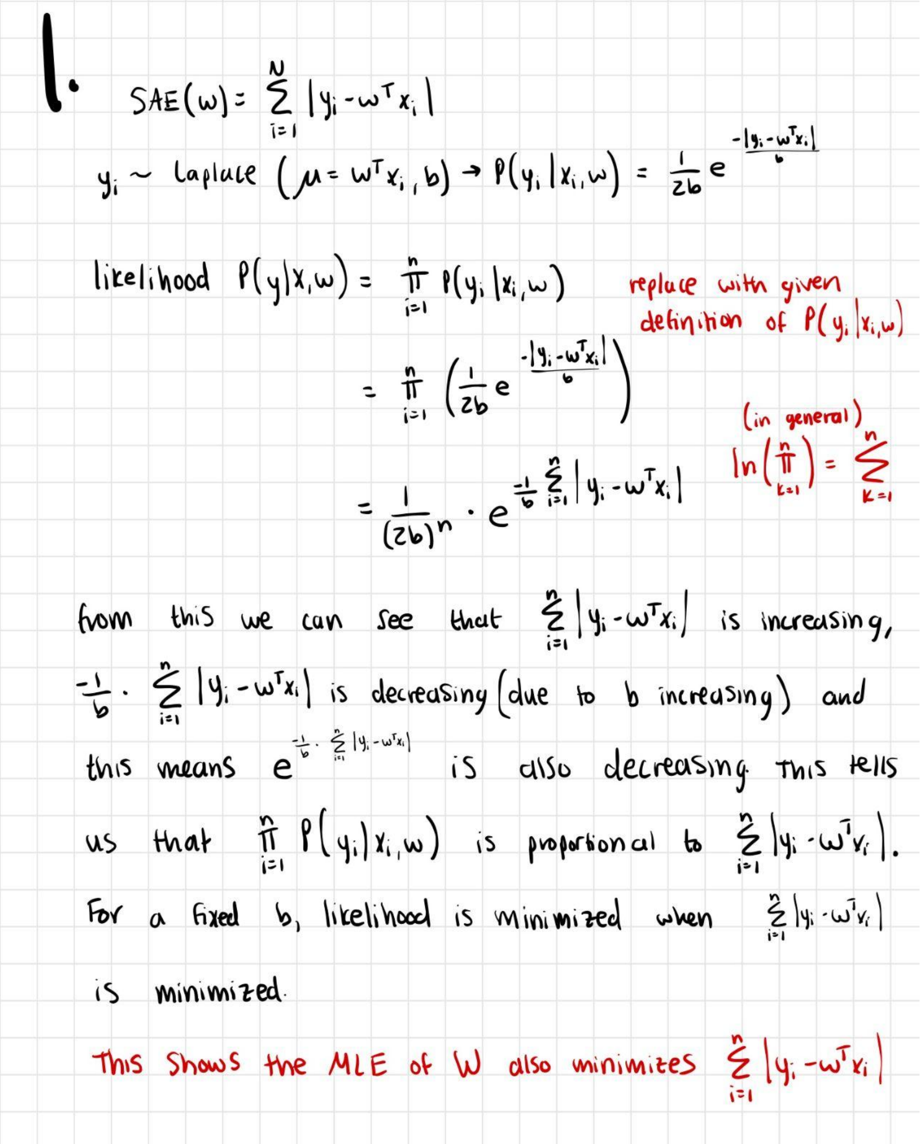
Kevin Pfeil
CS434
Dr. Lee
26 October 2021

Homework #2

Part 1

1.



