1972 - 2022

Society of Actuaries in Ireland

Machine Learning in General Insurance Reserving Method Comparison and Interpretation

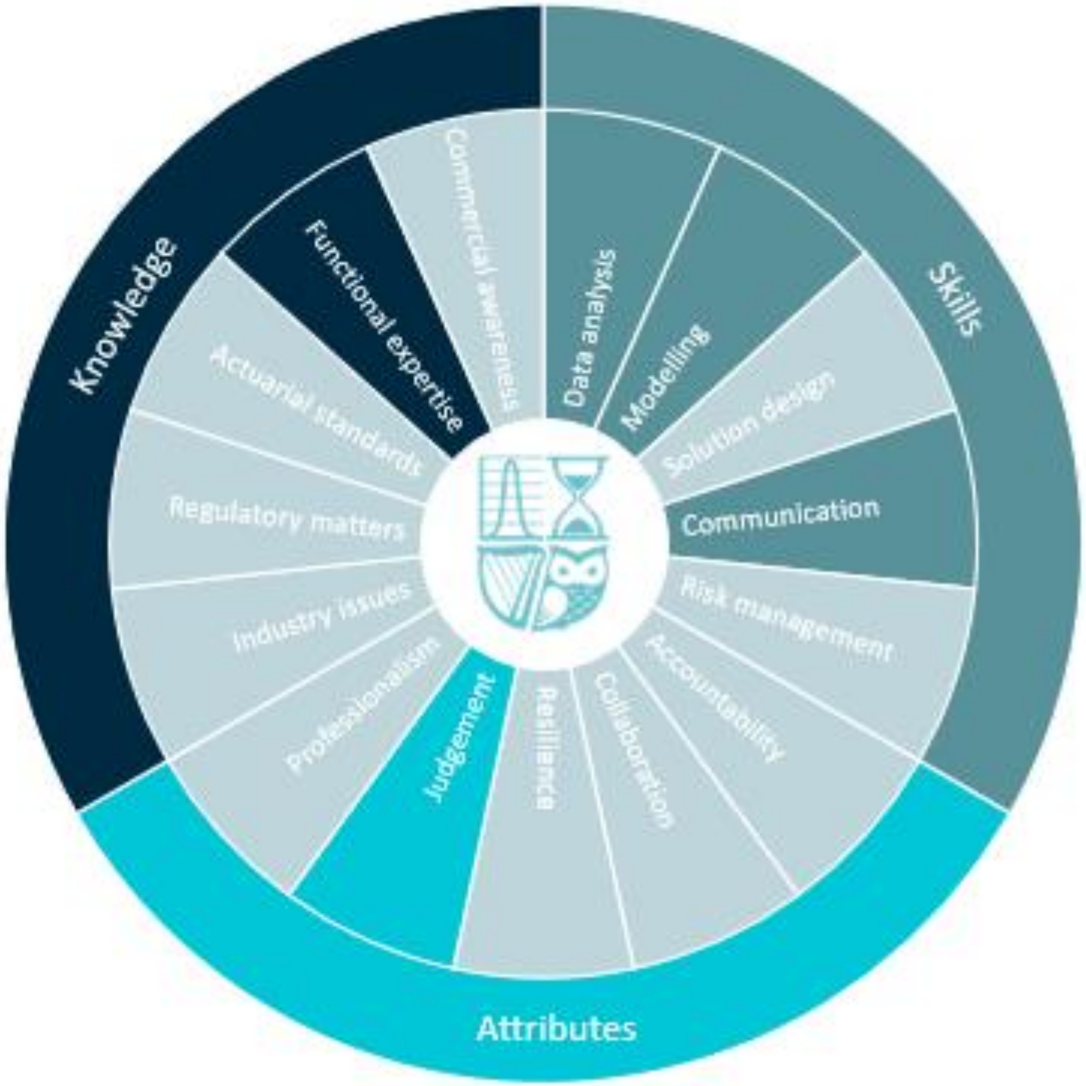
April Lu (she/her); John McCarthy (he/him)

14th June 2022

Disclaimer

The views expressed in this presentation are those of the presenter(s) and not necessarily those of the Society of Actuaries in Ireland or their employers.

