<u>Update:</u> Probabilistic Scoring

Client: Alan Malik, Patient Tools INC

Consultant: Lee Panter

Let's tell someone how they're feeling:

An Example Evaluation of a Probabilistic Score (P-Score) Calculation for a PHQ9

Lots of PHQ9s

Over the last 2 weeks, how often have you been bothered by any of the following problems?	**Not at all**	**Several Days**	**More than half the days**	**Nearly every day**
1. Little interest or pleasure in doing things	0	1	2	3
2. Feeling down, depressed, or hopeless	0	1	2	3
3. Trouble falling or staying asleep, or sleeping too much	0	1	2	3
4. Feeling tired or having little energy	0	1	2	3
5. Poor appetite or overeating	0	1	2	3
6. Feeling bad about yourself – or that you are a failure or have let yourself or your family down	0	1	2	3
7. Trouble concentrating on things, such as reading the newspaper or watching television	0	1	2	3
8. Moving or speaking so slowly that other people could have noticed. Or the opposite – being so fidgety or restless that you have been moving around a lot more than usual	0	1	2	3
9. Thoughts that you would be better off dead, or of hurting yourself in some way	0	1	2	3

i is the Depression Classification
i=1,2,3

j is the question number
j=1,2,3,4,5,6,7,8,9

k is the provided answer magnitude: k=1,2,3,4

Calculate the "look-up weights"

	Not Clincally Depressed				Sub-Threshold Depression				Major Depression			
	0	1	2	3	0	1	2	3	0	1	2	3
PHQ-1	1.535921	0.736532	0.254765	0.330620	0.494097	2.054314	1.236479	0.527800	0.301661	0.975541	2.184291	2.376285
PHQ-2	1.645223	0.701553	0.167270	0.016527	0.485251	2.332852	0.714763	0.188321	0.116167	0.909036	2.574183	3.075907
PHQ-3	1.679384	1.069410	0.451923	0.154546	0.382142	1.880743	1.397759	0.874984	0.104003	0.477565	1.768789	2.523103
PHQ-4	1.697888	1.136584	0.473656	0.161893	0.332800	1.653902	1.337789	0.792300	0.094443	0.464641	1.758481	2.548111
PHQ-5	1.547740	0.876037	0.185321	0.129121	0.569675	2.103316	1.106140	0.472926	0.246663	0.711266	2.364211	2.750739
PHQ-6	1.517303	0.569349	0.036616	0.016880	0.661431	2.531790	0.625862	0.235097	0.257552	1.047459	2.841324	3.053939
PHQ-7	1.452866	0.530565	0.116992	0.019785	0.739826	2.273619	0.525170	0.225445	0.333498	1.232574	2.747919	3.053310
PHQ-8	1.346661	0.347380	0.102221	0.039712	0.775626	2.247434	0.388266	0.150837	0.501317	1.562172	2.836033	3.052816
PHQ-9	1.185392	0.315133	0.026666		0.835730	2.162279	0.506434		0.753520	1.656964	2.913099	3.190537

Used to "update" a probabilistic score for being classified "Not Clinically Depressed" when they answers "3" on question # 2

$$P(C_i \mid E_j = k) = \frac{P(E_j = k \mid C_i)}{P(E_j = k)} * P(C_i)$$

- One observation in the data:

(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9)=(2, 2, 3, 3, 3, 2, 3, 0, 0)   Sum=18

- Traditional scoring algorithm assigns (C-Score) of: C3

C1 = sum scores {0,…,6} C2 = sum scores {7,…,9} C3 = sum scores {10,…,27}

Use "full data Look-up weights" to *Probabilistically* classify observation

- An observation converges to a Probabilistic Score class P_i if the Probabilistic Score (P-score) for that class exceeds 75%
- Starting P-Score is uninformative: $(P_1^0, P_2^0, P_3^0) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$
- Successive P-Scores are defined recursively:

$$(P_1^j, P_2^j, P_3^j) = (\,P_1^{j-1}\,W_{1k}^j, P_2^{j-1}\,W_{2k}^j, P_3^{j-1}\,W_{3k}^j) \quad for \quad j = 1, \dots, 9 \text{ k=1,2,3,4}$$

- W_{ik}^j are the lookup weights:
    - Depression Classification-i=1,2,3
    - after answering question j=1,2,…,9,
    - proving response k=0,1,2,3

(Q1, Q2, Q3, Q4, Q5, Q6, Q7, Q8, Q9)=(2, 2, 3, 3, 3, 2, 3, 0, 0)