1. $SAE(w) = \sum_{i=1}^{N} |y_i - w^T x_i|$

$y_i \sim \text{Laplace}\,(\mu = w^T x_i, b) \rightarrow P(y_i | x_i, w) = \frac{1}{2b} e^{\frac{-|y_i - w^T x_i|}{b}}$

likelihood $P(y|X,w) = \prod_{i=1}^{n} P(y_i | x_i, w)$ replace with given definition of $P(y_i|x_i,w)$

$$= \prod_{i=1}^{n} \left(\frac{1}{2b} e^{\frac{-|y_i - w^T x_i|}{b}} \right)$$

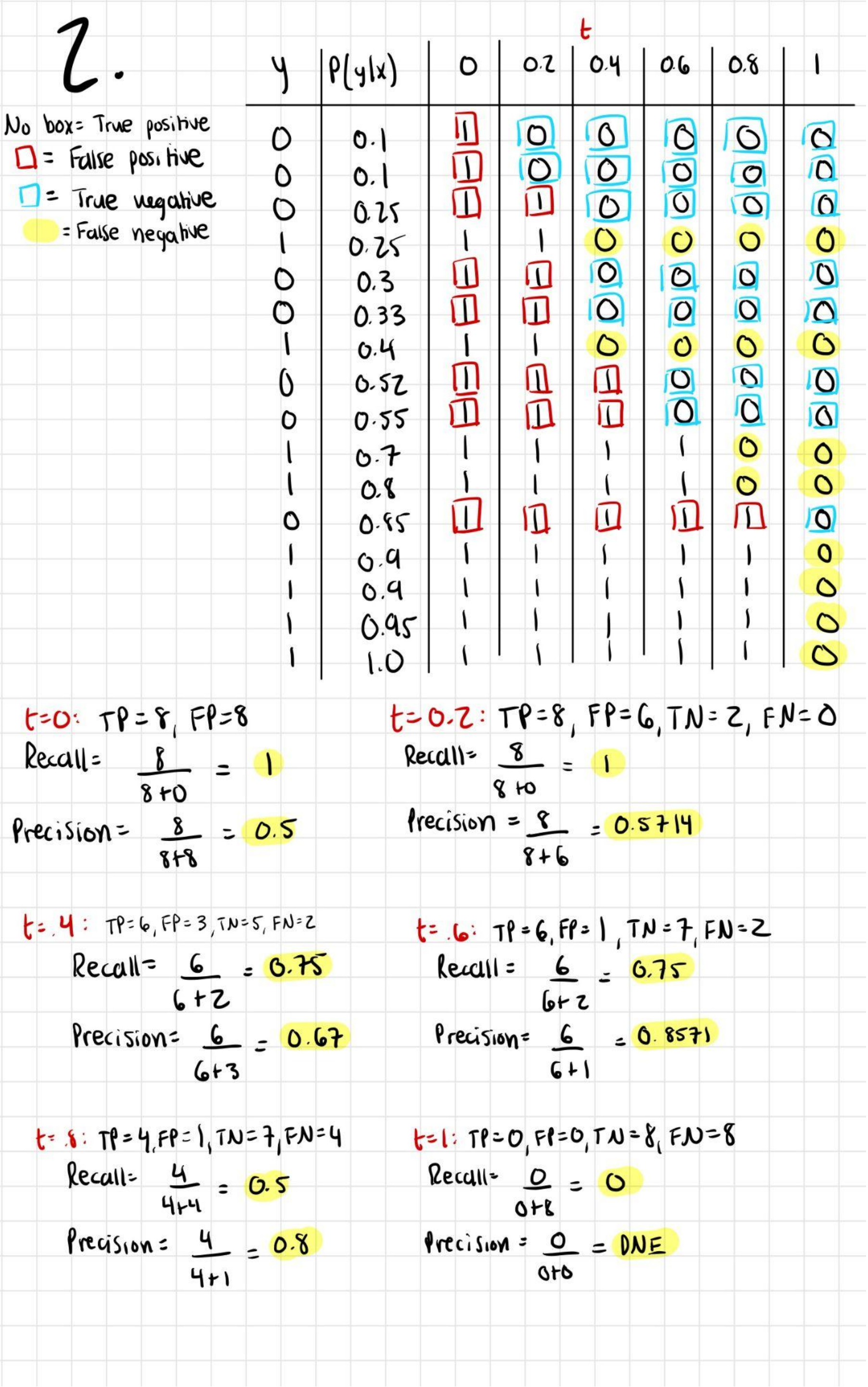
(in general) $\ln\left(\prod_{k=1}^{n}\right) = \sum_{k=1}^{n}$