Competency Framework Wheel



Agenda

This presentation builds on previous work presented at the 2021 IFoA Spring Conference* and is aimed at those relatively new to machine learning

- Reminder of machine learning framework for modelling triangle data
- Data
- Results
- Diagnostic charts
- Next steps
- Q&A

*https://institute-and-faculty-of-actuaries.github.io/mlr-blog/post/f-mlr3example/

Machine Learning in Reserving Working Party

- Who are we?
  - International group of actuaries, data scientists and academics from diverse backgrounds, chaired by Sarah MacDonnell
- What are our aims?
  - Learn how machine learning (ML) can be used in non-life reserving
  - Carry out research on the use of ML in reserving
- Our workstreams
  - Foundations
  - Literature Review
  - Survey
  - Data
  - Research

Find us at https://institute-and-faculty-of-actuaries.github.io/mlr-blog/

Framework

Incremental loss triangle

Available data

To be predicted

Development period

Accident period

	1	2	3	4	5
1	1,054,995	717,048	885,139	526,803	764,239
2	1,065,209	1,210,129	849,025	658,627	
3	1,077,450	1,041,976	866,843		
4	1,210,198	886,174			
5	985,520				

Incremental loss data table

Training data

Test data

Acc	Dev	Incremental loss
1	1	1,054,995
1	2	717,048
1	3	885,139
1	4	526,803
1	5	764,239
2	1	1,065,209
2	2	1,210,129
2	3	849,025
2	4	658,627
2	5	
3	1	1,077,450
3	2	1,041,976
3	3	866,843
3	4	
3	5	
4	1	1,210,198
4	2	886,174
4	3	
4	4	
4	5	
5	1	985,520
5	2	
5	3	
5	4	
5	5	

Framework

X = “Features” or “Predictors” or “Inputs” or “Independent variables”

$$Y \approx f(X)$$

Accident period	Development period	Incremental loss
1	1	1,054,995
1	2	717,048
1	3	885,139
1	4	526,803
1	5	764,239
2	1	1,065,209
2	2	1,210,129
2	3	849,025
2	4	658,627
2	5	
3	1	1,077,450

$$\left(Y - f(X)\right)^2$$

Y = “Target” or “Output” or “Response” or “Dependent variable”