	Not Clincally Depressed				Sub-Threshold Depression				Major Depression			
	0	1	2	3	0	1	2	3	0	1	2	3
PHQ-1	1.535921	0.736532	0.254765	0.330620	0.494097	2.054314	1.236479	0.527800	0.301661	0.975541	2.184291	2.376285
PHQ-2	1.645223	0.701553	0.167270	0.016527	0.485251	2.332852	0.714763	0.188321	0.116167	0.909036	2.574183	3.075907
PHQ-3	1.679384	1.069410	0.451923	0.154546	0.382142	1.880743	1.397759	0.874984	0.104003	0.477565	1.768789	2.523103
PHQ-4	1.697888	1.136584	0.473656	0.161893	0.332800	1.653902	1.337789	0.792300	0.094443	0.464641	1.758481	2.548111
PHQ-5	1.547740	0.876037	0.185321	0.129121	0.569675	2.103316	1.106140	0.472926	0.246663	0.711266	2.364211	2.750739
PHQ-6	1.517303	0.569349	0.036616	0.016880	0.661431	2.531790	0.625862	0.235097	0.257552	1.047459	2.841324	3.053939
PHQ-7	1.452866	0.530565	0.116992	0.019785	0.739826	2.273619	0.525170	0.225445	0.333498	1.232574	2.747919	3.053310
PHQ-8	1.346661	0.347380	0.102221	0.039712	0.775626	2.247434	0.388266	0.150837	0.501317	1.562172	2.836033	3.052816
PHQ-9	1.185392	0.315133	0.026666		0.835730	2.162279	0.506434		0.753520	1.656964	2.913099	3.190537

$$(P_1^0, P_2^0, P_3^0) = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$$

$$(P_1^1, P_2^1, P_3^1) = (\ P_1^0\ W_{12}^1,\ P_2^0\ W_{22}^1, P_3^0\ W_{32}^1)$$

$$= (\frac{1}{3}\ 0.2545, \frac{1}{3}\ 1.2375, \frac{1}{3}\ 2.1843)$$

$$= (0.0848, 0.4125, 0.7281) \propto (0.0692, 0.3366, 0.5942)$$

$$(P_1^2, P_2^2, P_3^2) = (\ P_1^1\ W_{12}^2,\ P_2^1\ W_{22}^2, P_3^1\ W_{32}^2)$$

$$= (0.0692 * 0.1673, 0.3366 * 0.7148, 0.5942 * 2.5742)$$

$$= (0.0116, 0.2406, 1.5296) \propto (0.0065, 0.1350, \boxed{0.8585}) \longrightarrow P_3$$

This sequence takes only two iterations of the Probabilistic Scoring Algorithm to converge!

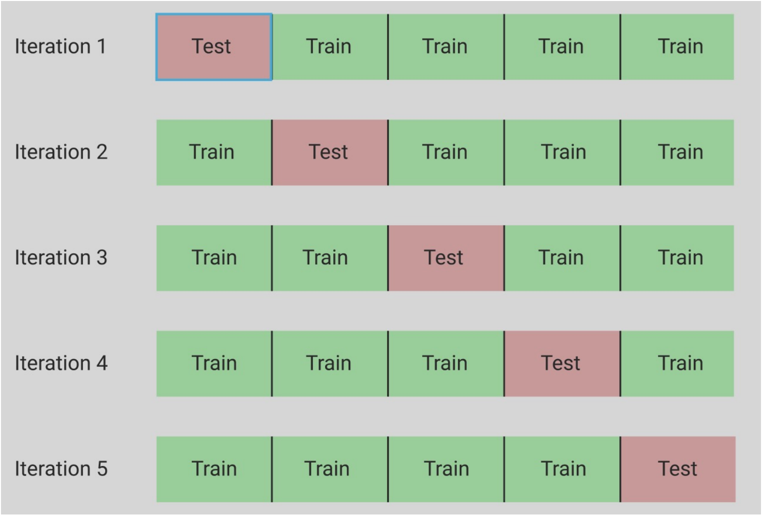
GREAT… NOW WHAT?

- “TRUE” classification. Called the TRUTH CLASS, D_i (same i index, but a fundamentally different class)
- D_i is a theoretical concept, threshold values cannot be defined to make D_i 100% accurate
- However, supposing those thresholds DID exist, the values 7 and 10 have been chosen so that the traditional method is as accurate as possible. Therefore, $C_i \approx D_i$ with the knowledge that $C_i \neq D_i$
- For your consideration:
  - C_i is 88% accurate
  - By choosing C_i to be as *close* to D_i as possible, those who took D_i away “embedded information about the TRUTH CLASS” into C_i
  - caveat: information is relevant ONLY to specific data from definition.

