# CSE512 Spring 2021 - Machine Learning - Homework 2

Name: Sabrina Margetic

Solar ID: 109898930

Net ID Email Address: sabrina.margetic@stonybrook.edu

Names of people whom you discussed the homework with: None (Only the TA)

# 1 Question 1:

## 1.1 MLE

### 1.1.1

Given:
$P(X=k|\lambda)=\frac{\lambda^k}{k!}e^{-\lambda}$
The likelihood becomes:
$L(x_1,x_2...x_n|\lambda)=\Pi_{i=1}^{n}\frac{\lambda^{x_i}}{x_i!}e^{-\lambda}$
Taking the log:
$log(L(x_1,x_2...x_n|\lambda))=log(\Pi_{i=1}^{n}\frac{\lambda^{x_i}}{x_i!}e^{-\lambda})$
$=\sum_{i=1}^{n}log(\frac{\lambda^{x_i}}{x_i!}e^{-\lambda})$
$=\sum_{i=1}^{n}[x_i log(\lambda)-log(x_i!)-\lambda log(e)]$
$=\sum_{i=1}^{n}(x_i log(\lambda))-\sum_{i=1}^{n}(log(x_i!))-\sum_{i=1}^{n}(\lambda)$
$\therefore \boxed{log(L(x_1,x_2...x_n|\lambda))=log(\lambda)\sum_{i=1}^{n}x_i-\sum_{i=1}^{n}log(x_i!)-n\lambda}$

### 1.1.2

Take the derivative:
$\frac{\partial log(L(x_1,x_2...x_n|\lambda))}{\partial\lambda}=\frac{1}{\lambda}\sum_{i=1}^{n}x_i-n$
Set the derivative to 0:
$0=\frac{1}{\lambda}\sum_{i=1}^{n}x_i-n$
$n=\frac{1}{\lambda}\sum_{i=1}^{n}x_i$
$\boxed{\hat{\lambda}=\frac{\sum_{i=1}^{n}x_i}{n}}$

### 1.1.3

$\hat{\lambda}=\frac{\sum_{i=1}^{n}x_i}{n}$
Using our data:
$\hat{\lambda}=\frac{4+12+3+5+6+9+17}{7}$
$\boxed{\hat{\lambda}=8}$

## 1.2 MAP

### 1.2.1

Bayes Theorem: $P(\lambda|X)=\frac{P(X|\lambda)P(\lambda)}{\int_{-\infty}^{\infty}P(X|\lambda)P(\lambda)d\lambda}$ [1]

Poisson: $P(X|\lambda)=\Pi_{i=1}^{n}\frac{\lambda^{x_i}}{x_i!}e^{-\lambda}$
Gamma: $P(\lambda|\alpha,\beta)=\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}$

Step 1. Numerator of Bayes Theorem:
$P(X|\lambda)P(\lambda)=\Pi_{i=1}^{n}(\frac{\lambda^{x_i}}{x_i!}e^{-\lambda})(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda})$

$$=[e^{-\lambda n}\lambda^{\sum_{i=1}^{n}x_i}][\Pi_{i=1}^{n}\frac{1}{x_i!}][\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda}]$$

$$=[\Pi_{i=1}^{n}\frac{1}{x_i!}][e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}\frac{\beta^\alpha}{\Gamma(\alpha)}]$$

Step 2. Denominator of Bayes Theorem:

[1] https://en.wikipedia.org/wiki/Bayes'_theorem

$\int_{-\infty}^{\infty} P(X|\lambda)P(\lambda)d\lambda = \int_{-\infty}^{\infty}[(\Pi_{i=1}^{n}\frac{\lambda^{x_i}}{x_i!}e^{-\lambda})(\frac{\beta^\alpha}{\Gamma(\alpha)}\lambda^{\alpha-1}e^{-\beta\lambda})d\lambda]$

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})\int_{-\infty}^{\infty} e^{-n\lambda}\lambda^{\sum_{i=1}^{n}x_i}\lambda^{\alpha-1}e^{-\beta\lambda}d\lambda$

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})\int_{-\infty}^{\infty} e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}d\lambda$

Now, let $\Omega = n+\beta$ and $\psi = \sum_{i=1}^{n}x_i+\alpha$

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})\int_{-\infty}^{\infty} e^{-\lambda\Omega}\lambda^{\psi-1}d\lambda$

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})(\frac{\frac{\Omega^\psi}{\Gamma(\psi)}}{\frac{\Omega^\psi}{\Gamma(\psi)}})\int_{-\infty}^{\infty} e^{-\lambda\Omega}\lambda^{\psi-1}d\lambda$

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})(\frac{1}{\frac{\Omega^\psi}{\Gamma(\psi)}})\int_{-\infty}^{\infty} \frac{\Omega^\psi}{\Gamma(\psi)}e^{-\lambda\Omega}\lambda^{\psi-1}d\lambda$

The integral now becomes that of the pdf of Gamma($\lambda|\alpha=\psi;\beta=\Omega$), and therefore evaluates to 1.

$= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})(\frac{1}{\frac{\Omega^\psi}{\Gamma(\psi)}})$

Substituting back for $\psi$ and $\Omega$:

denom $= \frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})(\frac{1}{\frac{(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}}{\Gamma(\sum_{i=1}^{n}x_i+\alpha)}})$

Step 3. Put Steps 1 and 2 Together:

$P(\lambda|X) = \frac{[\Pi_{i=1}^{n}\frac{1}{x_i!}][e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}\frac{\beta^\alpha}{\Gamma(\alpha)}]}{\frac{\beta^\alpha}{\Gamma(\alpha)}(\Pi_{i=1}^{n}\frac{1}{x_i!})(\frac{1}{\frac{(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}}{\Gamma(\sum_{i=1}^{n}x_i+\alpha)}})}$

$= \frac{[e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}]}{(\frac{1}{\frac{(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}}{\Gamma(\sum_{i=1}^{n}x_i+\alpha)}})}$

$= \frac{e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}}{\Gamma(\sum_{i=1}^{n}x_i+\alpha)}$

$$\boxed{P(\lambda|X) = \text{Gamma }(\lambda|\alpha=(\alpha+\sum_{i=1}^{n}x_i);\beta=(n+\beta))}$$

**1.2.2**

Let $P(\lambda|X) = \frac{e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}}{\Gamma(\sum_{i=1}^{n}x_i+\alpha)}$

$log(P(\lambda|X)) = log(e^{-\lambda(n+\beta)}\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}(n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}) - log(\Gamma(\sum_{i=1}^{n}x_i+\alpha))$

$= log(e^{-\lambda(n+\beta)}) + log(\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}) + log((n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}) - log(\Gamma(\sum_{i=1}^{n}x_i+\alpha))$

$= -\lambda(n+\beta)log(e) + (\sum_{i=1}^{n}x_i+\alpha-1)log(\lambda) + log((n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}) - log(\Gamma(\sum_{i=1}^{n}x_i+\alpha))$

$= -\lambda(n+\beta) + (\sum_{i=1}^{n}x_i+\alpha-1)log(\lambda) + log((n+\beta)^{\sum_{i=1}^{n}x_i+\alpha}) - log(\Gamma(\sum_{i=1}^{n}x_i+\alpha))$

Take the derivative:

$\frac{\partial log(P(\lambda|X)}{\partial\lambda} = -(n+\beta) + \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{\lambda}$

Set the derivative to 0:

$0 = -(n+\beta) + \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{\lambda}$
$n + \beta = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{\lambda}$
$\boxed{\hat{\lambda} = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{n+\beta}}$

## 1.3 Estimator Bias

### 1.3.1

Given: $\eta = e^{-2\lambda}$
$log(\eta) = -2\lambda$
$\lambda = \frac{-log(\eta)}{2}$

Poisson with a single observation:
$P(X = x|\lambda) = \frac{\lambda^x}{x!} e^{-\lambda}$
Substituting in: $P(X = x|\lambda) = \frac{(\frac{-log(\eta)}{2})^x}{x!} e^{-\frac{-log(\eta)}{2}}$
$log(P(X|\lambda)) = log(\frac{(\frac{-log(\eta)}{2})^x}{x!}) + log(e^{-\frac{-log(\eta)}{2}})$

$$= log((\frac{-log(\eta)}{2})^x) - log(x!) + \frac{log(\eta)}{2}$$

$$= xlog(\frac{-log(\eta)}{2}) - log(x!) + \frac{log(\eta)}{2}$$

$$= xlog(-log(\eta)) - xlog(2) - log(x!) + \frac{log(\eta)}{2}$$

$\frac{\partial P(X=x|\lambda)}{\partial\eta} = \frac{x}{\eta log(\eta)} + \frac{1}{2\eta}$
$0 = \frac{2x}{2\eta log(\eta)} + \frac{log(\eta)}{2\eta log(\eta)}$
$0 = \frac{2x + log(\eta)}{2\eta log(\eta)}$
$2x + log(\eta) = 0$
$log(\eta) = -2x$
$\boxed{\hat{\eta} = e^{-2x}}$

### 1.3.2

$bias(\hat{\eta}) = E[\hat{\eta}] - \eta$
1. $\eta = e^{-2\lambda}$
2. $E[\hat{\eta}] = \sum_{all \hat{\eta}} \hat{\eta} P(\hat{\eta})$

