Red Centre Software
Data Comes Alive

Validating Black Box Neural Nets

Dale Chant, Michael Potter, Red Centre Software Pty Ltd.

Version: 30 January 2019

Confidential.

Validating Black Box Neural Nets

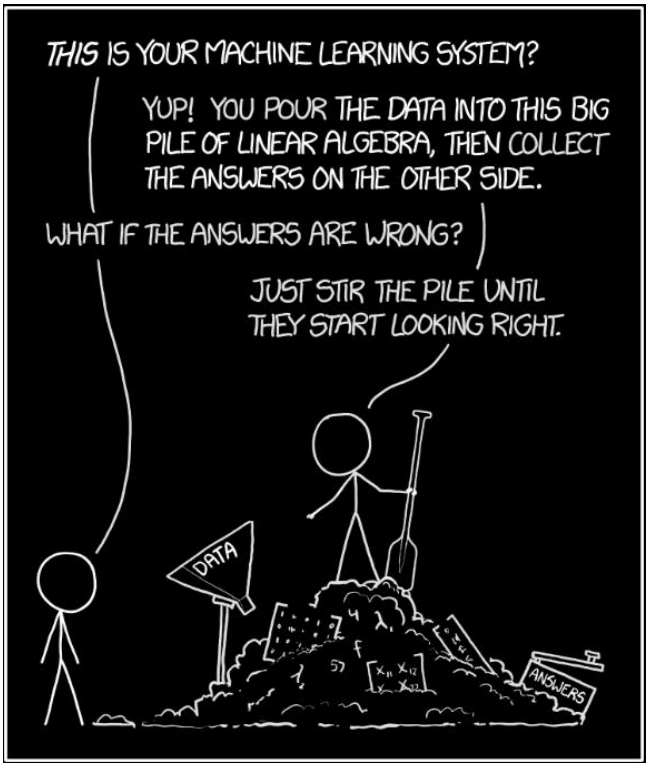
This document expands on the presentation given at the Association for Survey Computing Machine Learning conference, 15Nov2018, at London, Ort House.

Abstract

Machine learning, driven by recent advances in neural net technology, holds much promise, but how to validate any particular model? This paper looks at why validation is necessary, and describes some practical techniques for doing so. The example scenario (from a Health Insurance survey, N=29,145) is a sentiment analysis of verbatims arising from the Net Promoter Score (NPS) expatiate question "Why that rating?". We show that respondents' ratings can be predicted from the sentiment to a useful level of accuracy, and that the model can be validated by reference to external benchmarks.

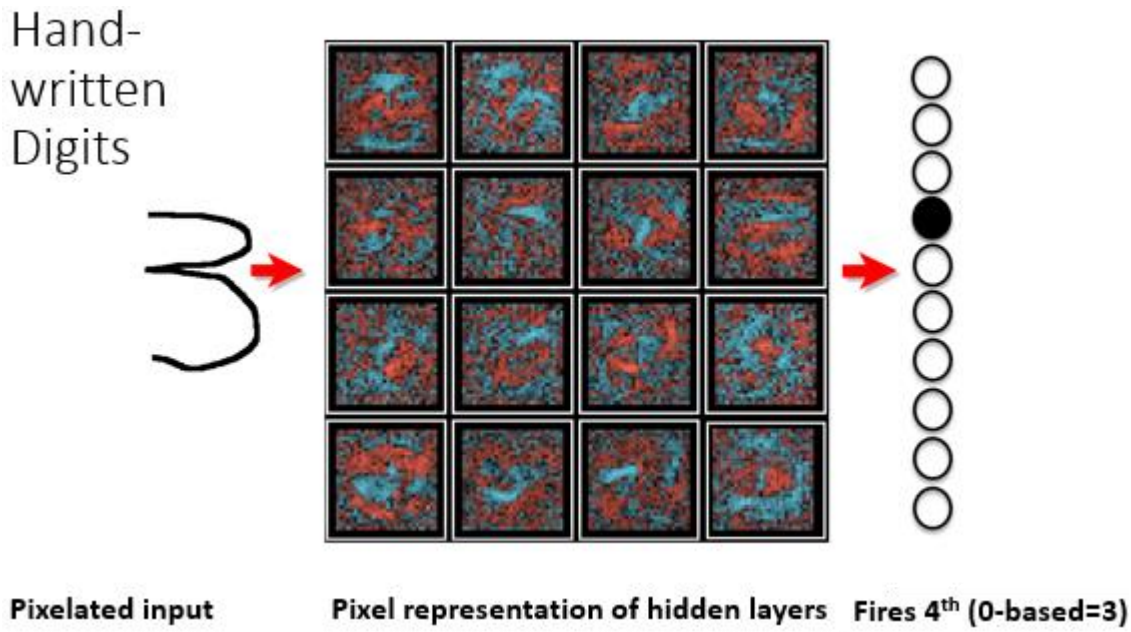
Why Need to Validate?

Neural nets are practically opaque to a debug walk-through. Reruns against the same training data can yield quite different results because network layers are initialised by random weights and biases, and thus there is no guarantee that the cost function has not found misleading local minima.



https://xkcd.com/1838/

Sanderson[1] shows that visualisations of the inner layers of a hand-written digit recognition model bear no evolving relationships to anything a human would recognise. His simple model operates in a 13,002 dimensional space (input pixels*layer weights and biases), well beyond our ability to fathom.



A neural net for distinguishing dogs from cats may in fact be identifying collars - to be discovered empirically by varying the inputs to determine why the model failed on unseen cases.

One simply cannot explain why or how a particular prediction was correct or incorrect.

Approaches to Validation

Therefore, indirect methods are required. First, ensure the model is internally consistent, and then compare to independent benchmarks.

To determine consistency, test the model on its own training data. Having already seen the answers, how well does it remember? Next, compare multiple runs, where disparities measure instability. Then check that the ratio of correct:total degrades naturally as resolution increases. One would expect Binary sentiment (predict just positive or negative) to be more accurate than predicting a value from 0 to 10.

Since a model's outputs cannot be practically traced from the inputs, the next recourse is benchmarking against procedures or algorithms which can be formally verified.

For hand-written digits, the output should be at least better than a random guess (one in ten chance of being correct). Measuring darkness, where a *1* has fewer pixels than an *8*, and similar approaches, can give up to 50% correct[2].

For Brand Coding, compare the correct rate to that obtained from human coders or approximate string-matching algorithms such as Damerau-Levenshtein[3,4].

For text sentiment, compare the correct rating to that obtained from the R Syuzhet package[5], which sums pre-defined word weights.

For all, run correlations of trained versus predicted on the same data set, looking for a score of at least 0.9.

Predicting Likely to Recommend (NPS) from Sentiment

Consider the standard Net Promoter Score (NPS) questions:

- Q4a: On a scale of 0 to 10, how likely are you to recommend <brand> to friend or relative? Answer: *8*
- Q4b: Why did you give that rating? Answer: *Good service, reasonable price*

Rating + expatiate questions provide ready-to-go self-tagged training sets. This respondent equates the sentiment of *Good service, reasonable price* to be *8*. Given just the text, how well can the sentiment score of 8 be predicted?

There are generic verbatim issues which inevitably complicate matters: truncation of long responses by the field service, slang and abbreviations, poor formatting and missing spaces, general nonsense, obscenities, neologisms, skimmers (random or repeated characters), and wildly unorthodox spellings.

