang*** **Diogo Almeida*** **Carroll L. Wainwright***
Pamela Mishkin* **Chong Zhang** **Sandhini Agarwal** **Katarina Slama** **Alex Ray**
John Schulman **Jacob Hilton** **Fraser Kelton** **Luke Miller** **Maddie Simens**
Amanda Askell† **Peter Welinder** **Paul Christiano*†**
Jan Leike* **Ryan Lowe***
OpenAI

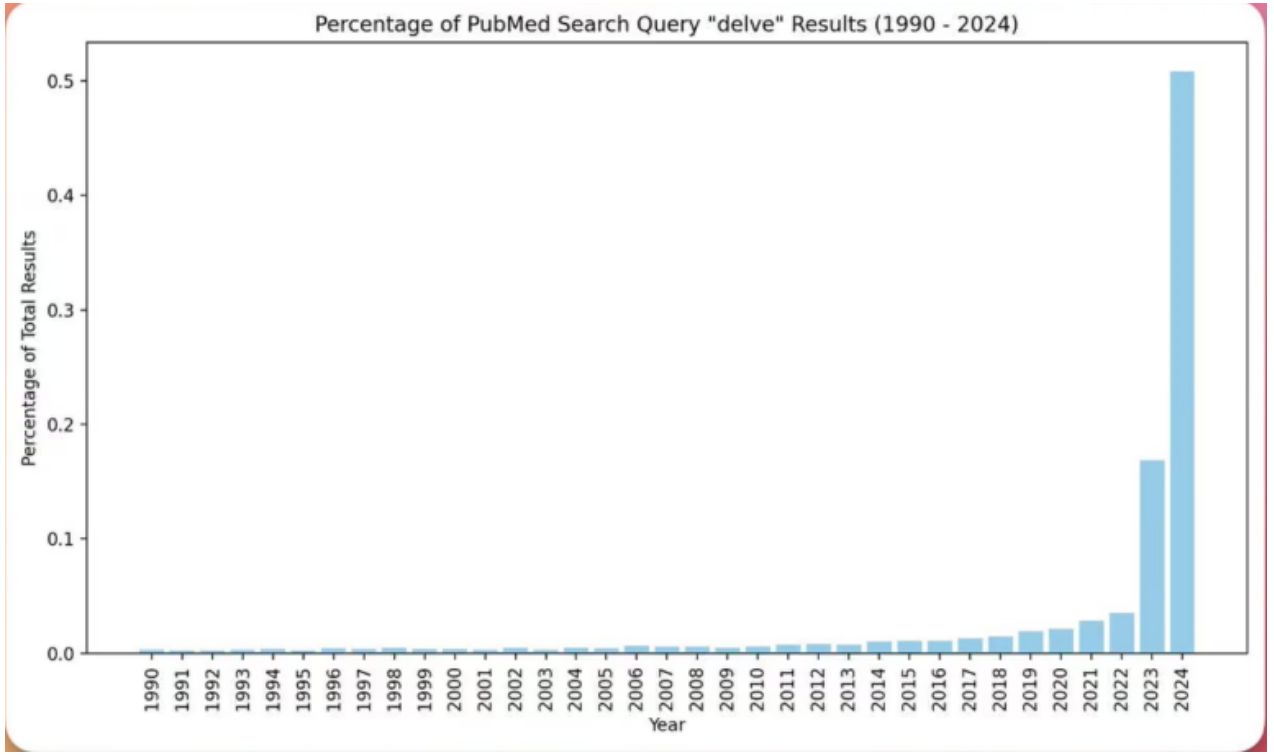
Abstract

Making language models bigger does not inherently make them better at following a user's intent. For example, large language models can generate outputs that are untruthful, toxic, or simply not helpful to the user. In other words, these models are not *aligned* with their users. In this paper, we show an avenue for aligning language models with user intent on a wide range of tasks by fine-tuning with human feedback. Starting with a set of labeler-written prompts and prompts submitted through the OpenAI API, we collect a dataset of labeler demonstrations of the desired model behavior, which we use to fine-tune GPT-3 using supervised learning. We then collect a dataset of rankings of model outputs, which we use to further fine-tune this supervised model using reinforcement learning from human feedback. We call the resulting models *InstructGPT*. In human evaluations on our prompt distribution, outputs from the 1.3B parameter InstructGPT model are preferred to outputs from the 175B GPT-3, despite having 100x fewer parameters. Moreover, InstructGPT models show improvements in truthfulness and reductions in toxic output generation while having minimal performance regressions on public NLP datasets. Even though InstructGPT still makes simple mistakes, our results show that fine-tuning with human feedback is a promising direction for aligning language models with human intent.

