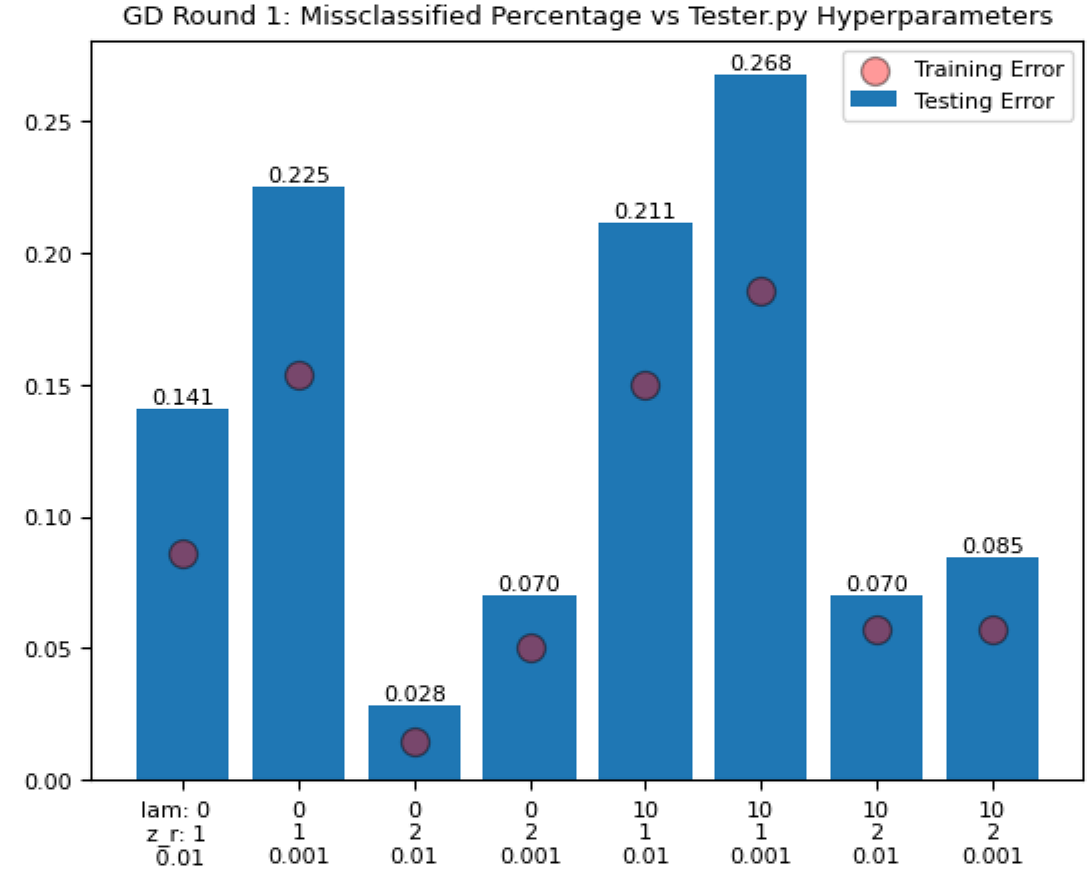
Schuyler Asplin
CSCD 439 Machine Learning
Professor Xu