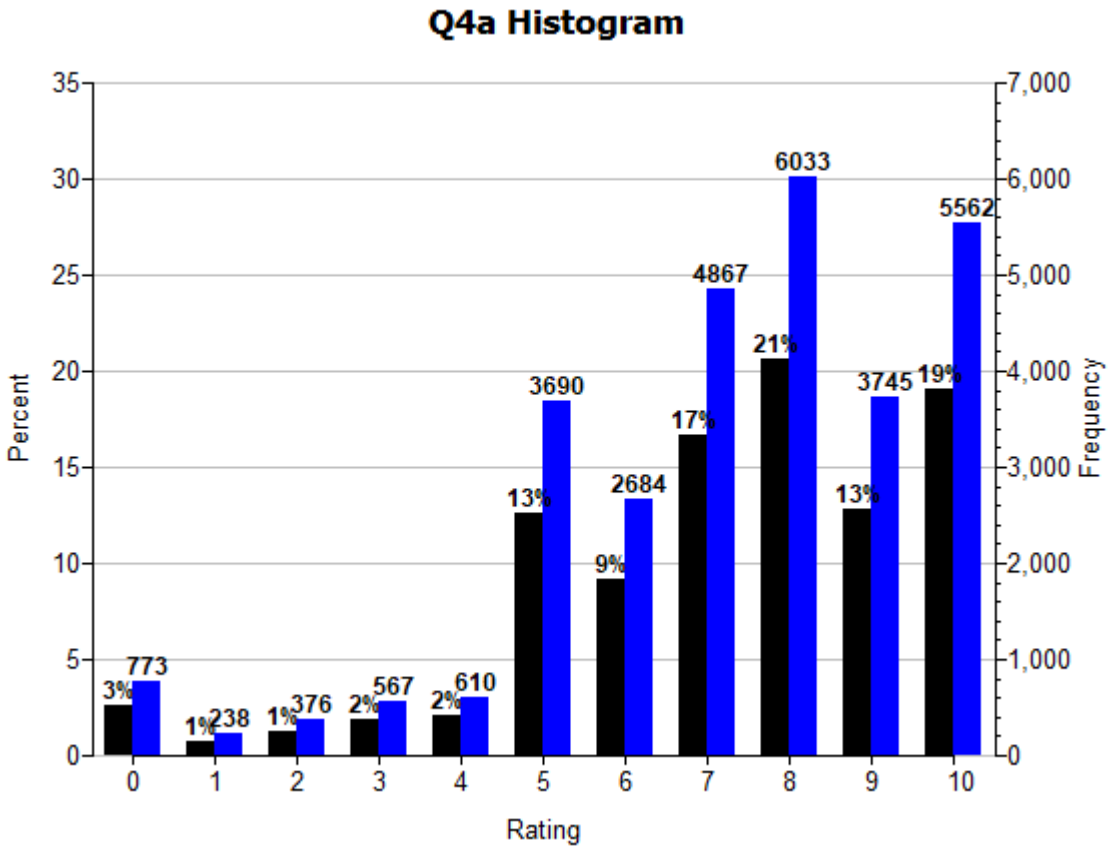
- Truncated by field: *Low annual premiums; option of no excess; ease of using hospital cover, i.e. bills sent directly* ***fro***
- Slang and abbreviations: *I* ***wanna*** *leave and change* ***asap***
- Formatting: *Happywiththeirsystem*
- Nonsense: *Daddy is good at it*
- Puzzling: *confiability*
- Skimmers: *434774244347742443477*
- Spelling: *physcologist*
- Obscene: [expletives deleted]

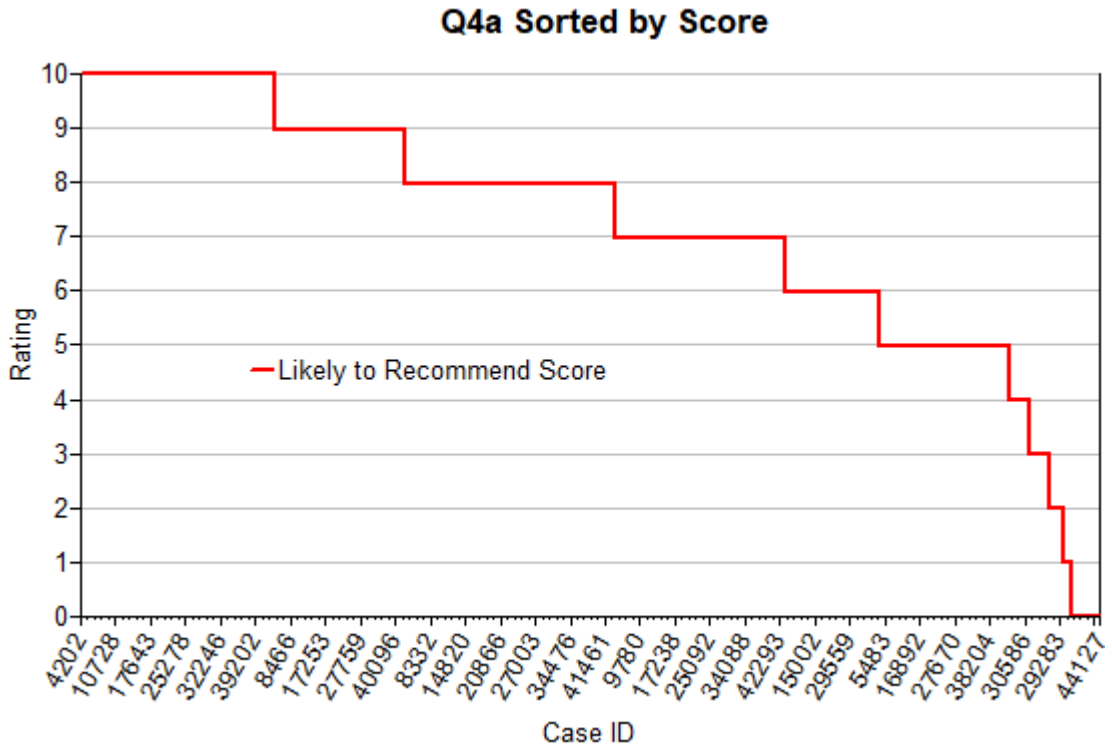
Such cases should be removed from consideration, since noise will degrade both benchmark and validation.

The Q4a frequency distribution is

	Q4a Likely to Recommend										
Rating	0	1	2	3	4	5	6	7	8	9	10
Frequency	773	238	376	567	610	3690	2684	4867	6033	3745	5562

Two visualisations:





Over 90% of the 29,145 respondents gave a rating >= 5. By manual verbatim review, the negative/positive apex is between 6 and 7, and disengaged respondents cluster at ratings 0 to 4.

Independent Benchmark: Syuzhet Scores

syuzhet is an R package for text analysis. Sentiment is evaluated by reference to a dictionary of weighted words. For example, if love=0.75, beauty=0.5 and good=0.5 then "Love of beauty is good" scores 0.75+0.5+0.5 = 1.75

The most negative response scores -4:

> *…lack of details concerning claimable areas…non-claimable costs…feels cheap and nasty to say a health service is claimable, only to discover it's not…disappointing and covert…*

Most positive response scores 8.4:

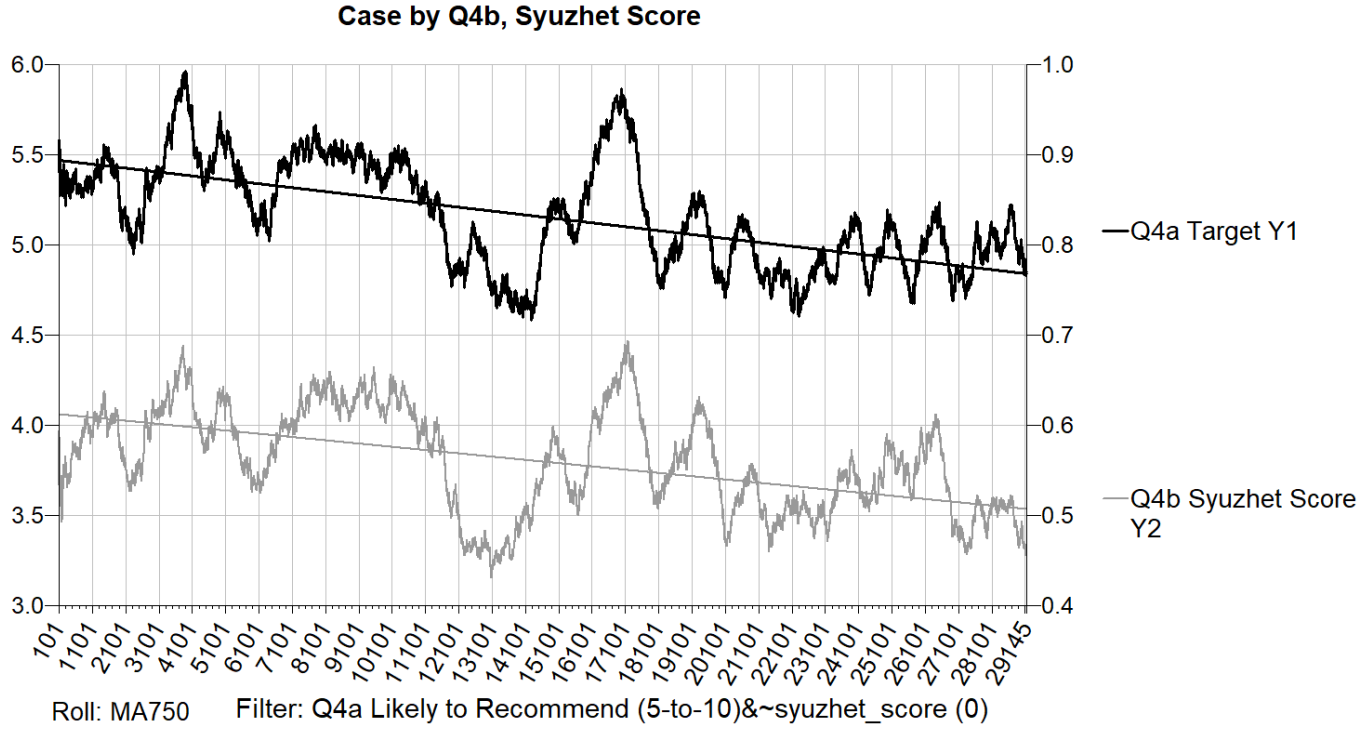
> *… The yearly cost was half … the benefits were immensely better…the best value available today… not for profit fund… accessible for helpful information...*