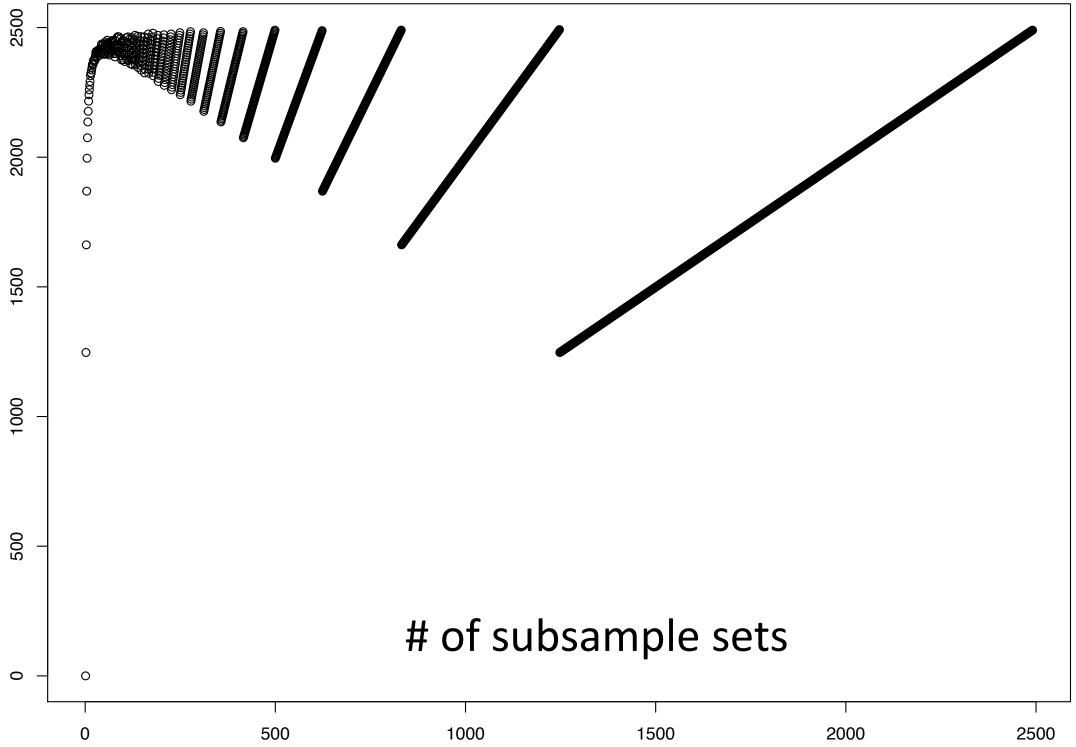
That…wasn't helpful

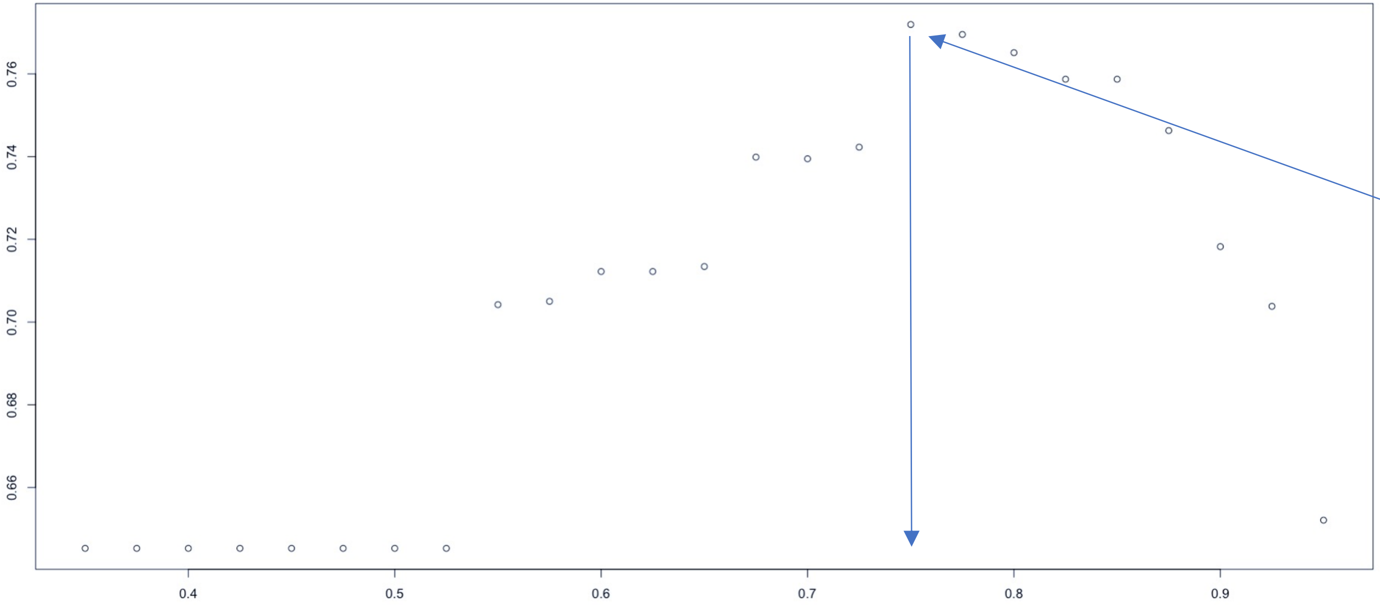
- Here are the two ideas I'm pursuing:
  - #1 – Make stuff up
    - C_i is 88% accurate. So simulate a new response
      - Start traditional classifications
      - Introduce 12% error (*induced classification outcome*)
      - Probability of error functionally dependent on:
        - absolute distance between nearest threshold and the traditional classification
        - number of observations within that same distance
    - Perform k-Cross Validation test/train analysis to compare accuracy of traditional method vs P-Scoring, using induced classification variable as the TRUTH CLASS
  - #2 – Make MORE stuff up
    - Use the Probabilistic Scoring algorithm itself to generate an outcome.
    - Use the same training data for the K-Cross Validation test/train process in #1 to create a "bootstrap" distribution of Lookup-weights.