Let’s “delve” into this

TechScape: How cheap, outsourced labour in Africa is shaping AI English

Workers in Africa have been exploited first by being paid a pittance to help make chatbots, then by having their own words become AI-ese. Plus, new AI gadgets are coming for your smartphones



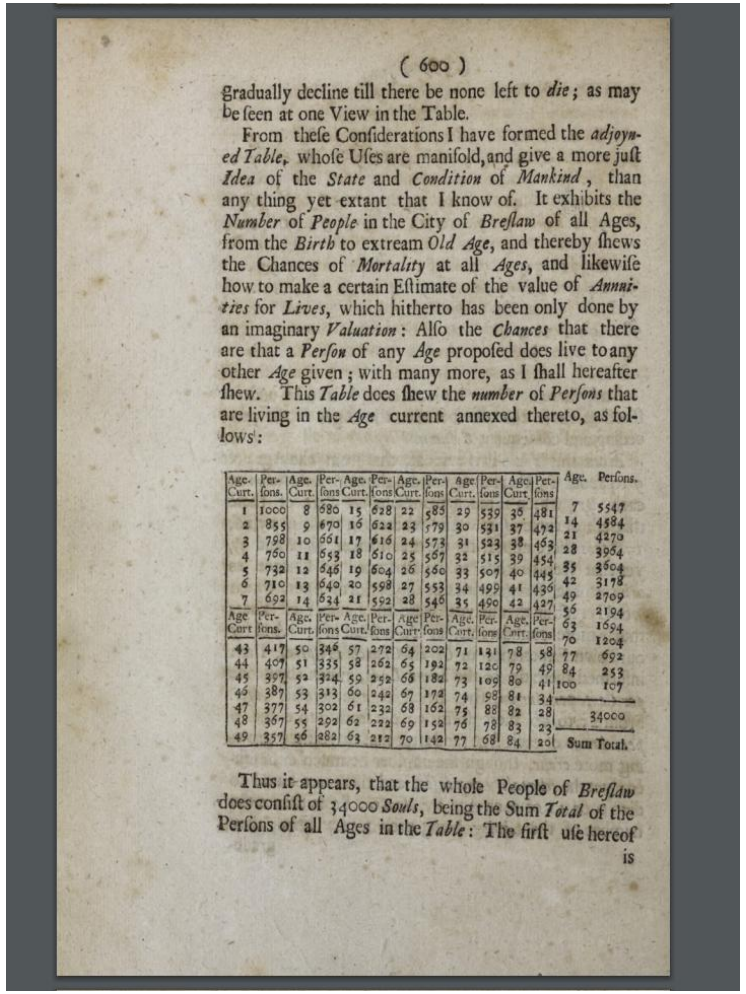
Halley had it right

- **Empirical DNA - Halley didn't start with a theoretical mortality law**
  - He measured it, then priced products straight off the data
  - Explaining mortality laws came much later

- **Same "loop" as language models:**
  - Count death events → infer conditional probabilities → compute expected values

- **Swap "token" for "death event" = blueprint that still powers both actuarial science and next-token LLMs on their march toward AGI**

- Actuaries have been turning raw experience into financial predictions since 1693…
  - … AI is scaling that method from Breslau parish registers to trillion-token corpora



(600)