Noise cases are classified as Syuzhet score = 0 or Q4a less than 5. If there is not enough sentiment for a non-zero Syuzhet score then machine learning will also have difficulties. Ratings 0 to 4 comprise only a 10% subset with many low engagement verbatims.

Syuzhet is limited by:

- many zeroes because a word misspelled so weight lookup fails
- words not in dictionary
- opposites can cancel, and
- long responses can accumulate greater absolute score.

The Syuzhet benchmark, de-noised by case filter and moving average, is however remarkably good:



The X axis is one case per point. Black is the target, being the respondents' own ratings. Grey is the Syuzhet score on verbatims.

Q4a is unusually dynamic for an NPS rating. Over the several years of data collection some not-for-profit health insurers were bought out by private interests, leading to declines in sentiment, and the field is highly competitive.

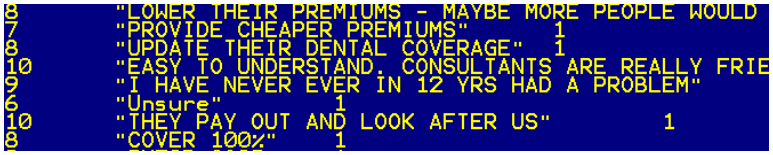
Can Machine Learning Improve on Syuzhet?

We used the new Microsoft ML.Net[6] suite. This is free and easy to deploy on Windows machines, with programming in Visual Basic. Adapting the supplied examples was not difficult.

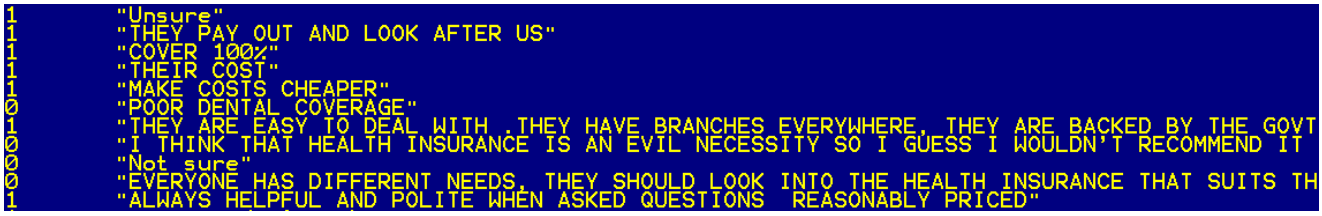
For the model Inputs, we assembled text files at each target resolution formatted as

<target rating>tab<verbatim>

For the 0 to 10 (full) scale:

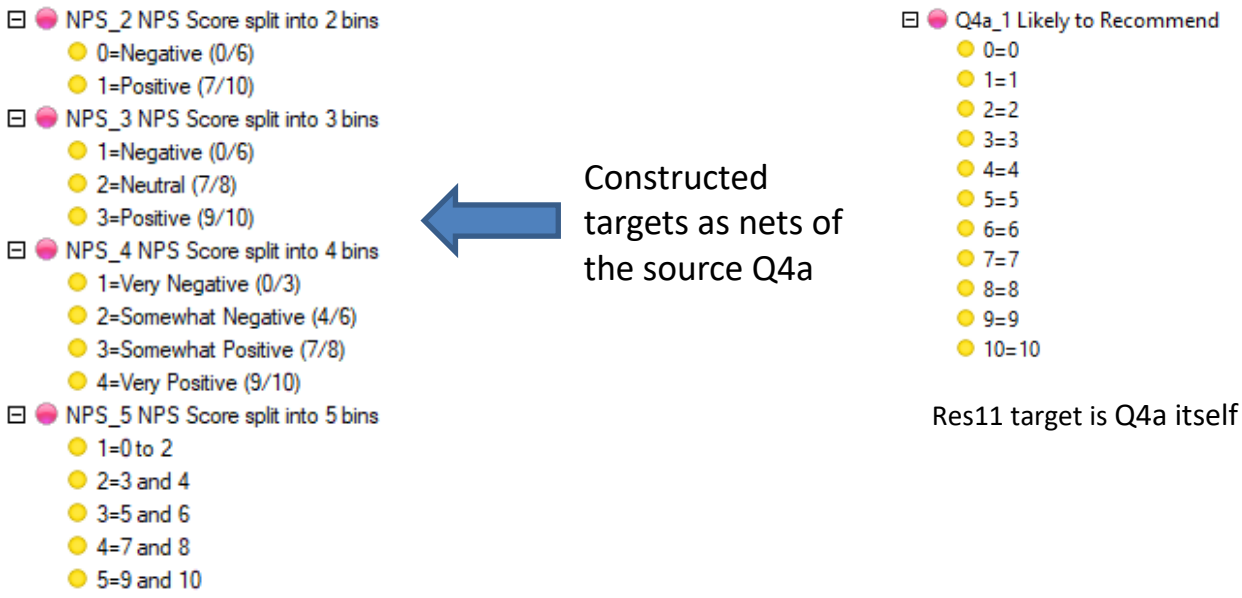


For the 0 to 1 scale (binary) as net of 0 to 6 = 0 = negative, and 7 to 10 = 1 = positive:



The targets are defined as nets of the respondent Q4a ratings.

Target code	Res2	Res3 (NPS)	Res4	Res5	Res11
0	Q4a(1/6)				
1	Q4a(7/10)	Q4a(1/6)	Q4a(0/3)	Q4a(0/2)	Q4a(1)
2		Q4a(7/8)	Q4a(4/6)	Q4a(3/4)	Q4a(2)
3		Q4a(9/10)	Q4a(7/8)	Q4a(5/6)	Q4a(3)
4			Q4a(9/10)	Q4a(7/8)	Q4a(4)
5				Q4a(9/10)	Q4a(5)
…					…
10					Q4a(10)



The model was executed five times at each resolution.

The key steps are:

```
For runID = 1 To 5
    model = TrainSentimentMultiClass("Q4a_1", InputTextVar)
    EvaluateSentimentMultiClass(model, InputTextVar, runID)
Next

Public Function TrainSentimentMultiClass(…)
    …
    Dim sdcac As New Trainers.StochasticDualCoordinateAscentClassifier
    …
    model = pipeline.Train(Of ClassificationData, ClassPredictionMultiClass)
     …
    Return model
End Function

Public Function EvaluateSentimentMultiClass(model…)
    …
    pred = model.Predict(verbatim)
    …
End Function
```

The black box is *StochasticDualCoordinateAscentClassifier*[7]. The ML.Net suite has about twenty algorithms. This was chosen empirically on the basis that it gave the best results.

The performance of each run at each resolution is quantified as the ratio of correct:total, calculated as

$$100*(\Sigma \text{ corrects})/(\Sigma \text{ cases})$$

For example, at Resolution 2, Run 1

96.30% correct		**Predicted Resolution 2 Run**	
		Negative	Positive
Q4a Likely to Recommend	Cases	9,060	20,085
	Net 0 to 6	**8,460**	478
		93%	2%

	Net 7 to 10	600	**19,607**
		7%	98%

The ratio of correct predictions is 96.30%, calculated as 100*(8460+19607)/(9060+20085).