$$= \frac{1}{(2b)^n} \cdot e^{\frac{-1}{b} \sum_{i=1}^{n} |y_i - w^T x_i|}$$

from this we can see that $\sum_{i=1}^{n} |y_i - w^T x_i|$ is increasing, $\frac{-1}{b} \cdot \sum_{i=1}^{n} |y_i - w^T x_i|$ is decreasing (due to b increasing) and this means $e^{\frac{-1}{b} \cdot \sum_{i=1}^{n} |y_i - w^T x_i|}$ is also decreasing. This tells us that $\prod_{i=1}^{n} P(y_i | x_i, w)$ is proportional to $\sum_{i=1}^{n} |y_i - w^T x_i|$. For a fixed b, likelihood is minimized when $\sum_{i=1}^{n} |y_i - w^T x_i|$ is minimized.

This shows the MLE of W also minimizes $\sum_{i=1}^{n} |y_i - w^T x_i|$

2.



2.

No box = True positive
🟥 = False positive
🟦 = True negative
🟨 = False negative

				t			
y	P(y|x)	0	0.2	0.4	0.6	0.8	1
0	0.1	🟥1	🟦0	🟦0	🟦0	🟦0	🟦0
0	0.1	🟥1	🟦0	🟦0	🟦0	🟦0	🟦0
0	0.25	🟥1	🟥1	🟦0	🟦0	🟦0	🟦0
1	0.25	1	1	🟨0	🟨0	🟨0	🟨0
0	0.3	🟥1	🟥1	🟦0	🟦0	🟦0	🟦0
0	0.33	🟥1	🟥1	🟦0	🟦0	🟦0	🟦0
1	0.4	1	1	🟨0	🟨0	🟨0	🟨0
0	0.52	🟥1	🟥1	🟥1	🟦0	🟦0	🟦0
0	0.55	🟥1	🟥1	🟥1	🟦0	🟦0	🟦0
1	0.7	1	1	1	1	🟨0	🟨0
1	0.8	1	1	1	1	🟨0	🟨0
0	0.85	🟥1	🟥1	🟥1	🟥1	🟥1	🟦0
1	0.9	1	1	1	1	1	🟨0
1	0.9	1	1	1	1	1	🟨0
1	0.95	1	1	1	1	1	🟨0
1	1.0	1	1	1	1	1	🟨0