Features

X

Y	1.30	3.00	8.30	6.30	11.00

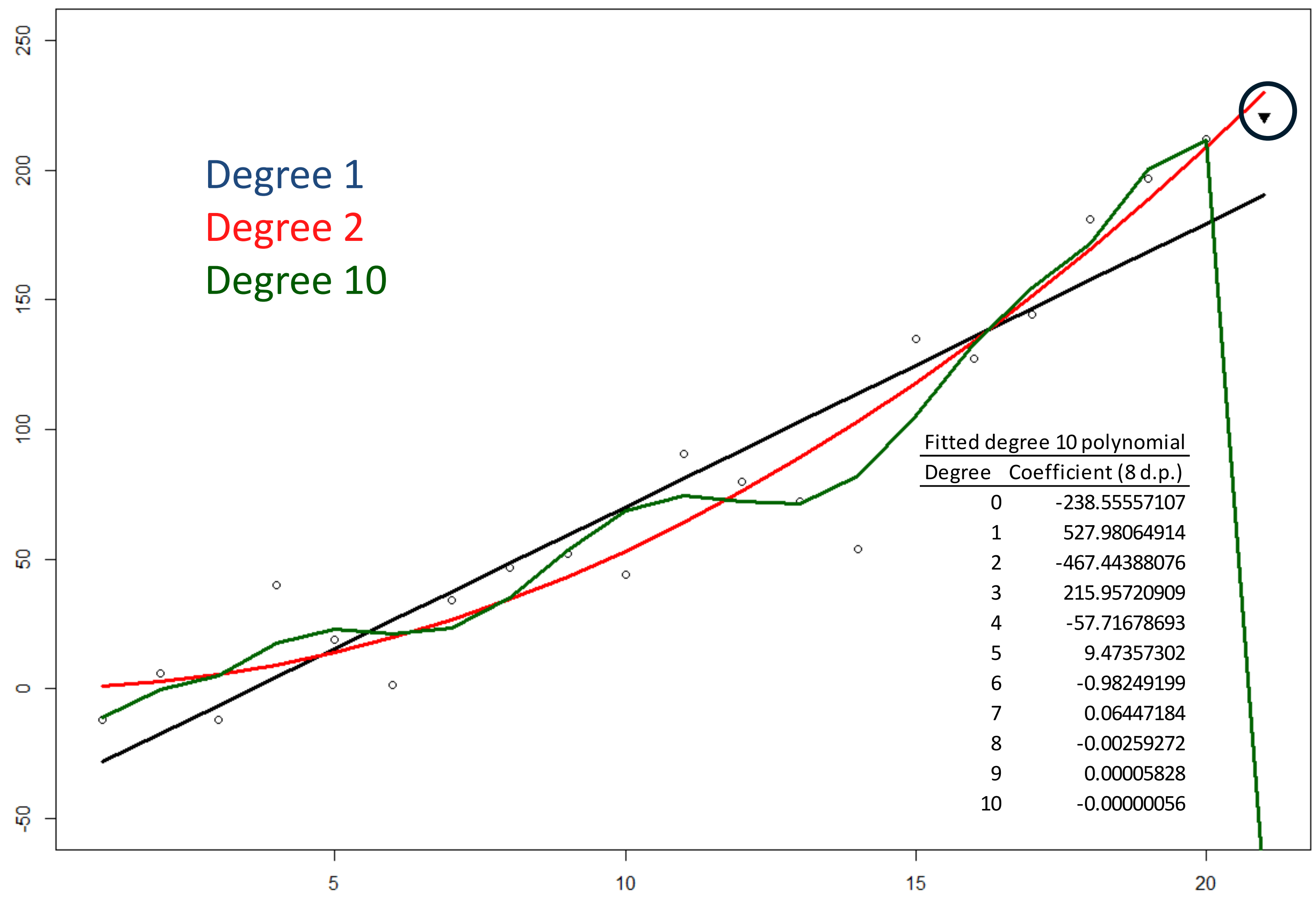
Angle_h	45	90	255	195	330
Angle_m	180	0	180	180	0

E.g. calendar period

*Based on an example from the book “Deep Learning with R” by Francois Chollet with J.J. Allaire

Hyperparameters and tuning

Polynomials of degree 1, 2 and 10 fitted to 20 x-y pairs of a quadratic (plus noise) and used to predict value at x = 21



Degree	Coefficient (8 d.p.)
0	-238.55557107
1	527.98064914
2	-467.44388076
3	215.95720909
4	-57.71678693
5	9.47357302
6	-0.98249199
7	0.06447184
8	-0.00259272
9	0.00005828
10	-0.00000056

- Example – quadratic plus random noise
- Fit a polynomial using first 20 points (training data)
- Predict the value at x = 21 (test data)
- Degree of polynomial is a hyperparameter

Cross validation

1972 - 2022

- Withhold some training data from fitting process and use this data to estimate performance out-of-sample for candidate hyperparameter

Acc	Dev	Incremental loss	**Cross validation fold**
1	1	1,054,995	**2**
1	2	717,048	**2**
1	3	885,139	**1**
~~1~~	~~4~~	~~526,803~~	~~**3**~~
1	5	764,239	**2**
2	1	1,065,209	**1**
2	2	1,210,129	**1**
~~2~~	~~3~~	~~849,025~~	~~**3**~~
~~2~~	~~4~~	~~658,627~~	~~**3**~~
2	5		**N/A**
~~3~~	~~1~~	~~1,077,450~~	~~**3**~~
~~3~~	~~2~~	~~1,041,976~~	~~**3**~~
3	3	866,843	**2**
3	4		**N/A**
3	5		**N/A**
4	1	1,210,198	**1**
4	2	886,174	**2**
4	3		**N/A**
4	4		**N/A**
4	5		**N/A**
5	1	985,520	**1**