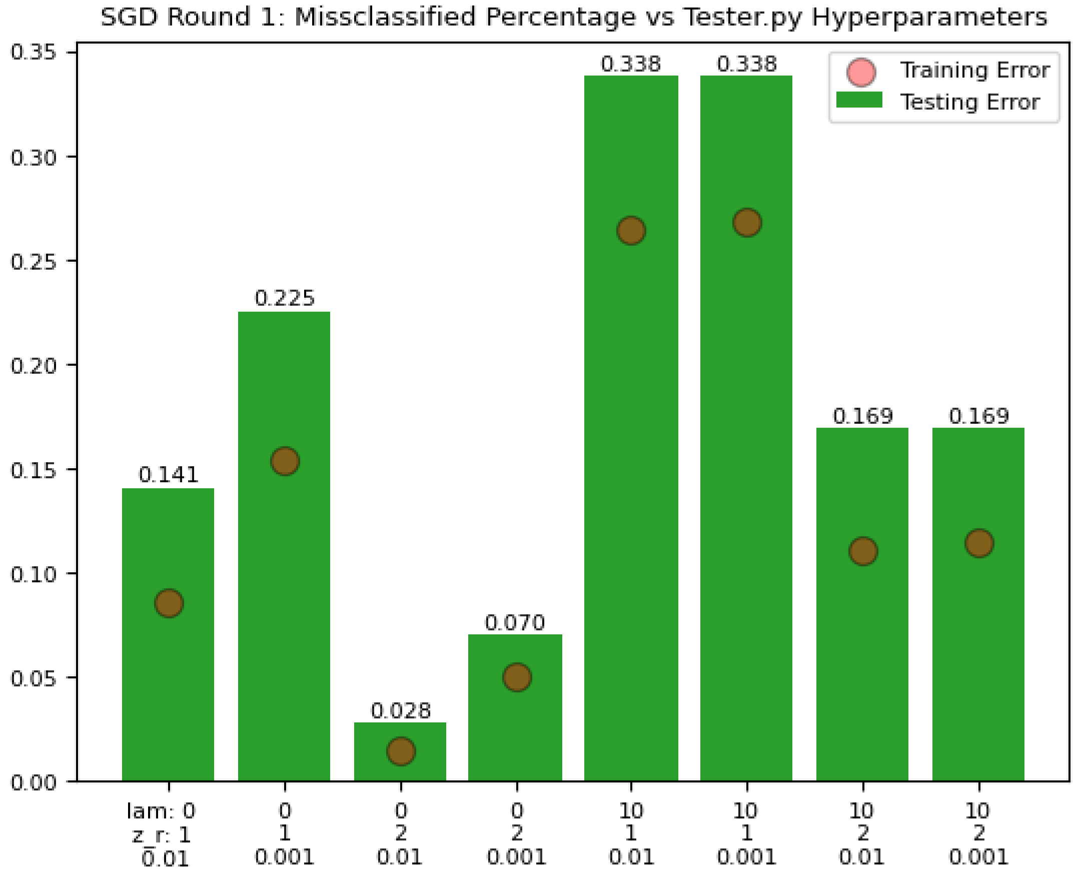
Training & Evaluation of Logistic Regression Model

(Predicting An Individuals Risk of Developing Diabetes)

Initial look at GD: This first test seems to suggest that higher z space was better, and an eta of 0.001 was closer to optimal than 0.01. I'm hesitant to draw any conclusions about lambda, since 0 might not give a good picture. It does seem safe to say that lambda of 10 is likely to be causing overfitting. Iterations are 10,000.



Taking a look at SGD: Again iterations are 10,000. The picture is similar although the results are worse and at lambda of 10, the eta’s make no difference whatsoever. This is because the percentage is of a discrete error count, meaning the bars are a function of an integer. The means that even if there are differences in the models’ quality, they can be subtle enough that they result in the same error count, and thus, the same percentage misclassified.