    - Use the mean, or median, or whatever of each distribution to estimate a single value
    - Using the set of mean/median Lookup-weights, classify each observation. Use as the second type of TRUTH CLASS
    - Re-perform k-cross Validation accuracy comparison (use a different test/train sample)

K-Fold Cross-Validation Train/Test Sets



- I split data into 15 different values of K.
- Might be possible to show training size-dependency
- Accuracy is higher for k-values corresponding to higher count training sets



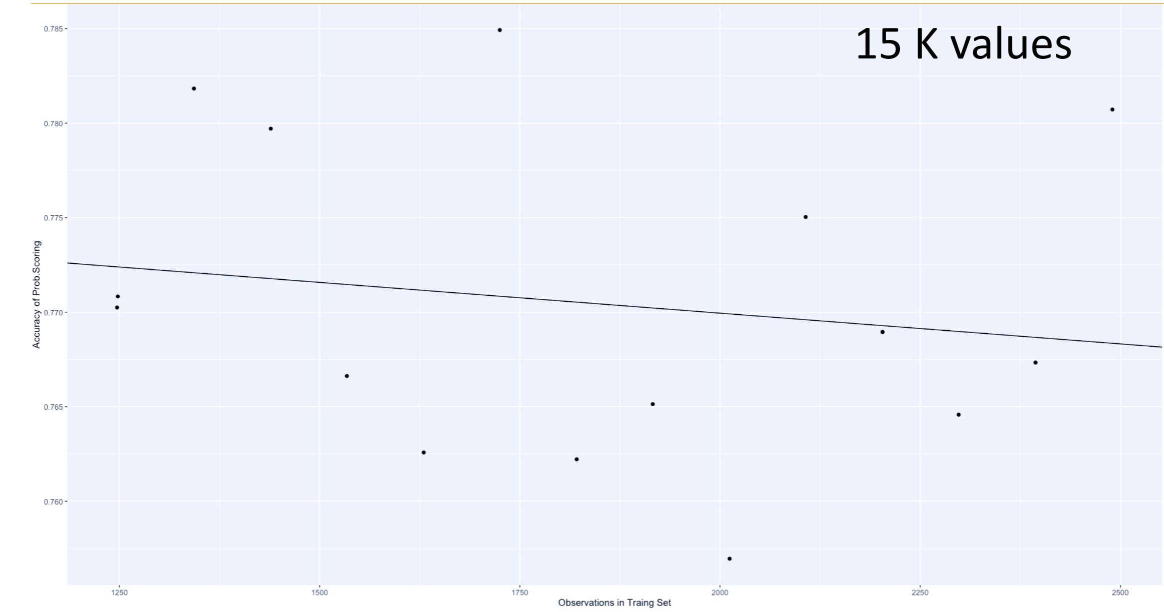


- I used the full data set to identify an adequate stopping probability threshold for the P-scoring algorithm.
- This value optimizes approach #1 accuracy
- This should probably be done for each k value, for each approach, on each data set, but that sounds…hard.