Training data - folds 1 and 2

Fold 3 - estimate test error for candidate hyperparameter

Initialise a candidate hyperparameter value
Train the model on folds 1 and 2
Estimate the test error using fold 3

Acc	Dev	Incremental loss	**Cross validation fold**
1	1	1,054,995	**2**
1	2	717,048	**2**
1	3	885,139	**1**
1	4	526,803	**3**
1	5	764,239	**2**
2	1	1,065,209	**1**
2	2	1,210,129	**1**
2	3	849,025	**3**
2	4	658,627	**3**
[redacted]			
3	1	1,077,450	**3**
3	2	1,041,976	**3**
3	3	866,843	**2**

LASSO

Select λ $\longrightarrow$ Fit model (fits a β_i for each feature x_i)

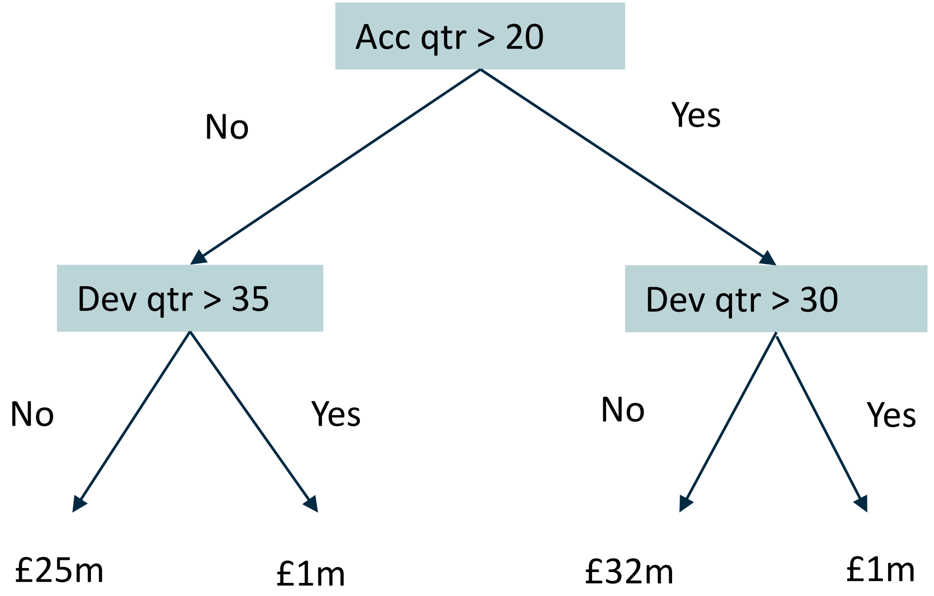
$$e^{\beta_0+\beta_1 x_1+\cdots\beta_p x_p}$$

Minimise the expression below:

$$-\sum_{m=1}^{n} l(y_m;\hat{\beta}) + \lambda \sum_{r=1}^{p} |\hat{\beta}_r|$$

Hyperparameter

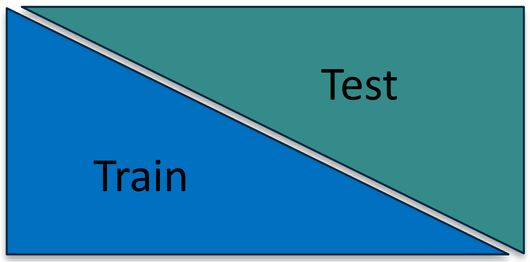
XGBoost



- Individual decision tree model typically performs poorly
- XGBoost outputs a collection of decision trees – combined prediction much better
- Several hyperparameters control how the collection of decision trees is constructed – number of trees to use, rate of adjustment from one tree to the next, tree depth and many more
- Outstanding track record in data science prediction competitions
- Not easy to grasp the details behind fitting procedure