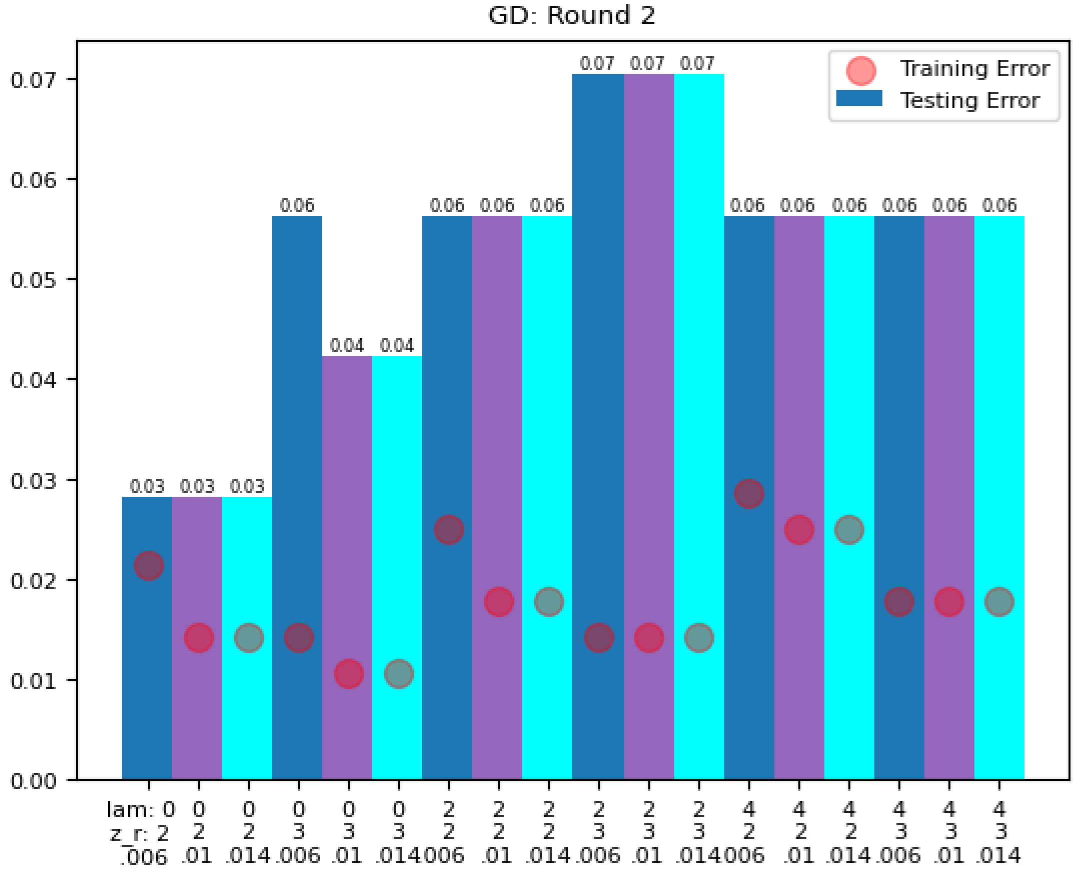


SECOND ROUND OF INVESTIGATION:

GD: I cycled in a similar fashion through lambda values of [0,2,4], z values of [2, 3], and eta values of [0.006, 0.01, 0.014]. The results were vast improvements over the initial run but didn't give as clear a direction with hyperparameter tuning.

Takeaways:

- Z space of 3 was uniformly worse than 2.
- Only lamda of 1 allowed the eta to make any difference. Next run, I'm going to test a lambda value between 0 and 1.
- Total time to run was 2:38
-