At the lowest resolution (binary) the prediction cross tabulations for runs 1 to 5 are

Correctly predicted : 96.30%

Column Percents	Predicted Res2 Run1	
	Negative (0/6)	Positive (7/10)
Cases	9,060	20,085
0	8%	0%
1	3%	0%
2	4%	0%
3	6%	0%
4	6%	0%
5	39%	1%
6	28%	1%
7	2%	23%
8	3%	29%
9	1%	18%
10	1%	27%
Net 0 to 6	93%	2%
Net 7 to 10	7%	98%

Q4a Likely to Recommend

Correctly predicted : 96.31%

Column Percents	Predicted Res2 Run2	
	Negative (0/6)	Positive (7/10)
Cases	9,059	20,086
0	8%	0%
1	3%	0%
2	4%	0%
3	6%	0%
4	6%	0%
5	39%	1%
6	28%	1%
7	2%	23%
8	3%	29%
9	1%	18%
10	1%	27%
Net 0 to 6	93%	2%
Net 7 to 10	7%	98%

Q4a Likely to Recommend

Correctly predicted : 96.30%

Column Percents	Predicted Res2 Run3	
	Negative (0/6)	Positive (7/10)
Cases	9,061	20,084
0	8%	0%
1	3%	0%
2	4%	0%
3	6%	0%
4	6%	0%
5	39%	1%
6	28%	1%
7	2%	23%
8	3%	29%
9	1%	18%
10	1%	27%
Net 0 to 6	93%	2%
Net 7 to 10	7%	98%

Q4a Likely to Recommend

Correctly predicted : 96.31%

Column Percents	Predicted Res2 Run4	
	Negative (0/6)	Positive (7/10)
Cases	9,062	20,083
0	8%	0%
1	3%	0%
2	4%	0%
3	6%	0%
4	6%	0%
5	39%	1%
6	28%	1%
7	2%	23%
8	3%	29%
9	1%	18%
10	1%	27%
Net 0 to 6	93%	2%
Net 7 to 10	7%	98%

Q4a Likely to Recommend

Correctly predicted : 96.31%

Column Percents	Predicted Res2 Run5	
	Negative (0/6)	Positive (7/10)
Cases	9,061	20,084
0	8%	0%
1	3%	0%
2	4%	0%
3	6%	0%
4	6%	0%
5	39%	1%
6	28%	1%
7	2%	23%
8	3%	29%
9	1%	18%
10	1%	27%
Net 0 to 6	93%	2%
Net 7 to 10	7%	98%

Q4a Likely to Recommend

Correctly predicted : 96.30%

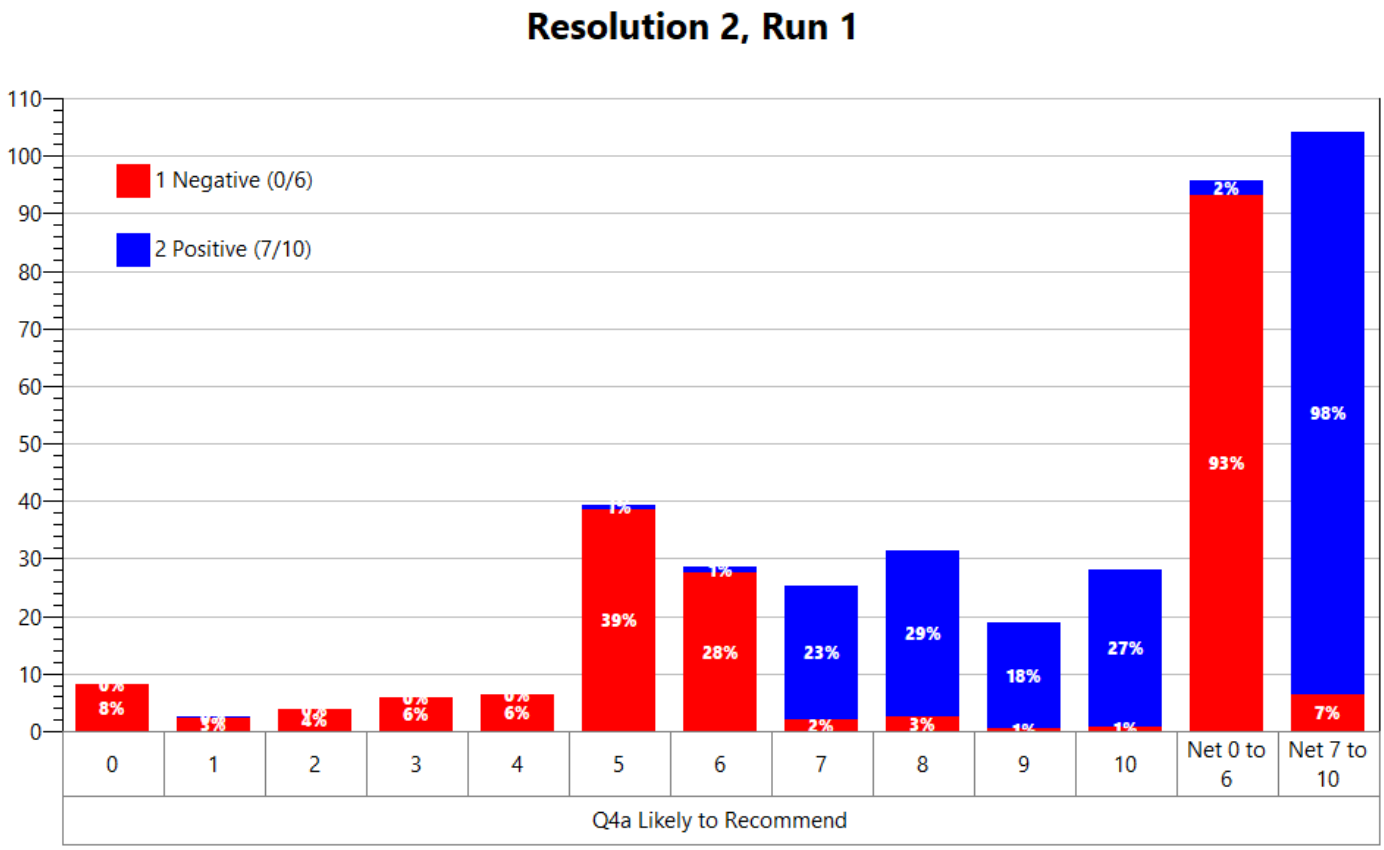
Frequencies Column Percents	Predicted Res2 Run1	
	Negative (0/6)	Positive (7/10)
Cases	9,060	20,085
Net 0 to 6	8,460	478
	93%	2%
Net 7 to 10	600	19,607
	7%	98%

Q4a Likely to Recom

	Min	Max	Difference
Negative	9059	9062	**3**
Positive	20083	20086	**3**

The runs are very stable, with identical percentages at 0DP, and a max-min difference of only 3 for both negative and positive base counts (at the Cases row).

As a chart for Run1:



Looking at the two Net columns at the right: 93% (red) + 7% (red) = 100%, and 2% (blue) + 98% (blue) = 100%.