Initial Results Approach #1

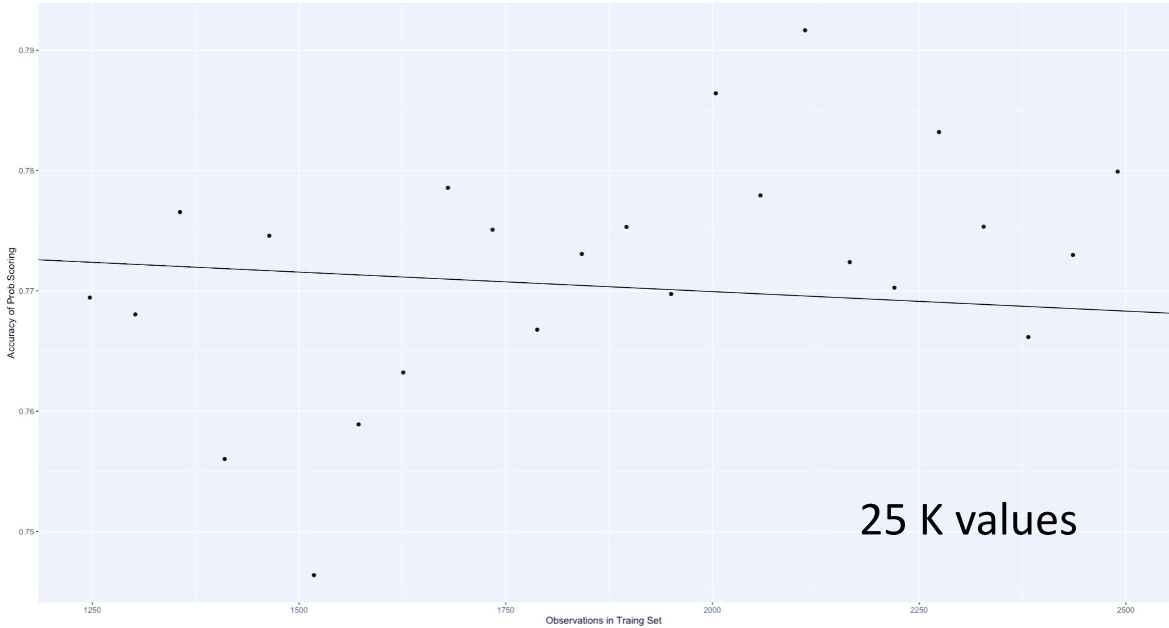
What's accomplished:

- Accuracy evaluation for P-score algorithm across 15 different values of cross fold validation training constructions
- Mean accuracy for each value of K used as comparison metric to traditional scoring (also evaluated on same test sets)
  - Minimum accuracy frequently resulted in "zero accuracy" for test sets with 3-7(ish) observations.
  - Did not try median, mode,…etc, but mean seems to be working… "OK"



What are the results:

- The P-Score algorithm has lower accuracy for all training data sizes compared to the traditional classification method.
- This is expected because the outcome is directly simulated from the traditional method
- More Importantly (further analysis still needed)
  - I might be able to show that accuracy increase with training sample size in the P-score algorithm
  - There may be an interaction between P-Score and traditional accuracy slopes