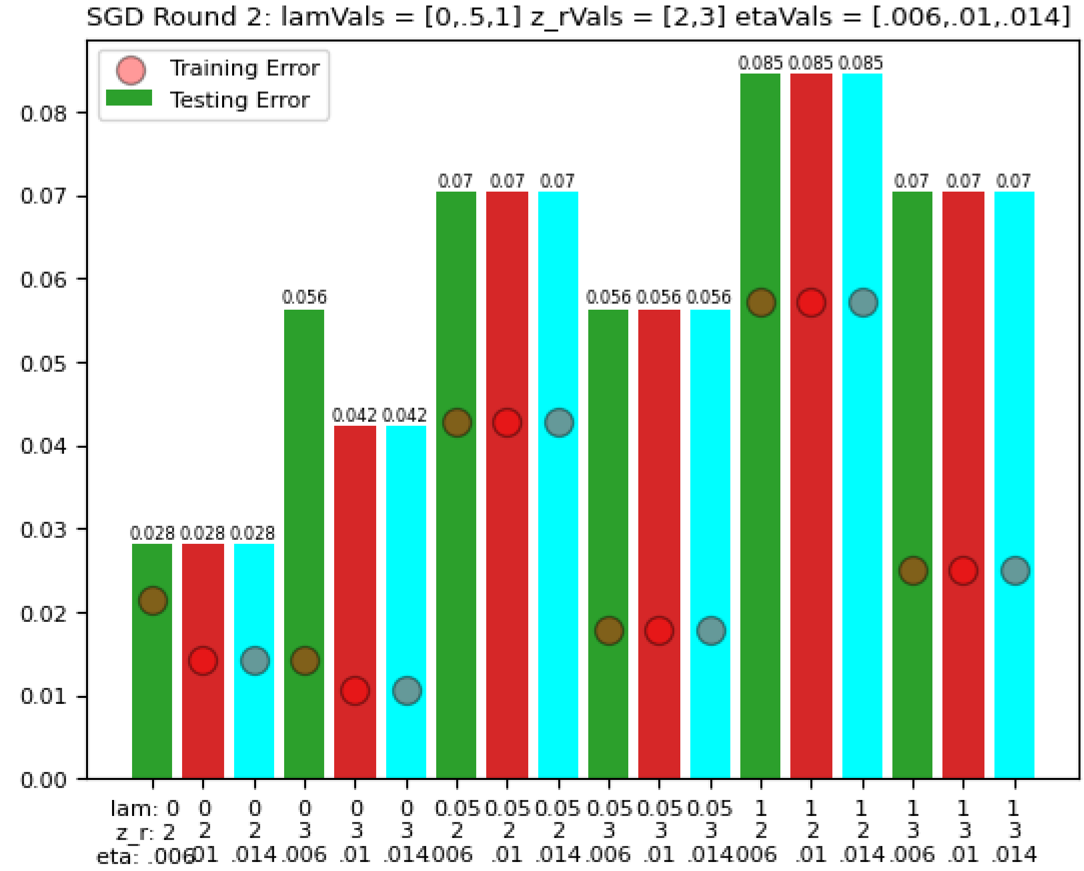


Ran a similar round 2 test for SGD:

Similar results. Time to run was 33 seconds, roughly 20% of the time for GD!

X_labels should be 0.5! Not 0.05.

Can't figure out why the two runs are overlapping between the end of lambda 2 and beginning of 4 - sorry for that!



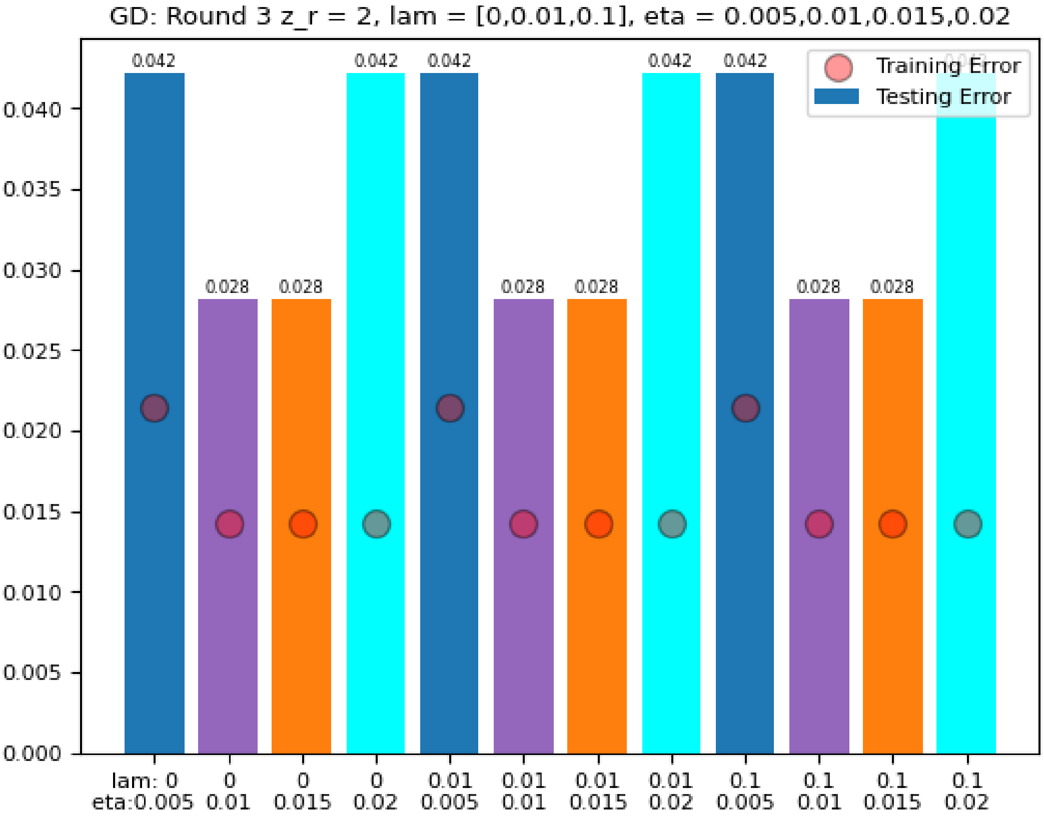
ROUND 2 TAKEAWAYS:

Takeaways:

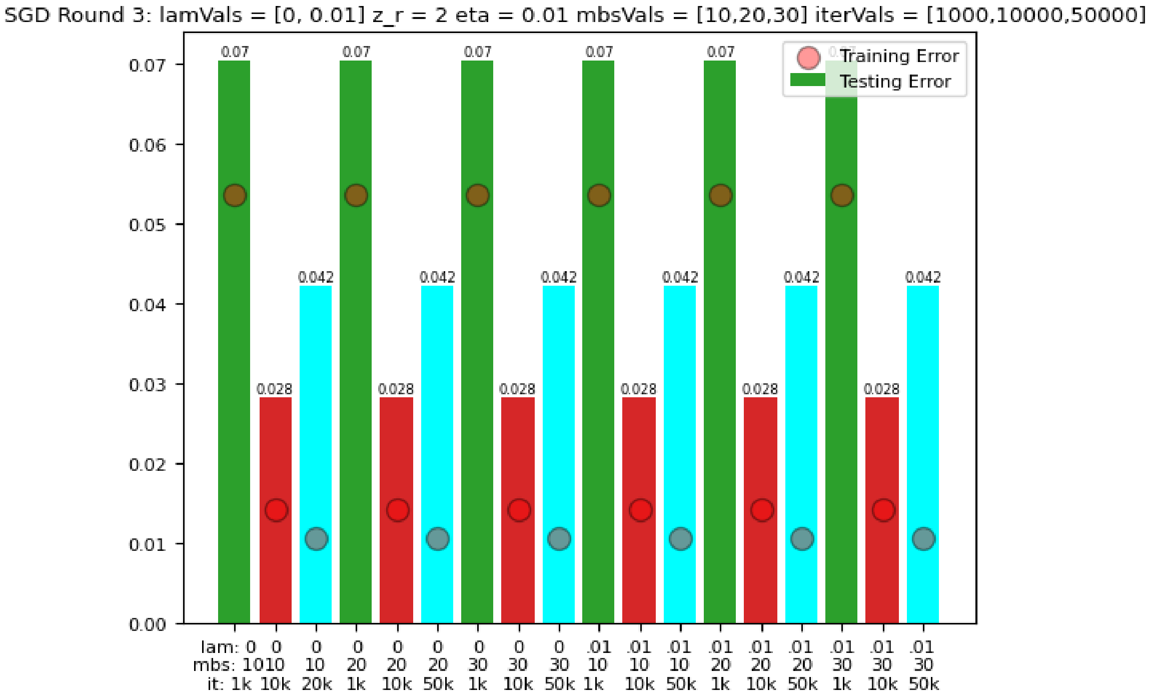
- Z_r of 2 seems optimal (I need to test lambda of 10 on it to be sure).
- Higher lambda consistently produces higher errors.
- The low on both GD and SGD was 2.8%. High was 8.5% for SGD, 7% for GD. We're not getting any feedback on eta…
- Need to test minibatch size and iterations for SGD.

ROUND 3

GD
Lambda making no difference. Eta of 0.01 produced identical error to 0.015. Can't break 2.8% error barrier.