At quad resolution:

Correctly predicted : 74.11%

Column Percents		Predicted Res4 Run1			
		Very Neg (0/3)	Some Neg (4/6)	Some Pos (7/8)	Very Pos (9/10)
Q4a Likely to Rec	Cases	346	7,640	11,754	9,405
	Net 0 to 3	61%	14%	4%	2%
	Net 4 to 6	29%	66%	11%	5%
	Net 7 to 8	8%	14%	74%	12%
	Net 9 to 10	3%	6%	10%	81%

Correctly predicted : 74.24%

Column Percents		Predicted Res4 Run2			
		Very Neg (0/3)	Some Neg (4/6)	Some Pos (7/8)	Very Pos (9/10)
Q4a Likely to Rec	Cases	909	7,010	11,294	9,932
	Net 0 to 3	47%	13%	3%	2%
	Net 4 to 6	36%	68%	11%	6%
	Net 7 to 8	11%	13%	76%	12%
	Net 9 to 10	6%	6%	9%	79%

Correctly predicted : 74.00%

Column Percents		Predicted Res4 Run3			
		Very Neg (0/3)	Some Neg (4/6)	Some Pos (7/8)	Very Pos (9/10)
Q4a Likely to Rec	Cases	93	9,064	11,076	8,912
	Net 0 to 3	71%	15%	3%	2%
	Net 4 to 6	18%	62%	9%	4%
	Net 7 to 8	5%	15%	77%	11%
	Net 9 to 10	5%	8%	11%	83%

Correctly predicted : 74.42%

Column Percents		Predicted Res4 Run4			
		Very Neg (0/3)	Some Neg (4/6)	Some Pos (7/8)	Very Pos (9/10)
Q4a Likely to Rec	Cases	147	8,352	10,944	9,702
	Net 0 to 3	63%	15%	3%	2%
	Net 4 to 6	27%	64%	10%	5%
	Net 7 to 8	5%	14%	78%	12%
	Net 9 to 10	4%	7%	9%	80%

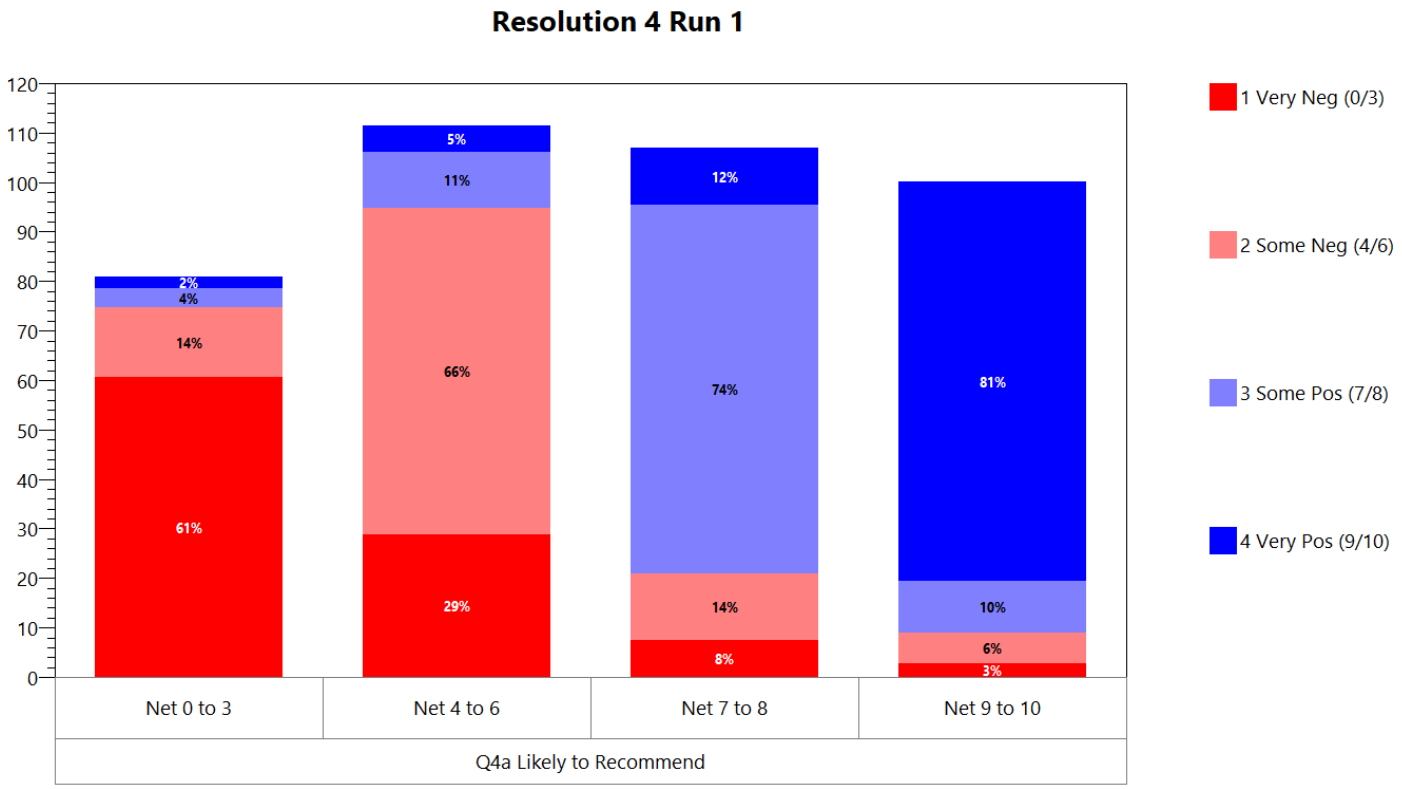
Correctly predicted : 74.63%

Column Percents		Predicted Res4 Run5			
		Very Neg (0/3)	Some Neg (4/6)	Some Pos (7/8)	Very Pos (9/10)
Q4a Likely to Rec	Cases	437	7,972	10,989	9,747
	Net 0 to 3	55%	14%	3%	2%
	Net 4 to 6	32%	65%	10%	6%
	Net 7 to 8	10%	14%	78%	12%
	Net 9 to 10	3%	7%	9%	80%

- Ratio of correct predictions has dropped to a 74.26% average
- The base counts are now quite disparate (especially for the *Very Neg (0/3)* columns, with max=909, min=93)
- Cells %s are no longer identical

But there is still a strong diagonal structure.

And as a chart for Run1:



At the highest resolution, predicting Q4a itself:

Correctly predicted : 46.86%

Column Percents	Predicted Res11 Run1										
Q4a Likely to Recommend	0	1	2	3	4	5	6	7	8	9	10
Cases	420	6	17	88	38	4,256	3,633	3,472	6,926	1,833	8,456
0	41%	33%	12%	9%	8%	5%	4%	1%	1%	0%	1%
1	3%	50%	6%	6%		2%	2%	0%	0%	0%	0%
2	2%		53%	5%	5%	3%	3%	1%	1%	0%	1%
3	6%		6%	42%	8%	4%	4%	1%	1%	1%	1%
4	5%			8%	42%	5%	5%	1%	1%	0%	1%
5	18%	17%	12%	22%	11%	44%	26%	4%	4%	2%	3%
6	11%		12%	8%	13%	18%	34%	4%	3%	2%	3%
7	5%				8%	6%	7%	48%	31%	5%	5%
8	6%			1%		7%	9%	31%	50%	10%	8%
9	2%				3%	2%	3%	3%	3%	48%	28%
10	2%				3%	3%	4%	5%	4%	31%	51%

Correctly predicted : 46.50%

Column Percents	Predicted Res11 Run2										
Q4a Likely to Recommend	0	1	2	3	4	5	6	7	8	9	10
Cases	133	7	43	20	36	4,447	3,000	3,405	8,312	985	8,757
0	54%	29%	9%	10%	14%	7%	4%	2%	1%	0%	1%
1	5%	43%	7%	5%		2%	2%	1%	0%	0%	0%
2	1%		53%	5%	6%	3%	2%	1%	1%	0%	1%
3	4%	14%	5%	60%	8%	4%	4%	1%	1%	1%	1%
4	4%		2%		42%	5%	5%	2%	1%	0%	1%
5	17%	14%	12%	20%	14%	44%	28%	6%	5%	3%	3%
6	6%		7%		11%	19%	35%	6%	4%	3%	3%
7	2%		2%		3%	6%	6%	46%	29%	4%	4%
8	6%					6%	8%	27%	47%	7%	7%
9	1%					2%	3%	3%	4%	54%	29%
10	1%		2%		3%	3%	3%	6%	5%	27%	50%

Correctly predicted : 46.38%

Column Percents	Predicted Res11 Run4										
	0	1	2	3	4	5	6	7	8	9	10
Q4a Likely to Recommend: Cases	124	20	28	15	22	5,041	2,935	2,310	8,938	1,556	8,156
0	55%	15%	11%	13%	18%	7%	4%	2%	1%	0%	1%
1	3%	55%	7%			2%	1%	1%	0%	0%	0%
2	2%		54%	7%	5%	3%	2%	1%	1%	0%	1%
3	5%	15%	11%	53%	5%	5%	4%	1%	1%	1%	1%
4	5%	5%			50%	5%	4%	2%	1%	0%	1%
5	15%	10%	7%	27%		42%	26%	5%	5%	3%	3%
6	6%		7%		14%	19%	35%	5%	4%	2%	2%
7	2%				9%	6%	7%	49%	31%	4%	4%
8	7%					6%	9%	27%	46%	7%	7%
9	1%		4%			2%	3%	2%	4%	51%	28%
10						3%	4%	5%	5%	32%	51%