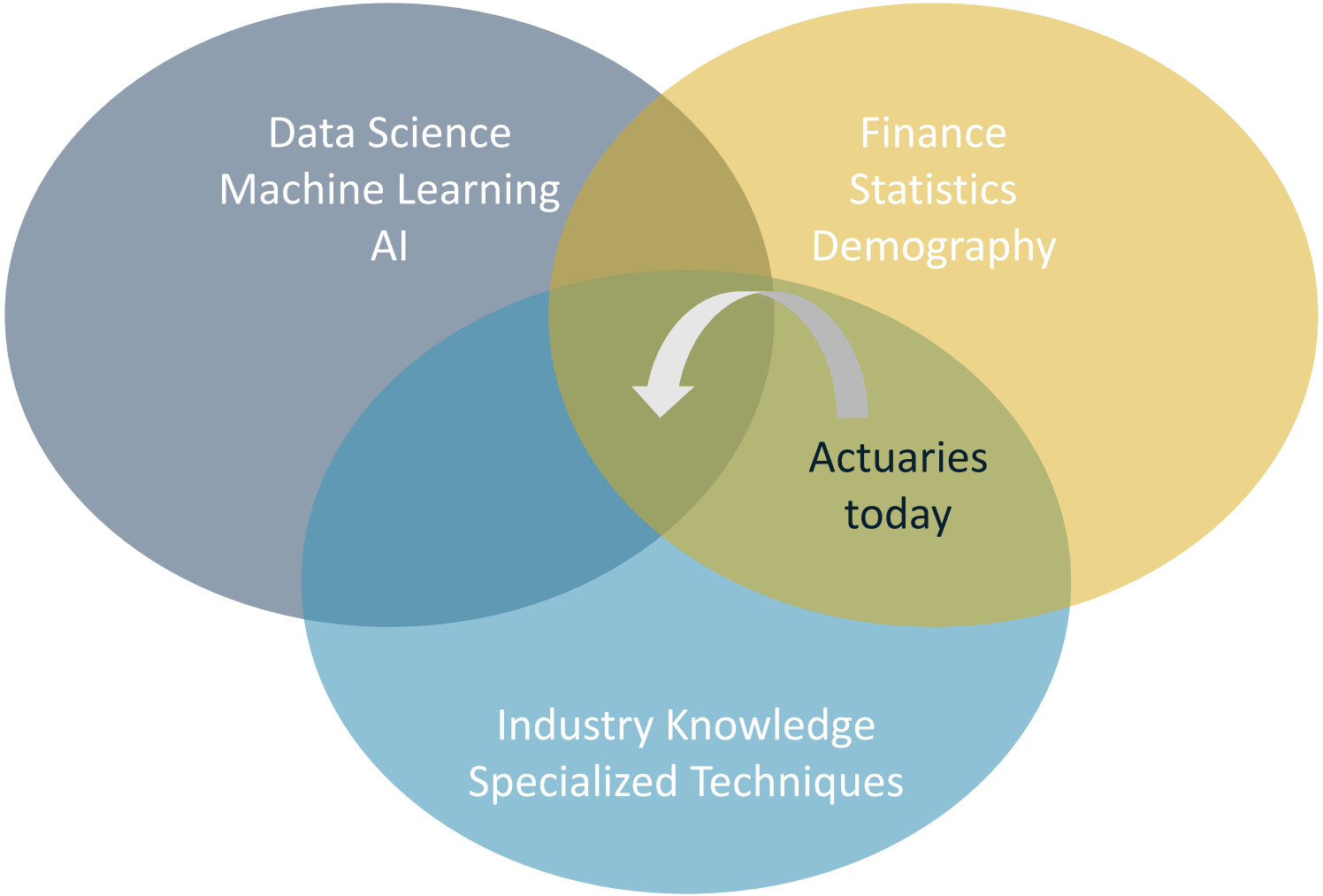
gradually decline till there be none left to *die*; as may be ſeen at one View in the Table.

From theſe Conſiderations I have formed the *adjoyned Table*, whoſe Uſes are manifold, and give a more juſt *Idea* of the *State* and *Condition* of *Mankind*, than any thing yet extant that I know of. It exhibits the *Number* of *People* in the City of *Breſlaw* of all Ages, from the *Birth* to extream *Old Age*, and thereby ſhews the Chances of *Mortality* at all *Ages*, and likewiſe how to make a certain Eſtimate of the value of *Annuities* for *Lives*, which hitherto has been only done by an imaginary *Valuation*: Alſo the *Chances* that there are that a *Perſon* of any *Age* propoſed does live to any other *Age* given; with many more, as I ſhall hereafter ſhew. This *Table* does ſhew the *number* of *Perſons* that are living in the *Age* current annexed thereto, as follows:

Age. Curt.	Perſons.	Age. Curt.	Perſons	Age. Curt.	Perſons	Age. Curt.	Perſons	Age. Curt.	Perſons	Age. Curt.	Perſons	Age.	Perſons.
1	1000	8	680	15	628	22	585	29	539	36	481	7	5547
2	855	9	670	16	622	23	579	30	531	37	472	14	4584
3	798	10	661	17	616	24	573	31	523	38	463	21	4270
4	760	11	653	18	610	25	567	32	515	39	454	28	3964
5	732	12	646	19	604	26	560	33	507	40	445	35	3604
6	710	13	640	20	598	27	553	34	499	41	436	42	3178
7	692	14	634	21	592	28	546	35	490	42	427	49	2709
												56	2194
Age Curt	Perſons.	Age. Curt.	Perſons	Age Curt.	Perſons	Age Curt.	Perſons	Age. Curt.	Perſons	Age. Curt.	Perſons	63	1694
43	417	50	346	57	272	64	202	71	141	78	58	70	1204
44	407	51	335	58	262	65	192	72	130	79	49	77	692
45	397	52	324	59	252	66	182	73	109	80	41	84	253
46	387	53	313	60	242	67	172	74	98	81	34	100	107
47	377	54	302	61	232	68	162	75	88	82	28		
48	367	55	292	62	222	69	152	76	78	83	23		34000
49	357	56	282	63	212	70	142	77	68	84	20		Sum Total.

Thus it appears, that the whole People of *Breſlaw* does conſiſt of 34000 *Souls*, being the Sum *Total* of the Perſons of all Ages in the *Table*: The firſt uſe hereof is

The (future) toolkit of actuarial science



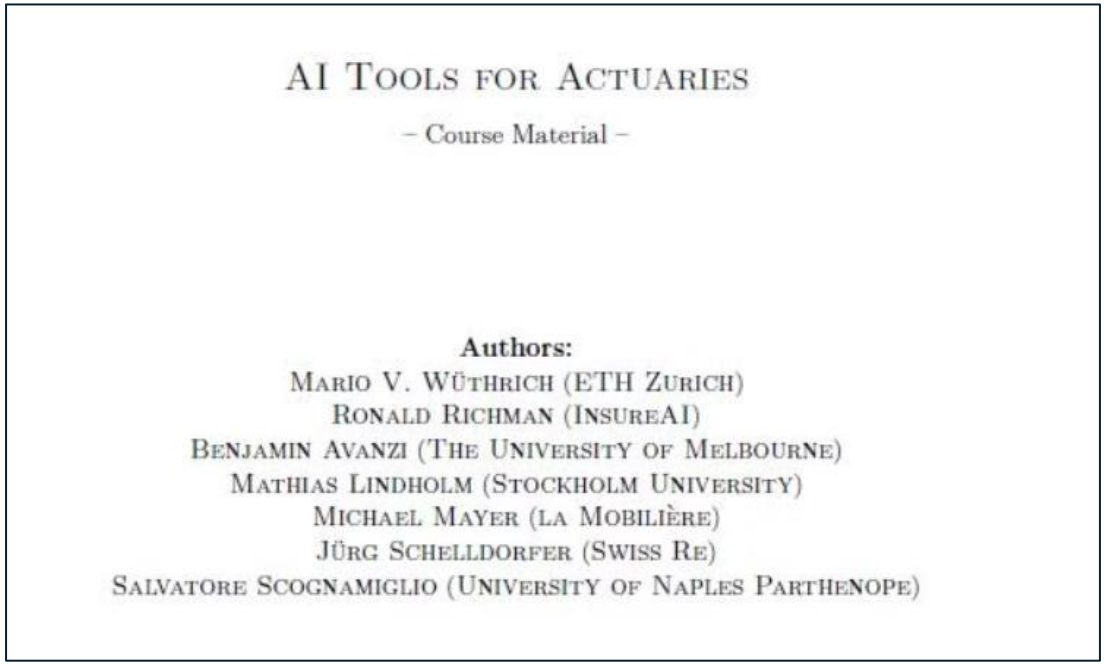
Where should we go as a profession?

- **In an AI-enabled world, we should use our deep knowledge of what works in our domains to build better models that deliver optimal outcomes to our stakeholders**
- **Instead of being intimidated by ML and AI, we should own our models and methodologies**
- We need more mathematics and deeper knowledge of code - not less
- **Remember - your actuarial training gives you 80% of what you need to know to become experts in these areas**

What should I read on the weekend?

- Read a book on AI Tools for Actuaries: https://aitools4actuaries.com/

- Run some code from the website



AI Tools for Actuaries

– Course Material –

Authors:
Mario V. Wüthrich (ETH Zurich)
Ronald Richman (InsureAI)
Benjamin Avanzi (The University of Melbourne)
Mathias Lindholm (Stockholm University)
Michael Mayer (la Mobilière)
Jürg Schelldorfer (Swiss Re)
Salvatore Scognamiglio (University of Naples Parthenope)

What was this talk all about?