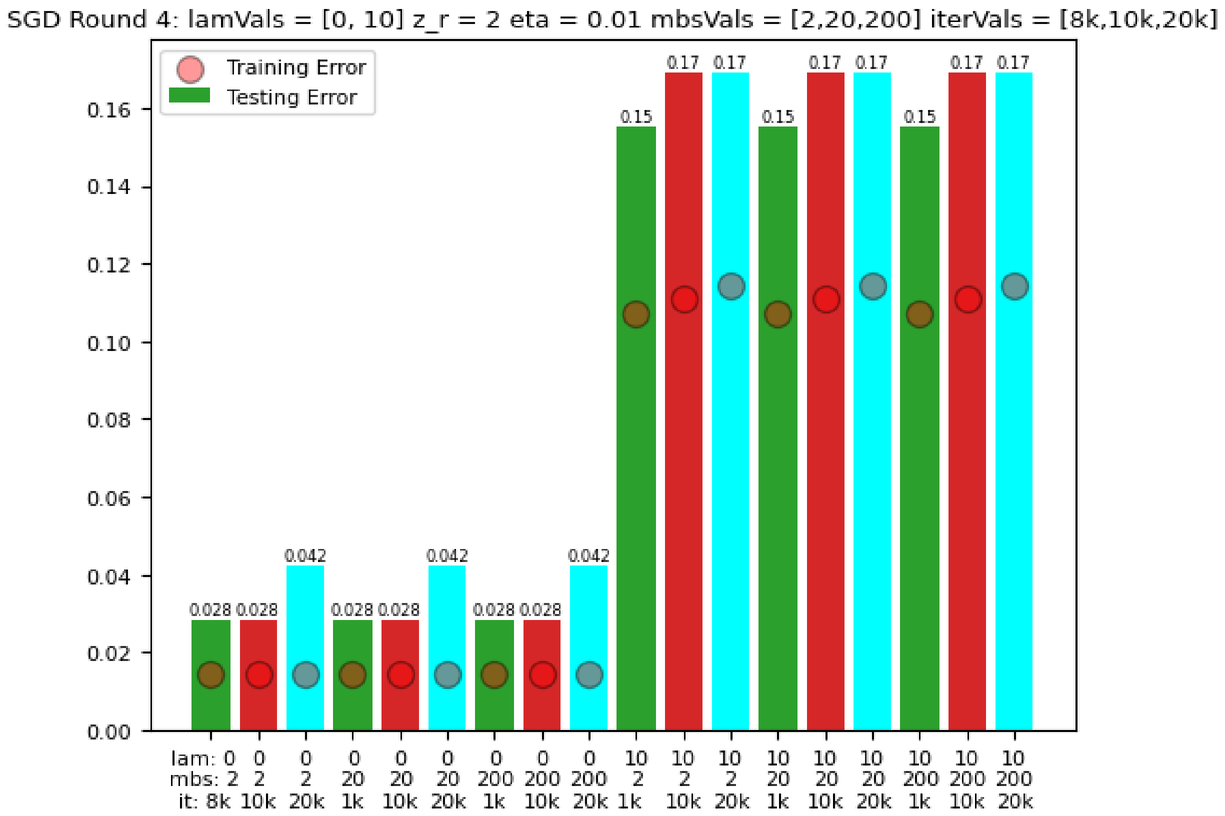
SGD ROUND 3:



Takeaways:

- Clear under/overfitting regarding iterations, which 10k being better than 1k and a little better than 50k (overfitting seems to usually have a slower rate of change judging by the slides, so 50k not necessarily ‘less far off’).
- No change regarding MBS.