Correctly predicted : 46.18%

Column Percents	Predicted Res11 Run3										
	0	1	2	3	4	5	6	7	8	9	10
Q4a Likely to Recommend: Cases	399	5	12	16	47	5,727	1,817	6,420	4,767	3,091	6,844
0	41%	20%	8%	19%	11%	6%	2%	1%	1%	1%	1%
1	4%	60%	8%			2%	1%	0%	0%	0%	0%
2	3%		42%	13%	9%	3%	2%	1%	1%	0%	1%
3	6%		17%	50%	6%	5%	3%	1%	1%	1%	1%
4	4%				47%	5%	4%	2%	1%	0%	1%
5	17%	20%		19%	11%	40%	28%	5%	5%	3%	3%
6	12%		17%		13%	21%	37%	5%	4%	2%	3%
7	4%				2%	6%	7%	43%	26%	4%	4%
8	7%					7%	9%	35%	51%	9%	7%
9	2%		8%			2%	3%	3%	4%	45%	26%
10	2%				2%	3%	4%	5%	5%	34%	54%

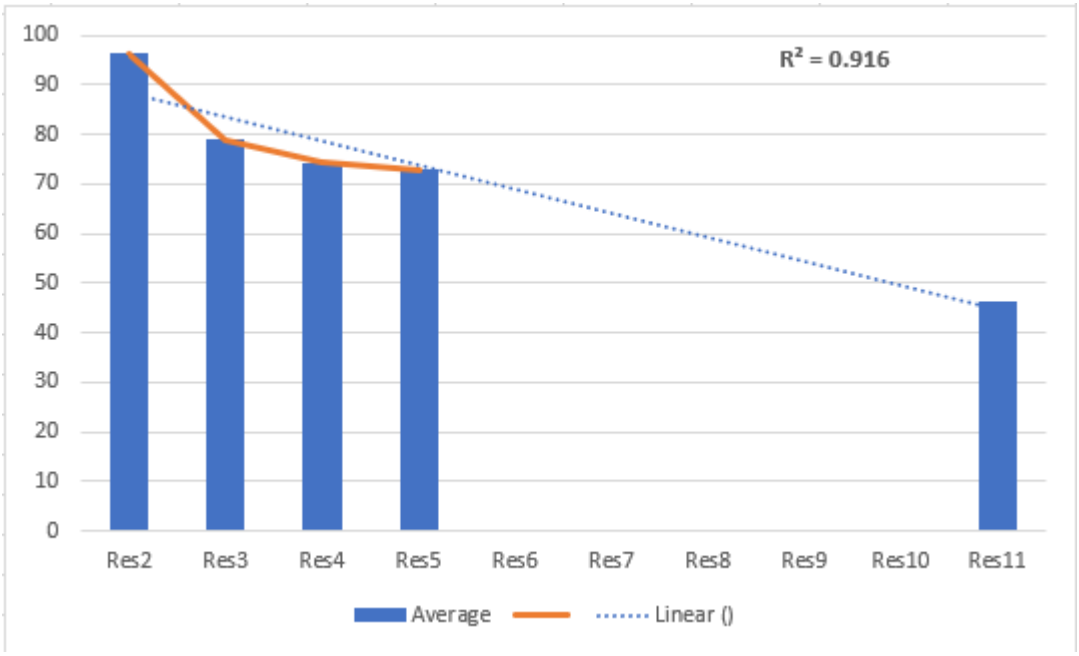
Correctly predicted : 46.42%

Column Percents	Predicted Res11 Run5										
	0	1	2	3	4	5	6	7	8	9	10
Q4a Likely to Recommend: Cases	214	7	13	11	25	3,859	3,607	2,164	9,059	2,455	7,731
0	47%	29%	8%	18%	12%	7%	4%	2%	1%	0%	1%
1	5%	43%	8%			2%	2%	0%	0%	0%	0%
2	3%		62%	9%	8%	3%	3%	1%	1%	0%	1%
3	6%	14%	8%	55%	8%	5%	4%	1%	1%	1%	1%
4	5%				44%	5%	5%	2%	1%	0%	1%
5	17%	14%	8%	18%	12%	46%	27%	5%	5%	3%	3%
6	8%		8%		12%	18%	34%	5%	4%	2%	3%
7	3%					5%	7%	51%	32%	5%	4%
8	5%					5%	8%	25%	46%	9%	7%
9	0%					2%	3%	3%	4%	45%	27%
10	1%				4%	3%	4%	5%	5%	34%	51%

- Now only 46.5% correct
- A lot more noise
- Overlaps on 5/6, 7/8, 9/10
- Base and % disparities

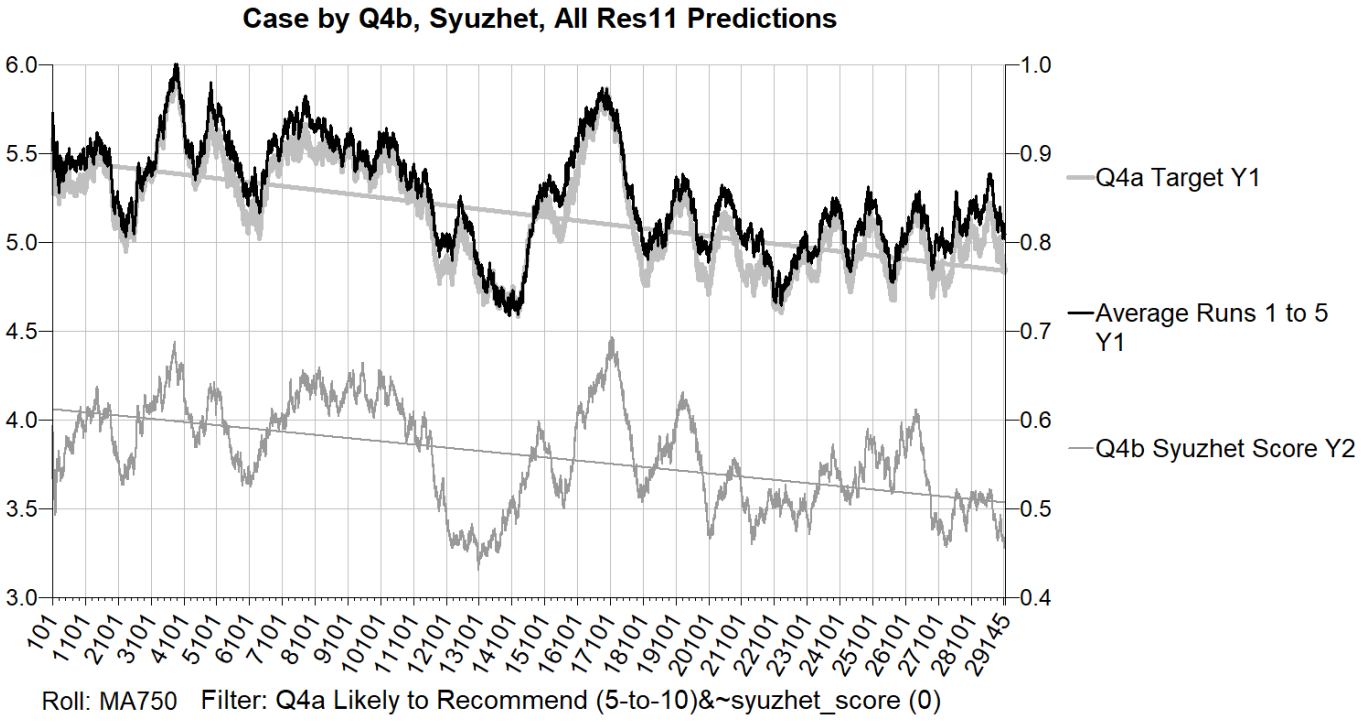
The results at all resolutions are

	Runs					
	1	2	3	4	5	Avg Correct
Res2	96.3	96.31	96.3	96.31	96.31	96.30
Res3	78.27	78.84	78.84	79.11	79.31	78.87
Res4	74.11	74.24	74	74.42	74.63	74.28
Res5	73.05	72.95	72.9	72.93	73.07	72.98
Res11	46.86	46.5	46.18	46.38	46.42	46.47