- **We built bridges between actuarial science and machine learning**
  - Bridge 1: AS and ML/AI are empirical disciplines
  - Bridge 2: scaling the empirical actuarial approach might lead to artificial intelligence
  - Bridge 3: actuarial thinking can improve machine learning

ACTUARIAL SOCIETY
OF SOUTH AFRICA
Advancing Actuarial Excellence

Thank you

Embedding Layer — Categorical Data

- Categorical data abound in actuarial modelling problems!
- **One-hot encoding expresses the prior that categories are orthogonal**
  - Similar observations are not grouped together
- Traditional actuarial solution — credibility
- **Embedding layer prior — related categories should cluster together:**
  - Learns a dense vector transformation of sparse input vectors
  - Clusters similar categories together

	Actuary	Accountant	Quant	Statistician	Economist	Underwriter
Actuary	1	0	0	0	0	0
Accountant	0	1	0	0	0	0
Quant	0	0	1	0	0	0
Statistician	0	0	0	1	0	0
Economist	0	0	0	0	1	0
Underwriter	0	0	0	0	0	1

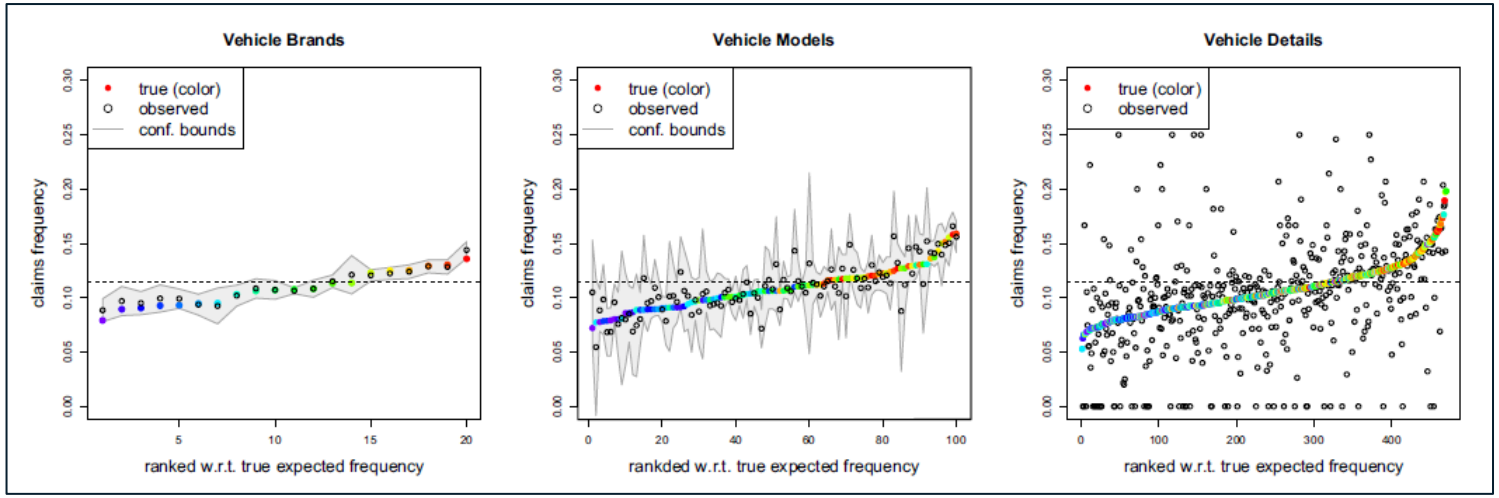
	Finance	Math	Stastistics	Liabilities
Actuary	0.5	0.25	0.5	0.5
Accountant	0.5	0	0	0
Quant	0.75	0.25	0.25	0
Statistician	0	0.5	0.85	0
Economist	0.5	0.25	0.5	0
Underwriter	0	0.1	0.05	0.75

What can go wrong with embeddings?

- We may have categorical variables where there are very few examples of each level
  - E.g.: Car brand / model / variant — lots of Toyotas but very few Abarths!