Data

- Exposure level
- Frequency
- Severity
- Inflation
- Occurrence delay
- Notification delay
- Settlement delay



claim_no	pmt_no	occurrence_period	occurrence_time	claim_size	notidel	setldel	payment_time	payment_period	payment_size	payment_inflated	payment_delay
1	1	1	0.10	570,164	4.16	3.37	4.63	5	11,186	11,575	0.37
1	2	1	0.10	570,164	4.16	3.37	5.16	6	11,702	12,156	0.53
1	3	1	0.10	570,164	4.16	3.37	5.63	6	8,971	9,352	0.48
1	4	1	0.10	570,164	4.16	3.37	6.28	7	10,806	11,319	0.64
1	5	1	0.10	570,164	4.16	3.37	6.51	7	9,451	9,917	0.24
1	6	1	0.10	570,164	4.16	3.37	6.87	7	13,237	13,926	0.35
1	7	1	0.10	570,164	4.16	3.37	7.48	8	428,551	452,907	0.62
1	8	1	0.10	570,164	4.16	3.37	7.63	8	76,260	80,683	0.15
2	1	1	0.93	153,137	0.10	99.01	17.43	18	5,657	6,435	16.40
2	2	1	0.93	153,137	0.10	99.01	33.20	34	5,184	6,625	15.77
2	3	1	0.93	153,137	0.10	99.01	48.45	49	4,786	6,847	15.26
2	4	1	0.93	153,137	0.10	99.01	64.20	65	5,468	8,787	15.75
2	5	1	0.93	153,137	0.10	99.01	78.01	79	5,085	9,049	13.81
2	6	1	0.93	153,137	0.10	99.01	91.41	92	119,927	216,602	13.40
2	7	1	0.93	153,137	0.10	99.01	100.04	101	7,030	12,696	8.63

Data

The SynthETIC R package* implements a simulation machine for claims data using the methodology described by Avanzi et al, 2020.

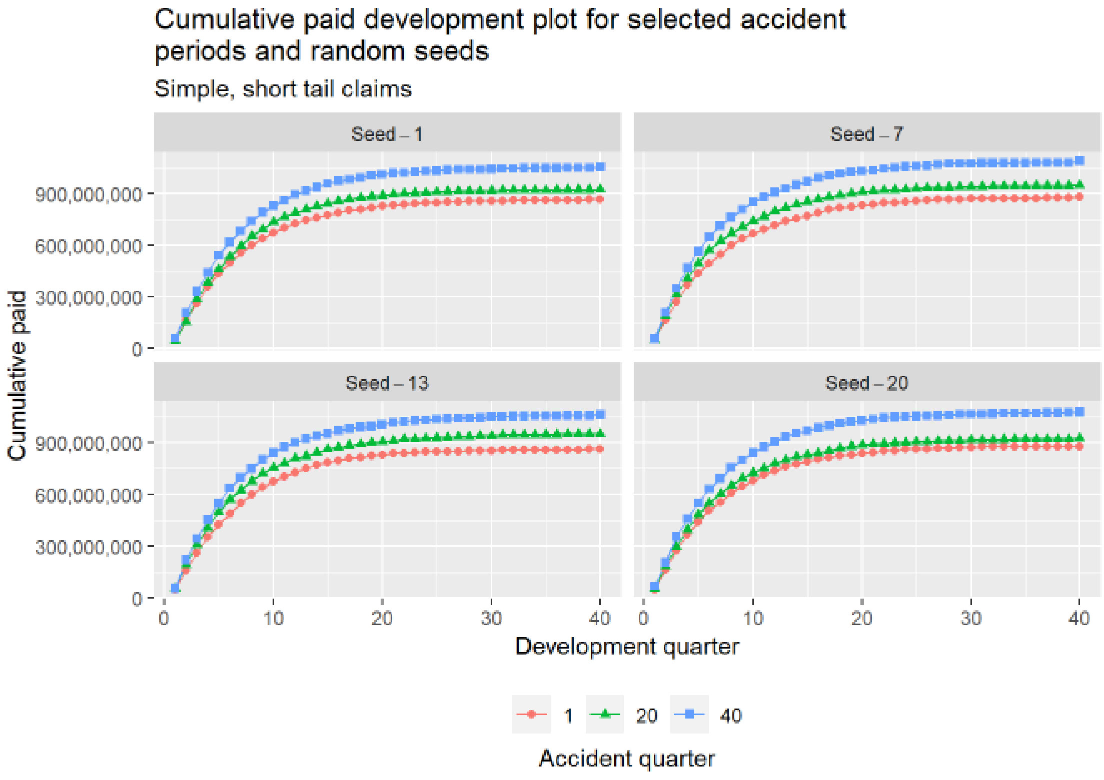
Four interesting environments are already in the public domain**