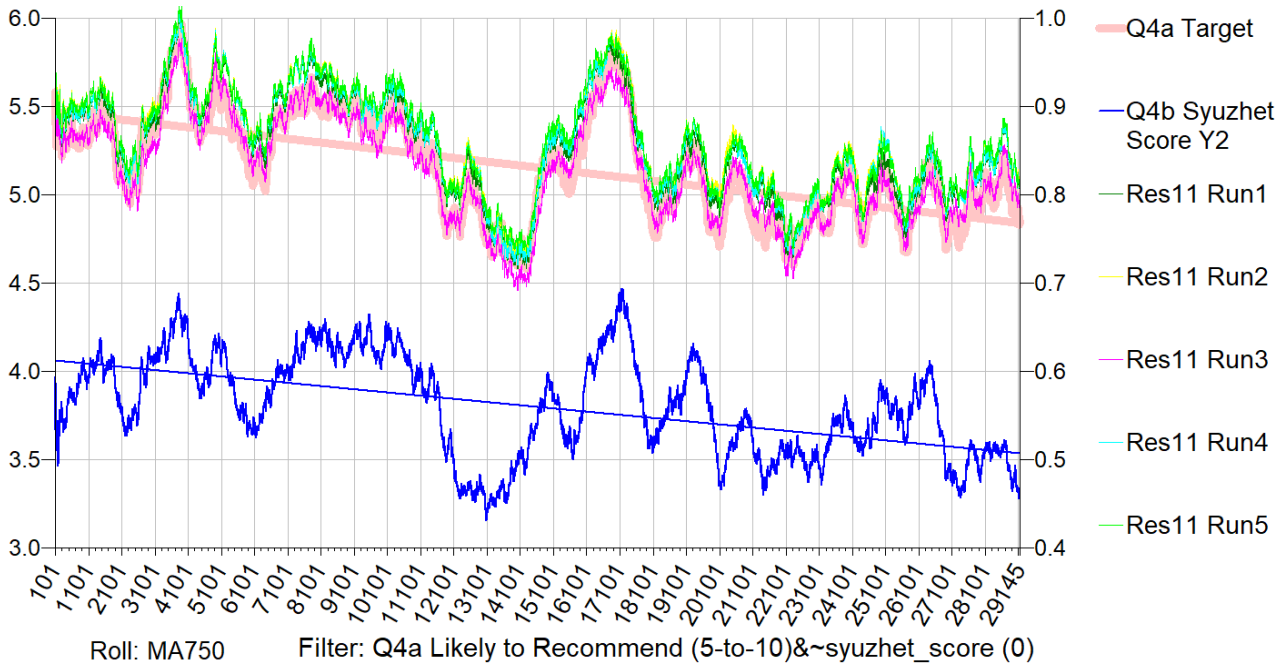
The ratios at each resolution are stable across all runs, varying only by tenths of a percent. Resolutions 3, 4 and 5 degrade naturally, from average 78.87% down to 72.98%. Resolution 2 (binary) at 96.3% is best, but that is the easiest of targets. The big drop to 46.47% for resolution 11 is because there are no further even subdivisions to eleven targets. Res11 is nonetheless exactly where expected on a linear trend prediction, with strong $R^2 = 0.916$.

Is resolving to 11 points too ambitious? The model runs are on the full case data, including nonsense verbatim cases and verbatims with no measurable sentiment content. The Syuzhet plot was smoothed and filtered to syuzhet_score not 0 and Q4a is 5 to 10, so for a fair comparison, we must here do likewise.



The fat grey series is the prediction target. The black series sitting on top is the average of the five runs at resolution 11.

The five actual runs comprising the average are nearly identical, and although a little higher, all have better coherence with the target than the Syuzhet plot, particularly around case 12,500 and from cases 26,000 ff.



Correlation

The final check for validity is to run standard pairwise correlations.

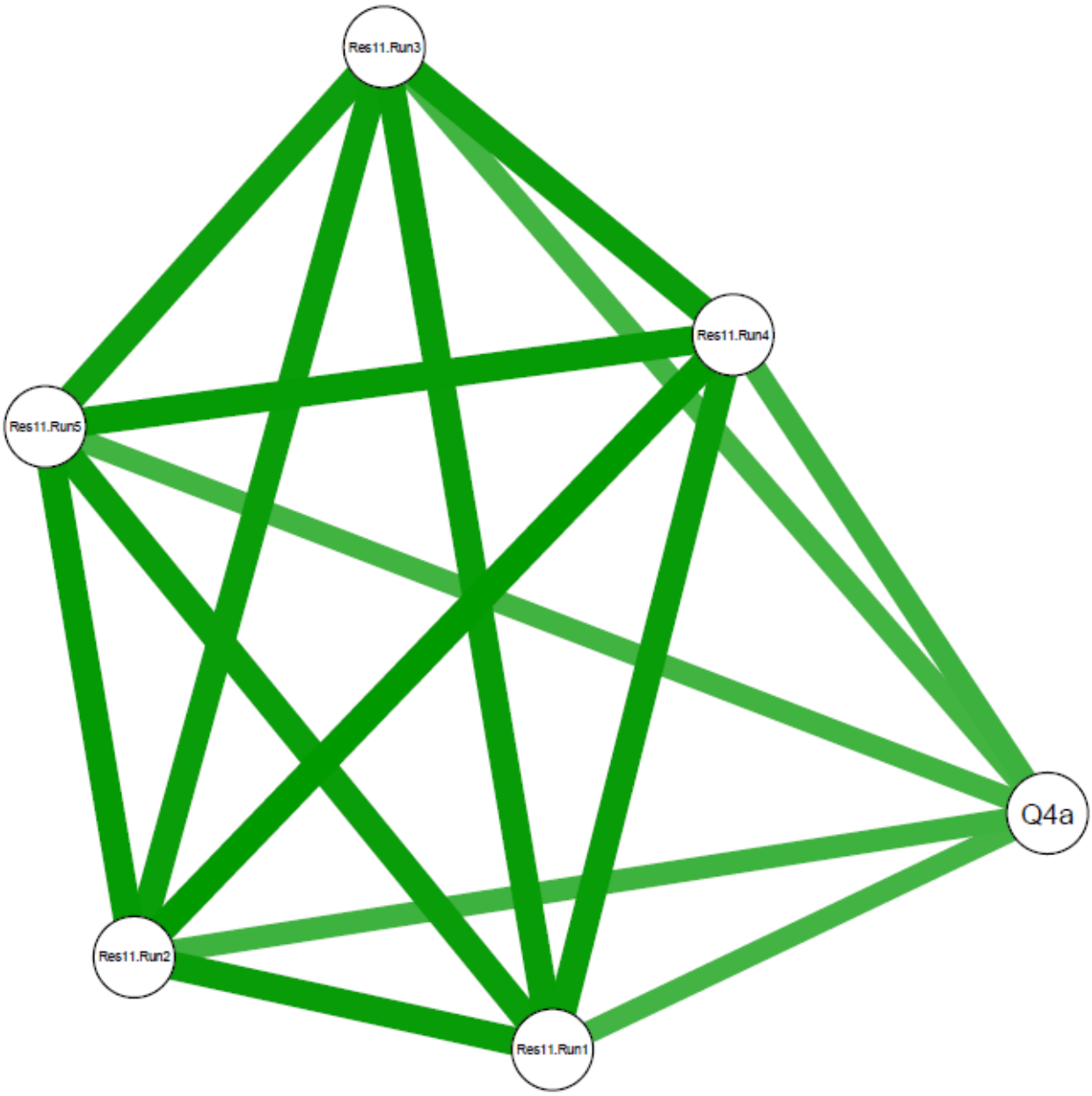
First, using the per case input matrix as 29,145 rows (one per case) of the form

Filter: Q4a Likely to Recommend (5-to-10)&~syuzhet_score (0)

Frequencies		Q4a	Res11 Run1	Res11 Run2	Res11 Run3	Res11 Run4	Res11 Run5
CaseSeq	1	8	8	8	7	8	8
	2	7	8	8	8	8	8
	3	7	8	8	8	8	8
	4	8	7	7	7	7	7
	6	7	6	7	7	6	6
	7	6	5	5	5	5	5
	9	10	10	10	10	10	10
	11	10	10	10	10	10	10
	14	9	10	10	10	10	10
	15	10	10	10	10	10	10
	16	9	9	10	9	10	10
	17	6	6	8	7	8	8
	18	10	10	10	10	10	10
	19	10	9	10	9	9	9
	20	10	10	10	10	10	10
	21	7	8	8	8	8	8
	23	10	10	10	10	10	10

gives an average correlation (using R's Pearson Pairwise-Complete) of 0.71.