t=0: TP=8, FP=8

Recall = $\frac{8}{8+0}$ = 1

Precision = $\frac{8}{8+8}$ = 0.5

t=0.2: TP=8, FP=6, TN=2, FN=0

Recall = $\frac{8}{8+0}$ = 1

Precision = $\frac{8}{8+6}$ = 0.5714

t=.4: TP=6, FP=3, TN=5, FN=2

Recall = $\frac{6}{6+2}$ = 0.75

Precision = $\frac{6}{6+3}$ = 0.67

t=.6: TP=6, FP=1, TN=7, FN=2

Recall = $\frac{6}{6+2}$ = 0.75

Precision = $\frac{6}{6+1}$ = 0.8571

t=.8: TP=4, FP=1, TN=7, FN=4

Recall = $\frac{4}{4+4}$ = 0.5

Precision = $\frac{4}{4+1}$ = 0.8

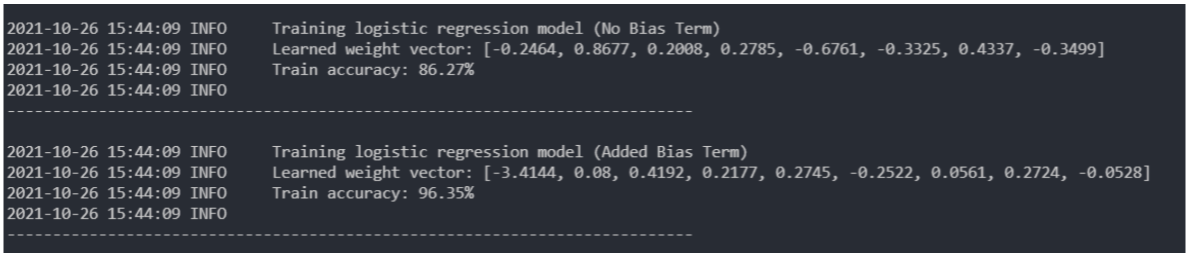
t=1: TP=0, FP=0, TN=8, FN=8

Recall = $\frac{0}{0+8}$ = 0

Precision = $\frac{0}{0+0}$ = DNE

Part 2

3. Code attached
4. Code attached
5.



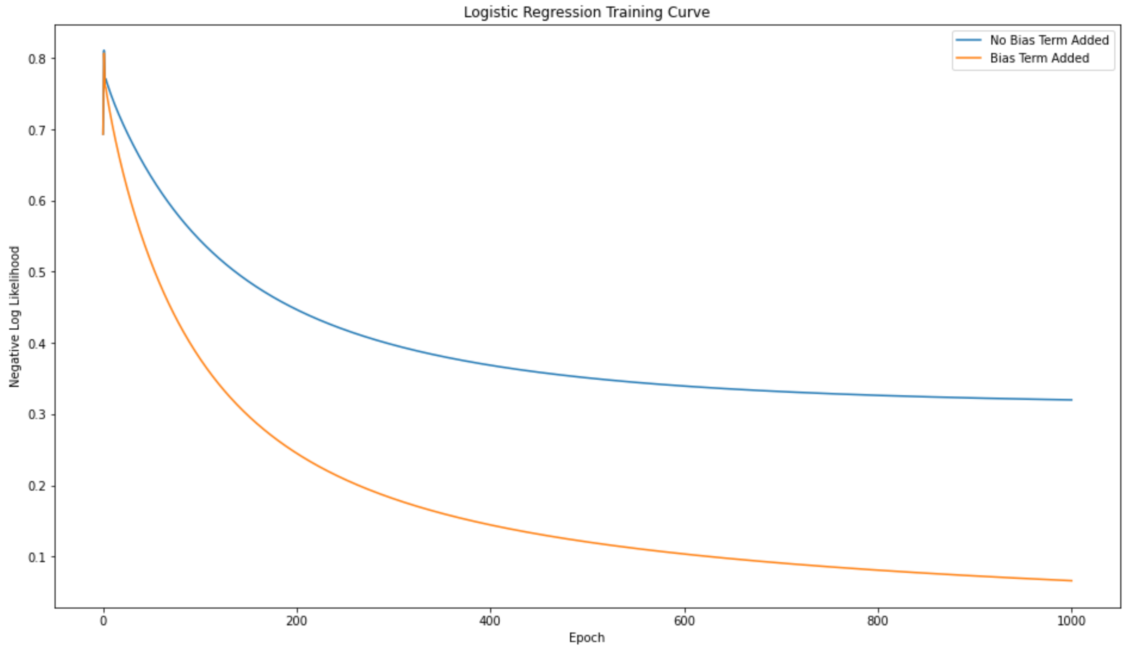
```
2021-10-26 15:44:09 INFO     Training logistic regression model (No Bias Term)
2021-10-26 15:44:09 INFO     Learned weight vector: [-0.2464, 0.8677, 0.2008, 0.2785, -0.6761, -0.3325, 0.4337, -0.3499]
2021-10-26 15:44:09 INFO     Train accuracy: 86.27%
2021-10-26 15:44:09 INFO
---------------------------------------------------------------------------

2021-10-26 15:44:09 INFO     Training logistic regression model (Added Bias Term)
2021-10-26 15:44:09 INFO     Learned weight vector: [-3.4144, 0.08, 0.4192, 0.2177, 0.2745, -0.2522, 0.0561, 0.2724, -0.0528]
2021-10-26 15:44:09 INFO     Train accuracy: 96.35%
2021-10-26 15:44:09 INFO
---------------------------------------------------------------------------
```

Yes, this is a meaningful difference in my opinion. The accuracy went up a little over 10% from the addition of the bias term.