We simulated twenty triangles for each environment

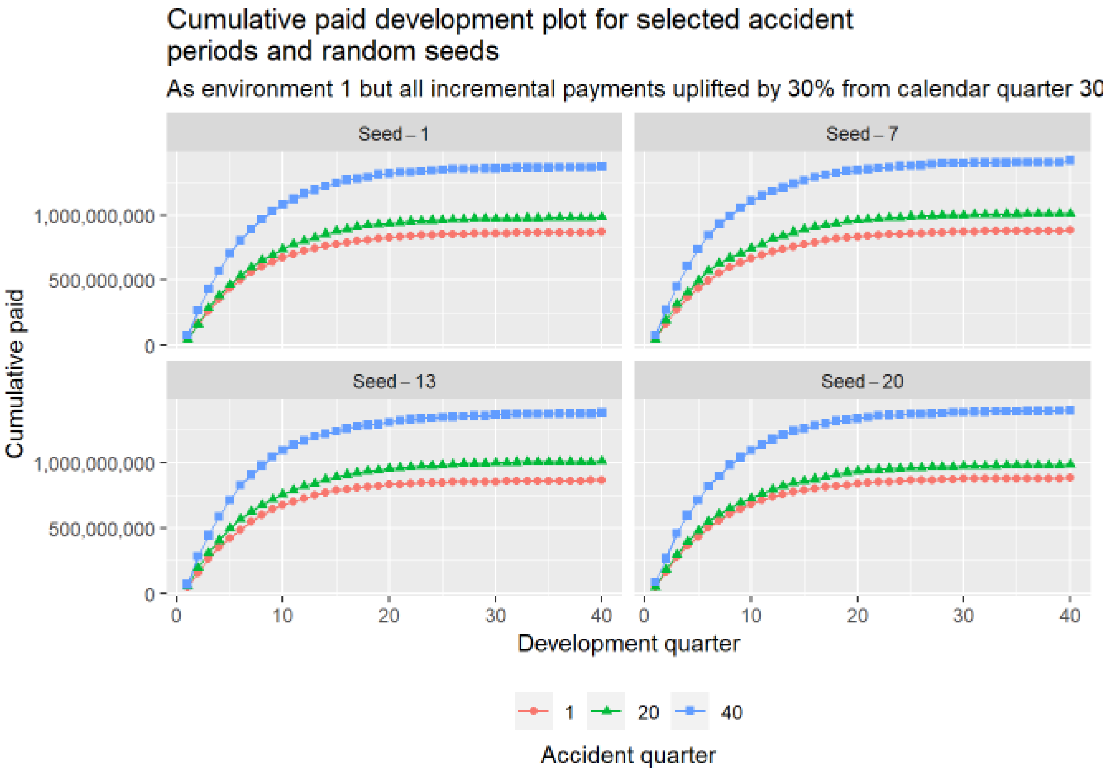
*https://institute-and-faculty-of-actuaries.github.io/mlr-blog/post/synthetic/ for background **https://github.com/agi-lab/reserving-MDN-ResMDN