	Q4a	Res11.Run1	Res11.Run2	Res11.Run3	Res11.Run4	Res11.Run5
Q4a	1	0.701	0.72	0.704	0.725	0.711
Res11.Run1	0.701	1	0.917	0.925	0.919	0.917
Res11.Run2	0.72	0.917	1	0.912	0.949	0.938
Res11.Run3	0.704	0.925	0.912	1	0.916	0.903
Res11.Run4	0.725	0.919	0.949	0.916	1	0.933
Res11.Run5	0.711	0.917	0.938	0.903	0.933	1



The runs correlate very well with each other, but not as well as hoped for the target Q4a.

However, if the correlation is performed on the aggregated verbatims instead, justified by the fact that common responses will often have a different respondent-assigned per-case target, the correlation score averages over 0.99.

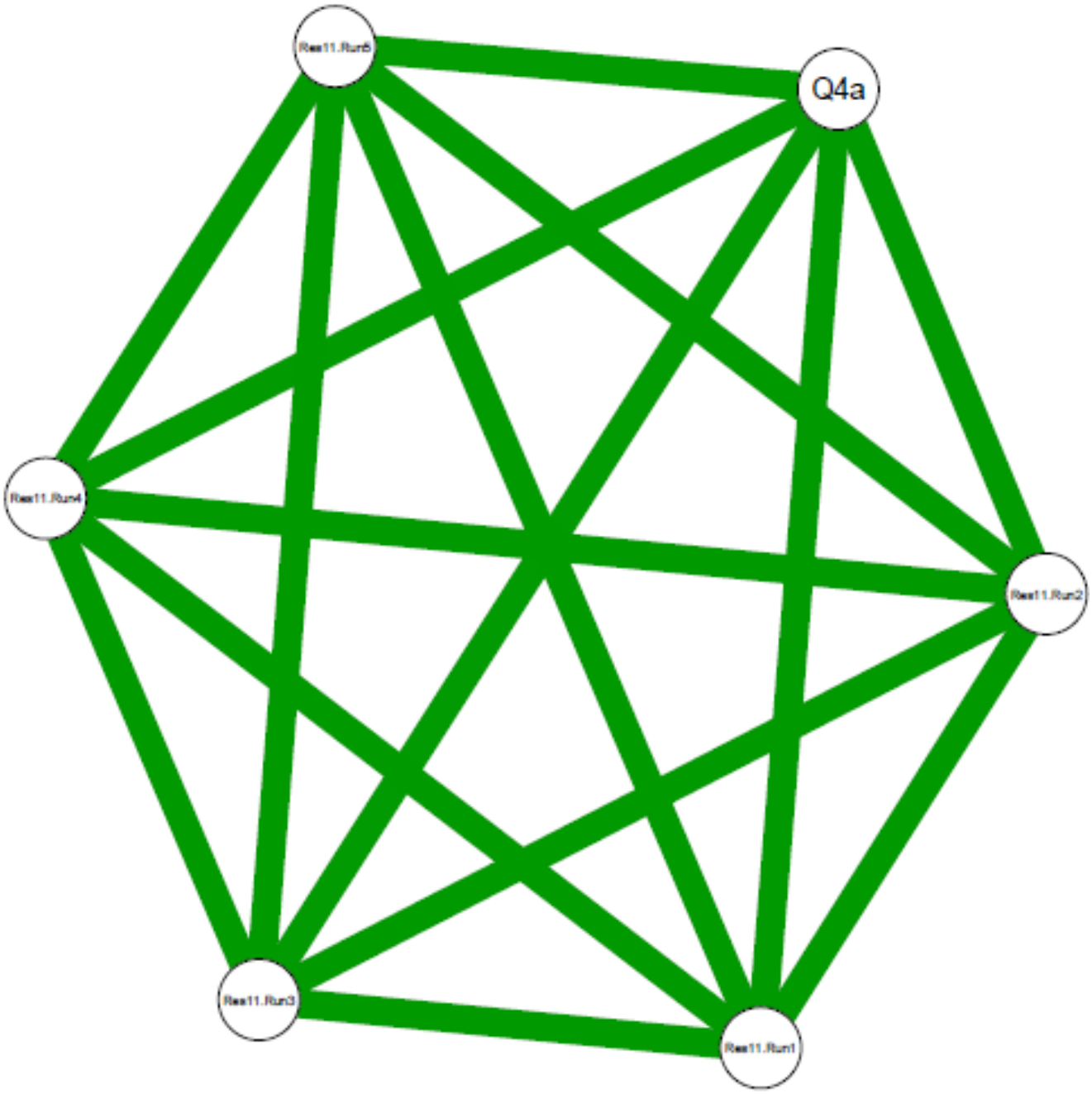
The input matrix is 17,275 rows (being the number of unique responses) of the form

Filter: Q4a Likely to Recommend (5-to-10)&~syuzhet_score (0)

Frequencies	Q4a	Res11 Run1	Res11 Run2	Res11 Run3	Res11 Run4	Res11 Run5
100% refund	8	8	8	7	8	8
AFTER 60 YEARS WITH THEM AND NO	10	10	10	10	10	10
All good	8	9	10	9	10	9
Always give good service, no hassles	10	10	10	10	10	10
always used them never had any probl	10	9	9	9	9	9
best service best fees least gaps best in	9	10	10	10	10	10
better packages for all insurance and di	7	8	8	7	8	8
better rates - less hassle	7	7	7	7	7	7
Better Rebates	129	126	126	126	144	126
Better use of limits on benefits. Being	7	7	7	7	7	7
cheaper ins and better payouts	8	8	8	7	8	8

Note the aggregation of common verbatims such as *Better rebates*. The correlation matrix gives better than 0.99 for each run against Q4a, with average as 0.9924, and the correlation of the runs against each other as 0.9967.

	Q4a	Res11.Run1	Res11.Run2	Res11.Run3	Res11.Run4	Res11.Run5
Q4a	1	0.993	0.993	0.992	0.992	0.992
Res11.Run1	0.993	1	0.997	0.997	0.996	0.996
Res11.Run2	0.993	0.997	1	0.996	0.998	0.997
Res11.Run3	0.992	0.997	0.996	1	0.997	0.996
Res11.Run4	0.992	0.996	0.998	0.997	1	0.997
Res11.Run5	0.992	0.996	0.997	0.996	0.997	1



The node plot shows the correlations between all runs and the target are practically equal.

Conclusions

- The StochasticDualCoordinateAscentClassifier is a better predictor than Syuzhet.
- Consistent correct ratios across runs.
- Expected degradation with increasing resolution
- Excellent match on de-noised and smoothed per-case plots.
- De-noised correlation on aggregate responses > 0.99.

The model can therefore be deemed to be sufficiently valid (in that we are measuring what we think we are measuring) to deploy against unseen cases.

Notes:

1. G Sanderson, "Gradient descent, how neural networks learn", online video presentation at https://www.youtube.com/c/3blue1brown
2. Michael A. Nielsen, "Neural Networks and Deep Learning", Determination Press, 2015, chapter 1. Online at http://neuralnetworksanddeeplearning.com/chap1.html
3. V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions and reversals", Doklady Akademii Nauk SSSR 163(4) p845-848, 1965, also Soviet Physics Doklady 10(8) p707-710, Feb 1966.
4. E. Ukkonen, "On approximate string matching", Proc. Int. Conf. on Foundations of Comp. Theory, Springer-Verlag, LNCS 158 p487-495, 1983.
5. M. Jockers, "Introduction to the Syuzhet Package", 2017, https://cran.r-project.org/web/packages/syuzhet/vignettes/syuzhet-vignette.html. For API documentation see https://cran.r-project.org/web/packages/syuzhet/syuzhet.pdf
6. ML.Net is a new AI platform by Microsoft. For general introduction see https://dotnet.microsoft.com/apps/machinelearning-ai/ml-dotnet
7. Documentation and discussion are online at https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.legacy.trainers.stochasticdualcoordinateascentclassifier

[end]