- Noisy data generated by claims process
  - May not have enough credibility to set values of factor levels / embeddings appropriately

- **Example from simulated data (colour = Vehicle Brand)**

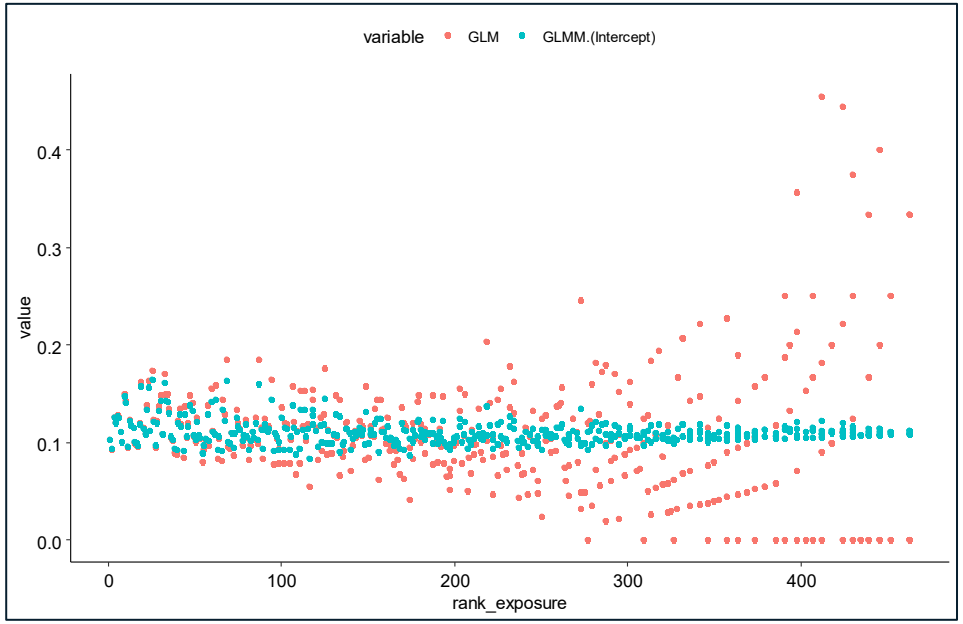


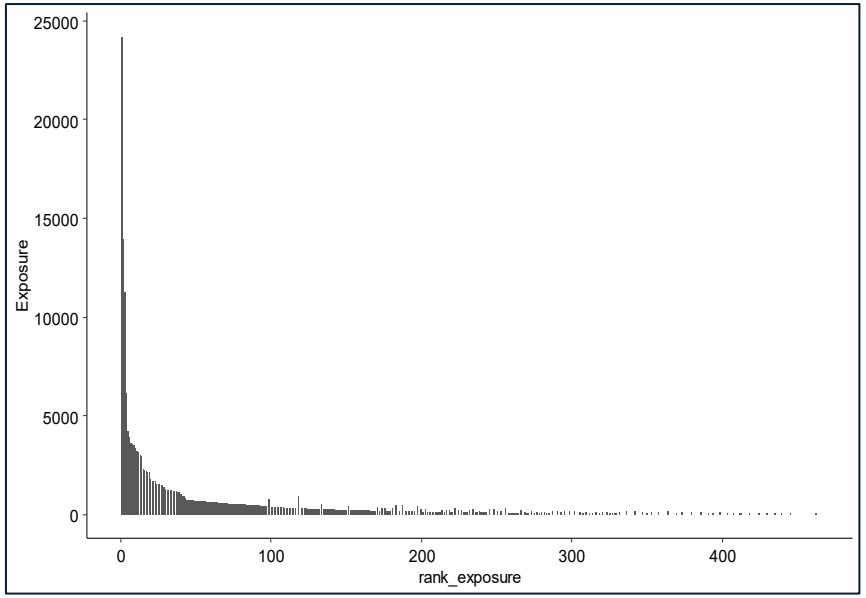
GLMMs offer a solution

- Classic actuarial solution to these problems – apply credibility theory

$$\boldsymbol{\beta_{credibility} = z * \bar{\beta} + (1 - z) * \beta_i}$$

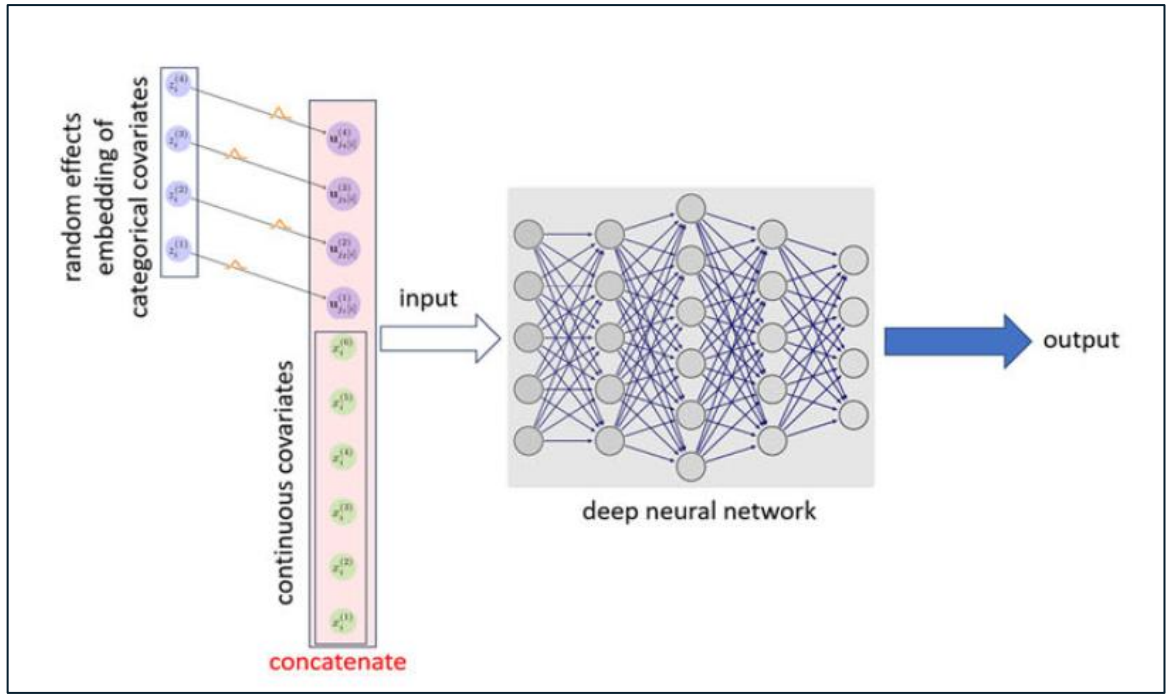
- **Generalized Linear Mixed Models (GLMMs) incorporate Bühlmann-Straub credibility automatically to pool individual coefficients towards the mean**





Putting it together

- Embeddings can be used to capture complex high-dimensional information about categorical data…
  - … with the risk that we overfit to noise, particularly when there is high cardinality
- Recently we developed Bayesian methods to credibility-weight embeddings towards zero, which depend on exposure
- **Incorporate these into a deep neural network so that both continuous and categorical variables can contribute to network predictions**
- Diagram of network used shown on right



Credibility saves the day!