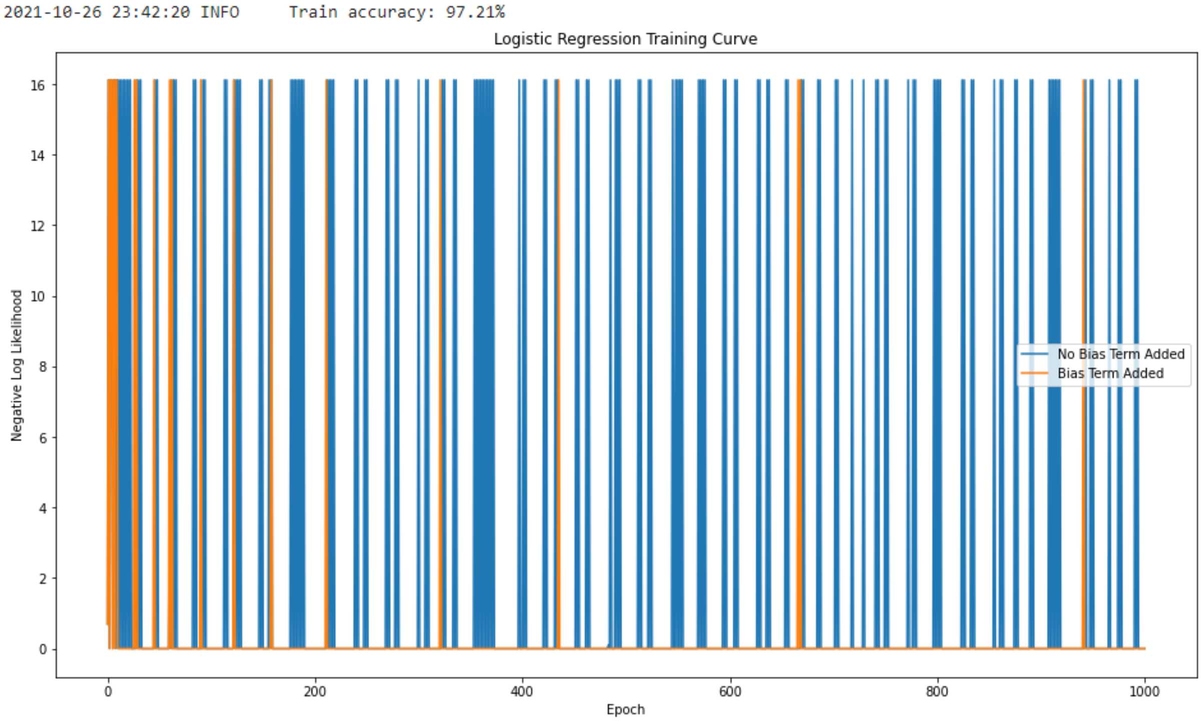
6. Based on the quick flattening of the curve that appears using the hyperparameters given, I would say that the gradient descent algorithm has nearly converged. I would hypothesize that a larger `max_iters` would make the negative log-likelihood fully flatten out and converge.

Step size = 0.0001 (base):

```
2021-10-26 23:55:28 INFO     Train accuracy: 96.35%
```