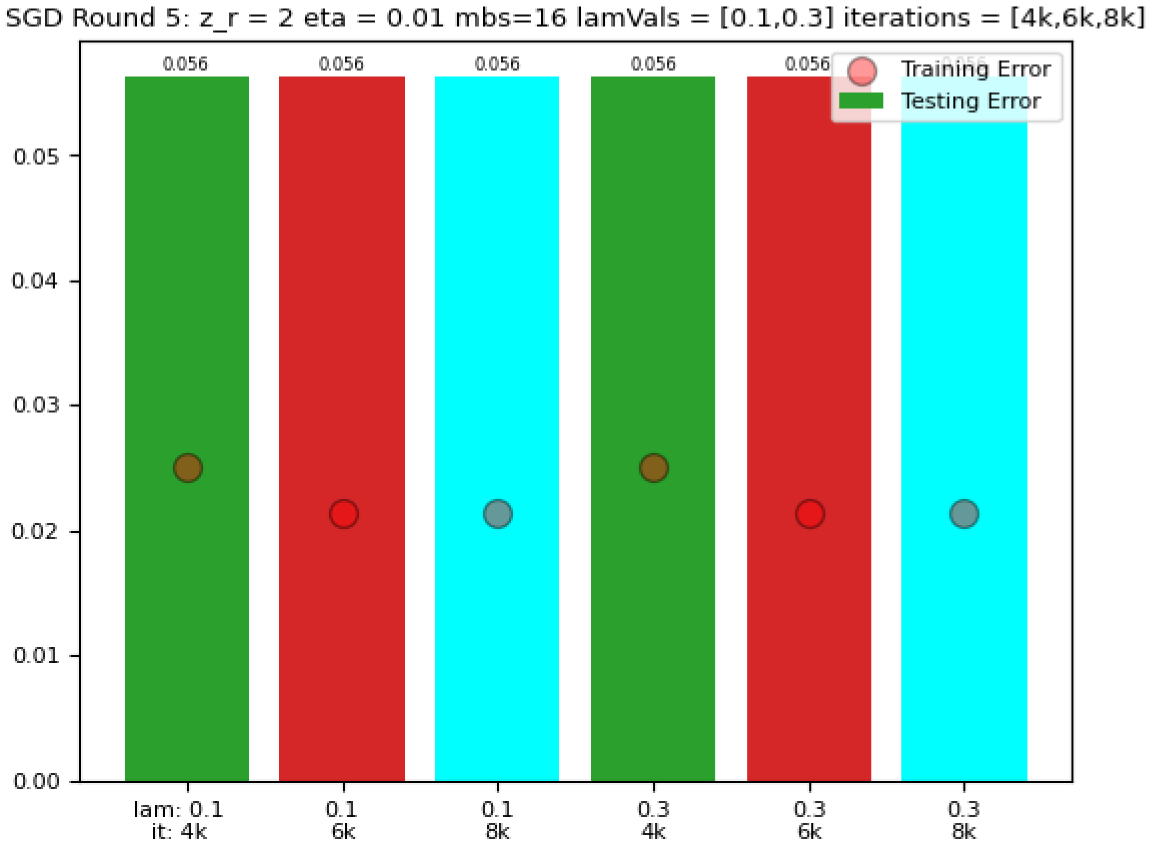
SGD ROUND 4:



Takaways:

- No difference between 8k-10k, but we know that 1k is worse. So it's possible there's a 1k < more optimal number of iterations < 8k, but we're likely in the ballpark there.
- Lambda of 0 means less punishment for a step, so it makes some sense that it's always better. 10 is obviously overfit, but we did see a more reasonable uptick between 0.5 and 1, BUT we NO difference between 0 and 0.1 (I double checked with float values).

SGD ROUND 5:



Literally no distinguishable difference between any of these.. But we've beaten this with a lambda > 0 in Round 3…

Summary:

SGD is undeniably more performant, with times as consistently less than 30 seconds, even into degree space 3, while GD ranged from 22 seconds to 3 minutes for tests. The tests I did weren't perfectly controlled for comparing speed, but similar enough to tell that SGD was more performant.

The lowest error seemed to reach a lower bound of 2.8169014%. I tried researching this number as well as lower bounds in logistic regression but didn't find anything that shed any light on this. The lower bound was always achieved with a lambda between [0, 0.5).

Results seemed to clump together. For example, 1000 is clearly underfit for iterations and 50,000 is clearly overfit. But the results for 4k, 6k, 8k, and 10k were all the same (and optimal from my test results).

Eta often made no noticable difference, but 0.01 seemed to be reliable for optimal results. Mini_Batch_Size affected performance but not misclassification rates.