Data

Environment 1

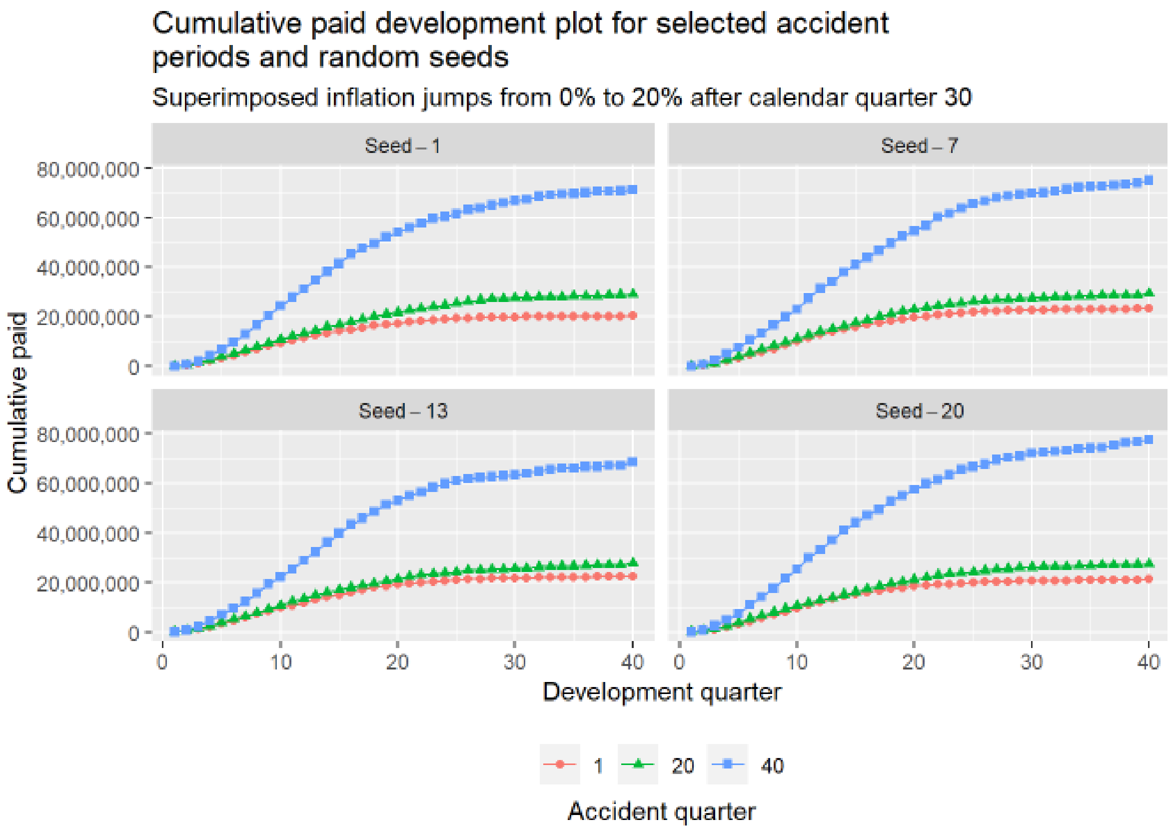


Environment 2

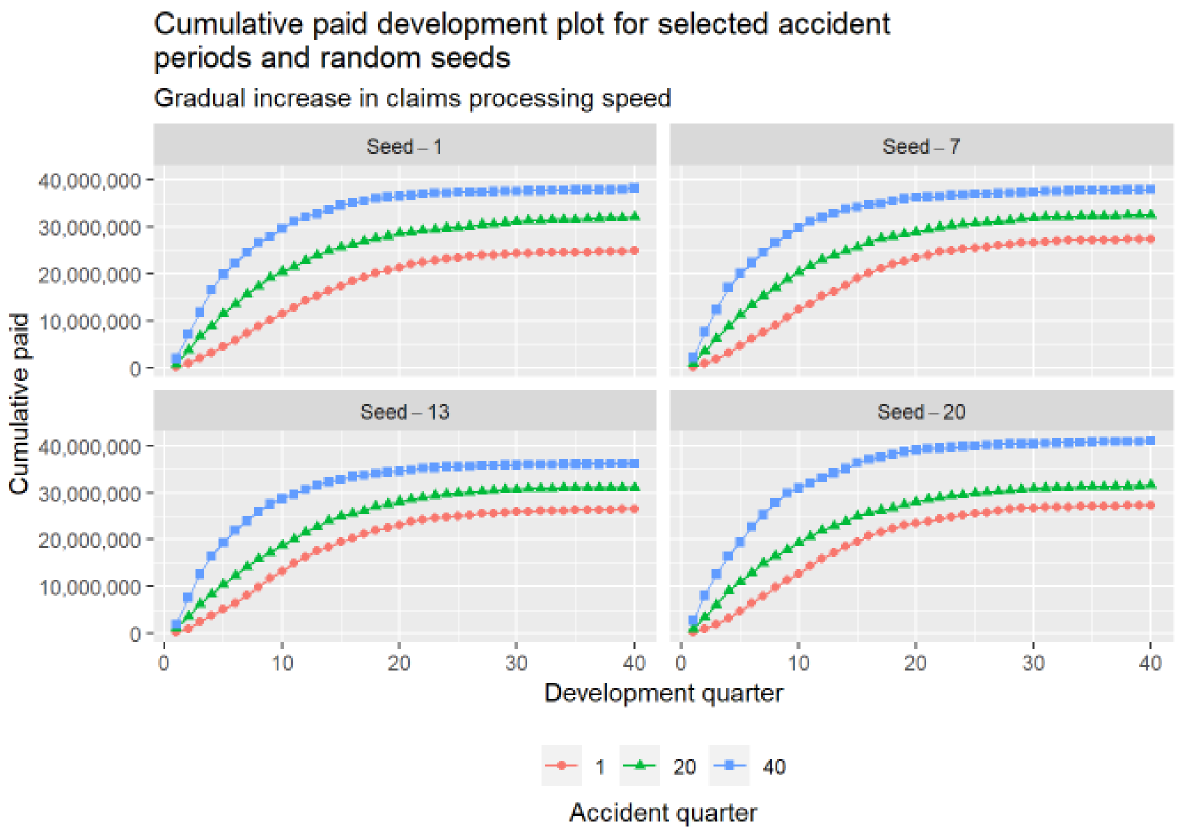


Data

Environment 3



Environment 4



Data

Environment 5

Summary of modelling approach

20 simulations of 40 x 40 triangle of accident x development quarter.

Training data is calendar quarter <= 40, test data is calendar quarter>40

Chain ladder (volume all), LASSO and XGBoost fit using accident and development quarter factors as features ("_Basic" models)

5

5-fold random cross validation

LASSO lambda tuned per blog post* and XGBoost n_rounds tuned

Additional features engineered based on LASSO blog post* to capture interactions and calendar/accident/development period trends. LASSO and XGBoost fitted to this data (" Extra" models)

*https://institute-and-faculty-of-actuaries.github.io/mlr-blog/post/f-lasso/

Caveat

- The examples here are intended to be instructional rather than conclusive
- We make no claims about the superiority/inferiority of any individual machine learning method for reserving in general.
- Real world data will introduce more problems
- Better performance in our examples may be possible with more time to tune the hyperparameters/different cross validation approach/different loss function

Results

Average reserve error [(predicted future paid / actual future paid) – 1] across all 20 random seeds

Environment	Description	Chain ladder	LASSO_Basic	LASSO_Extra	XGBoost_Basic	XGBoost_Extra
1	Simple, short tail	1%	13%	0%	2%	-3%
2	30% uplift to incremental paid from cal qtr 30 onwards	9%	21%	1%	6%	0%
3	Superimposed inflation jumps to 20% after cal qtr 30	-33%	-39%	-3%	-54%	-25%
4	Gradual increase in claims processing speed	95%	111%	2%	65%	9%
5	Longer tail, more volatile claims development	53%	3%	23%	-21%	-25%

Results

- Shiny app walkthrough

Conclusion

- In simulated data, ML methods were able to reproduce CL results on simple development data and pick up on calendar / accident period trends that cause CL problems
- Reviewing a range of diagnostics is useful for interpreting machine learning (any) models
- Lots more work to do!

Further work on triangles

- Rolling origin cross validation
- Loss function – claims development result
- Real-world data
- Further model interpretation and diagnostics

Q&A

Please use the **Q&A function** to ask a question