- Fit basic neural network with embeddings (2D), another proposed approach to incorporating embeddings (GLMMnet) and LightGBM
  - Note that networks suffer with VehDetail!

		Average KL divergence		
		Network (2.30)	GLMMNet (4.3)	LightGBM
(0)	Null model (empirical mean)	1.0342	1.0342	1.0342
(0)	w/o categorical covariates	0.3947	0.3970	0.3958
(1)	With `VehBrand`	0.2622	0.3076	0.2763
(1)	With `VehModel`	0.2188	0.2961	0.2499
(1)	With `VehDetail`	0.2615	0.3292	0.2240
(2)	With `VehBrand`, `VehModel`	0.2312	0.2958	0.2618
(3)	With `VehBrand`, `VehModel`, `VehDetail`	0.2694	0.3305	0.2191

- **Adding regularisation to the network produces excellent results:** VI slightly better than MAP

	Average KL divergence	
	Network (2.30)	GLMMNet (4.3)
No regularization: with `VehBrand`, `VehModel`	0.2312	0.2958
MAP regularization: with `VehBrand`, `VehModel`	0.2212	0.2799
Ad-hoc regularization: with `VehBrand`, `VehModel`	0.2260	0.2794
VB regularization: with `VehBrand`, `VehModel`	0.2331	0.2800
No regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.2694	0.3305
MAP regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.1446	0.2584
Ad-hoc regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.1449	0.2596
VB regularization: with `VehBrand`, `VehModel`, `VehDetail`	0.1410	0.2589