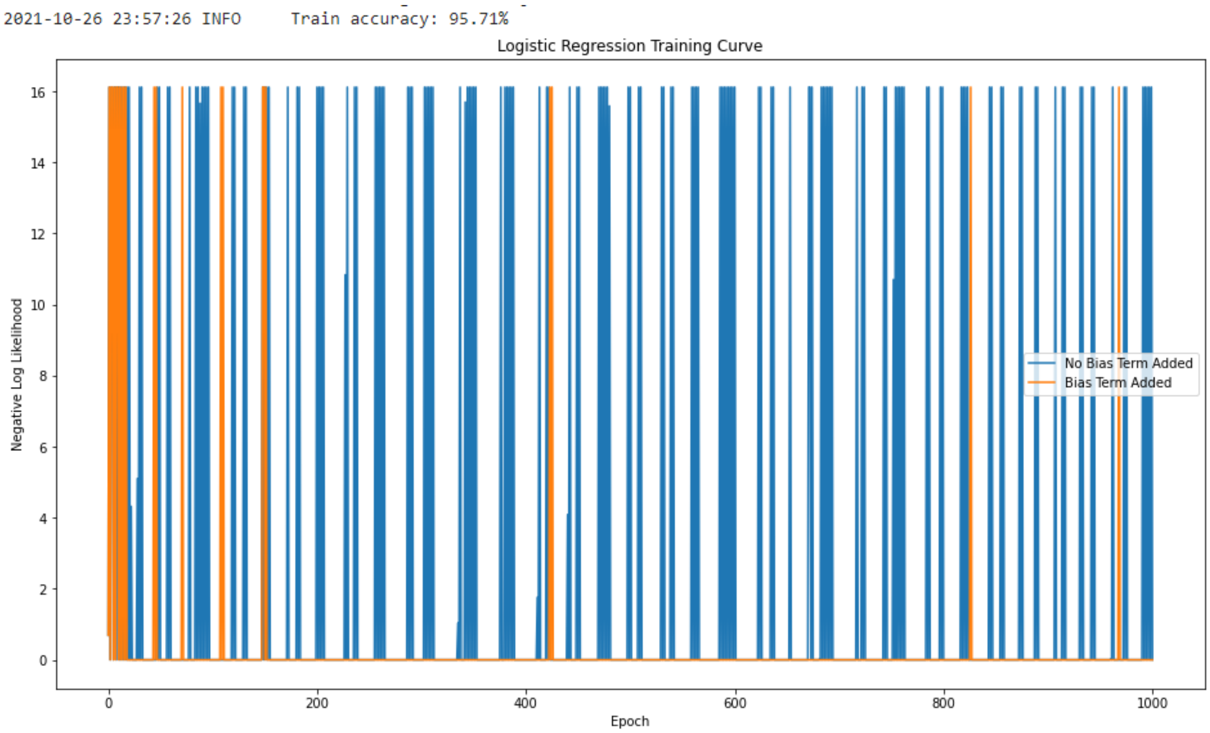


Step size = 1:



```
2021-10-26 23:42:20 INFO     Train accuracy: 97.21%
```

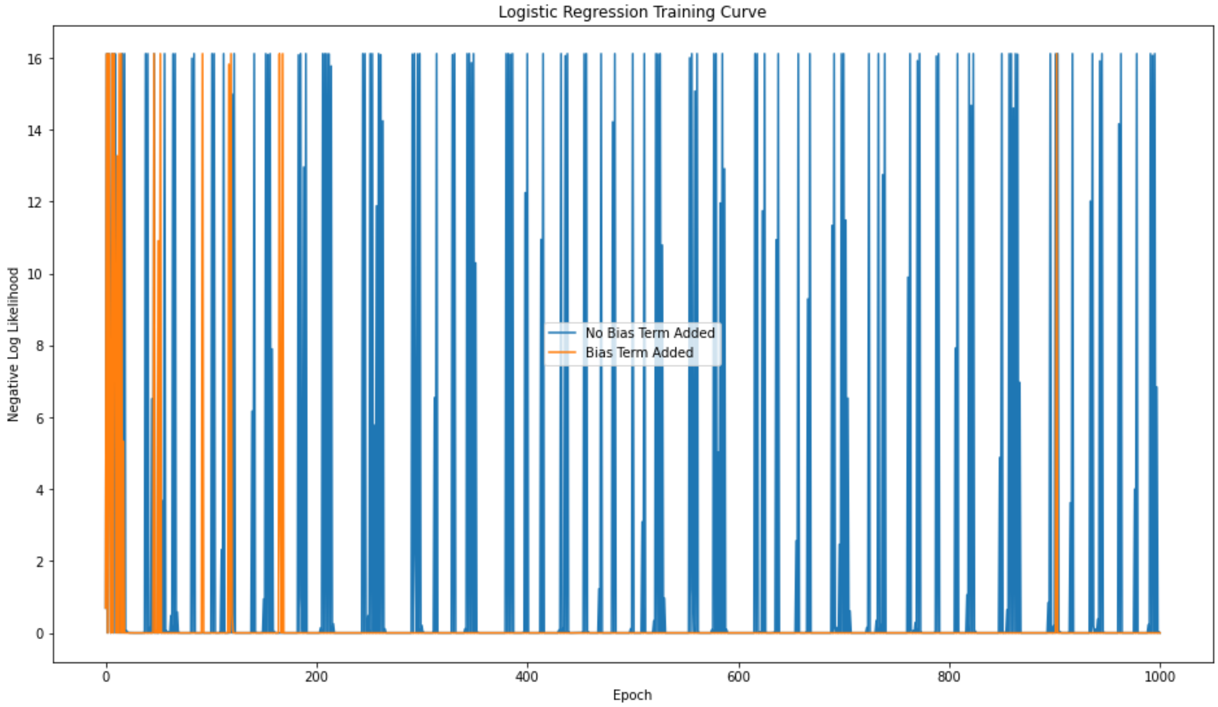
Step size = 0.1:



```
2021-10-26 23:57:26 INFO     Train accuracy: 95.71%
```