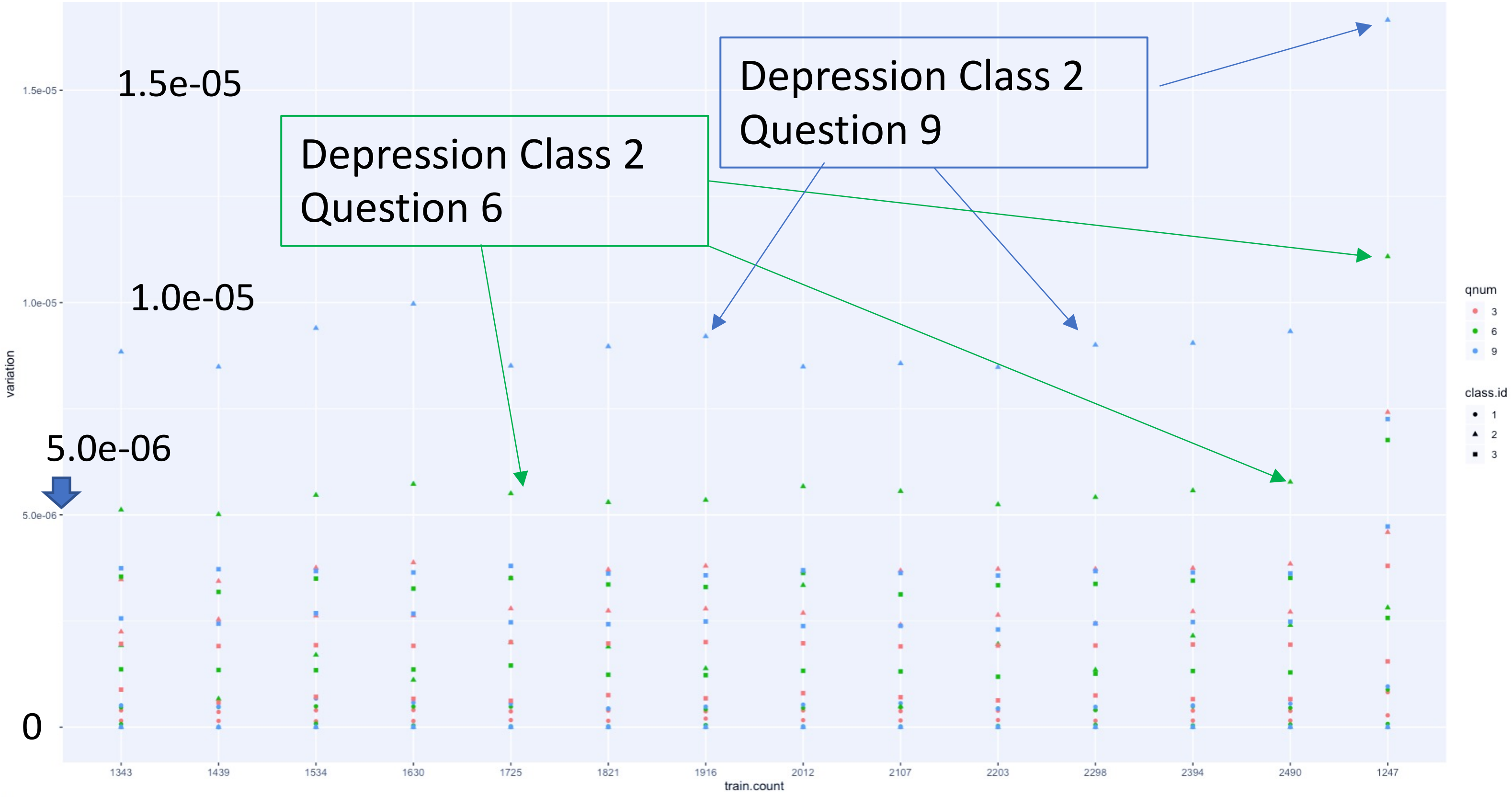
Initial Results Approach #2

What's Done:

- Variance plot for 17 different lookup weights, calculated for the 15 different values of K-cross fold training data sets
- Each value of K generated one bootstrap distribution for each Look-up weight
- WHY???
  - If this is a reasonable idea, the variances of the bootstrap distributions for each look-up weight should feature some of:
    - Convergent/converging to zero
    - Bounded above
    - Monotonic
    - "small" in value

What are the result's:

- The plot noted is displayed below…and the numbers are small
- Not the most powerful result, but I will most likely proceed with creating the 2nd outcome



What’s Leftover, and What’s the Timeline

- Things I still needs to do for the analysis:
  - Complete Analysis #2
    - Generate P-score outcomes
    - Repeat CV accuracy analysis
  - Get K iterations UP & Experiment approach #1 Training Set definitions
    - Limited success with k=50
    - Saving data has been an issue
- Projected Timeline
  - I will be finishing the analysis between 4/14 - 4/21
  - Following methods presentation

Questions?

- Keeping in mind that Alan is going for Peace of Mind, not “proof”, are there any other methods you suggest?

- Approach #2 is “hand wavy”. I get it
  - How can I justify this approach?
  - Note: my justification so far is “…why not?”

- I am still working on notation and nomenclature. I realize everything is confusing. Please let me know what is clear, so I can stop trying to un-confuse stuff.

THANK YOU!