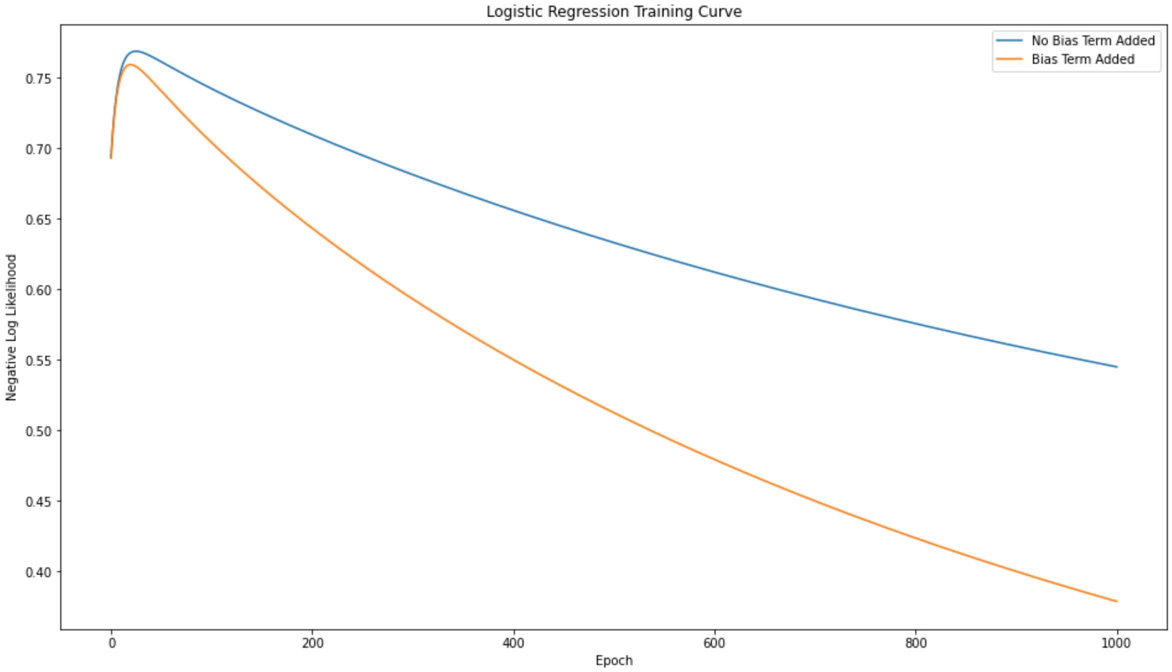
Step size = 0.01:

```
2021-10-27 00:00:11 INFO     Train accuracy: 96.35%
```



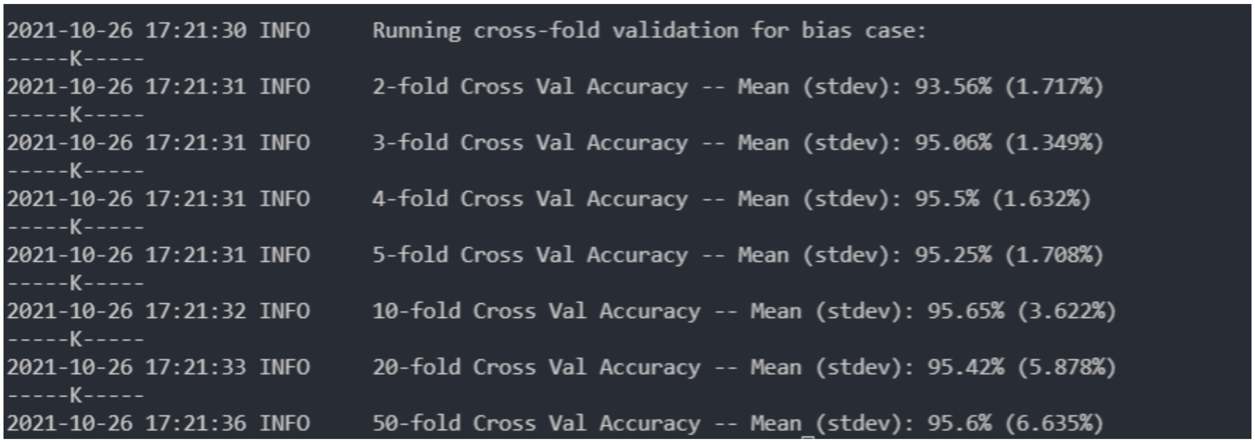
Step size = 0.00001:

```
2021-10-27 00:10:38 INFO     Train accuracy: 90.77%
```



At larger step sizes, the negative log-likelihood more sporadically goes up to infinity and back down to 0 as the epoch increases. As the step size gets smaller, this behavior is still seen but much less frequently. When the step size is as small as the given value for the hyperparameter (0.0001) and the smallest of all of the steps tested (0.00001) the negative log-likelihood does not find its way to infinity and is much more controlled. The negative log-likelihood peaks at about 0.8 for both of these values and then finds its way closer to 0 as the epoch gets larger.

7.



```
2021-10-26 17:21:30 INFO     Running cross-fold validation for bias case:
-----K-----
2021-10-26 17:21:31 INFO     2-fold Cross Val Accuracy -- Mean (stdev): 93.56% (1.717%)
-----K-----
2021-10-26 17:21:31 INFO     3-fold Cross Val Accuracy -- Mean (stdev): 95.06% (1.349%)
-----K-----
2021-10-26 17:21:31 INFO     4-fold Cross Val Accuracy -- Mean (stdev): 95.5% (1.632%)
-----K-----
2021-10-26 17:21:31 INFO     5-fold Cross Val Accuracy -- Mean (stdev): 95.25% (1.708%)
-----K-----
2021-10-26 17:21:32 INFO     10-fold Cross Val Accuracy -- Mean (stdev): 95.65% (3.622%)
-----K-----
2021-10-26 17:21:33 INFO     20-fold Cross Val Accuracy -- Mean (stdev): 95.42% (5.878%)
-----K-----
2021-10-26 17:21:36 INFO     50-fold Cross Val Accuracy -- Mean (stdev): 95.6% (6.635%)
```

These are the means and standard deviations that the various ks output during the cross validation. The mean steadily increases (for the most part) as k gets larger and the standard deviation generally increases as k does. The standard deviation tends to increase more rapidly with a higher k, whereas the mean did the majority of its increasing between k = 2 and k = 3. My value on the leaderboard was closest to the mean when k = 2, but none of these values are that far off.

Part 3

1. I spent about 5 hours on this assignment.
2. Compared to the previous assignment, I would rate it easy. I would not outright call this assignment “easy”, though. Definitely still took lots of thinking and problem solving.
3. I worked on this assignment mostly alone as I felt it was lots of implementing equations rather than working with tough concepts like the previous assignment.
4. I feel that I understand about 95% of the assignment deeply. The only part that I could not grasp was how to improve my score on the kaggle leaderboard by doing things other than just changing `step_size`.
5. Fun assignment!