From previously: $\hat{\eta} = e^{-2x}$
$\therefore E[\hat{\eta}] = E[e^{e^{-2x}}]$

$$= \sum_{x=0}^{\infty} e^{-2x} \frac{\lambda^x e^{-\lambda}}{x!}$$

$$= [\sum_{x=0}^{\infty} e^{-2x} \frac{\lambda^x}{x!}] e^{-\lambda}$$

$$= [\sum_{x=0}^{\infty} (e^{-2})^x \frac{\lambda^x}{x!}] e^{-\lambda}$$

$$= [\sum_{x=0}^{\infty} \frac{(e^{-2}\lambda)^x}{x!}] e^{-\lambda}$$

Through Taylor expansion the integral becomes: $e^{e^{-2}\lambda}$

$\therefore E[\hat{\eta}] = e^{e^{-2}\lambda}e^{-\lambda}$

$= e^{e^{-2}\lambda-\lambda}$

$= e^{-\lambda[-e^{-2}+1]}$

$= e^{-\lambda[1-\frac{1}{e^2}]}$

Putting steps 1. and 2. together, we obtain:

$\boxed{bias(\hat{\eta}) = e^{-\lambda[1-\frac{1}{e^2}]} - e^{-2\lambda}}$

### 1.3.3

Given: $\hat{\eta} = (-1)^x$

$bias(\hat{\eta}) = E[\hat{\eta}] - \eta$

1. $\eta = e^{-2\lambda}$

2. $E[\hat{\eta}] = E[(-1)^x]$

$= \sum_{x=0}^{\infty}(-1)^x \frac{\lambda^x e^{-\lambda}}{x!}$

$= [\sum_{x=0}^{\infty}(-1)^x \frac{\lambda^x}{x!}]e^{-\lambda}$

$= [\sum_{x=0}^{\infty} \frac{(-\lambda)^x}{x!}]e^{-\lambda}$

Through Taylor expansion, this summation becomes: $e^{-\lambda}$

$\therefore E[\hat{\eta}] = e^{-\lambda}e^{-\lambda}$

$= e^{-2\lambda}$

Putting steps 1. and 2. together:

$bias(\hat{\eta}) = e^{-2\lambda} - e^{-2\lambda} = 0$

$\boxed{\therefore unbiased}$

The reason this unbiased estimator is not good is because it doesn't resemble what we would expect given a data point x. For example, if x were 100, we would expect the true value to be near 0. Our unbiased estimator, however, predicts 1.

When we view the case for when $x = 101$, we have that our unbiased estimator is -1. This case is simply wrong since our true value is unable to be negative. The value should rather be once again close to 0.

# 2 Question 2:

## 2.1

Given: $P(Y^i|\bar{X^i};\boldsymbol{\theta}) = \frac{exp(\boldsymbol{\theta}_{Y^i}^T\bar{X^i})}{1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T\bar{X^i})}$

<u>Step 1. Take log:</u>

$log(P(Y^i|\bar{X^i};\boldsymbol{\theta})) = log(exp(\boldsymbol{\theta}_{Y^i}^T\bar{X^i})) - log(1 + \sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T\bar{X^i}))$

$= \boldsymbol{\theta}_{Y^i}^T\bar{X^i} - log(1 + \sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T\bar{X^i}))$

<u>Step 2. Take derivative of log:</u>

$\frac{\partial log(P(Y^i|\bar{X^i};\boldsymbol{\theta}))}{\partial\boldsymbol{\theta}_c} = \frac{\partial(\boldsymbol{\theta}_{Y^i}^T\bar{X^i})}{\partial\boldsymbol{\theta}_c} - \frac{\partial log(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T\bar{X^i}))}{\partial\boldsymbol{\theta}_c}$

Step 2.a Looking at $\frac{\partial(\boldsymbol{\theta}_{Y^i}^T \bar{X}^i)}{\partial \boldsymbol{\theta}_c}$:

There are two different scenarios here: 1. when $c = Y_i$ and 2. when $c \neq Y_i$.

1. The derivative when $c = Y_i$: $\frac{\partial(\boldsymbol{\theta}_c^T \bar{X}^i)}{\partial \boldsymbol{\theta}_c}$ is $\bar{X}^i$.
2. The derivative when $c \neq Y_i$: $\frac{\partial(\boldsymbol{\theta}_{Y^i \neq c}^T \bar{X}^i)}{\partial \boldsymbol{\theta}_c}$ is 0, since it behaves as a constant because it does not rely on the parameter we're taking the derivative in respect to.

The combination of these two events can be represented as: $\delta(c = Y_i)\bar{X}^i$
where $\delta(c = Y^i)$ evaluates to 1 when $c = Y_i$ and 0 when $c \neq Y_i$

Step 2.b Looking at $\frac{\partial log(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c}$:

$\frac{\partial log(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c} = \frac{1}{1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i)} \frac{\partial(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c}$

To understand $(1 + \sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))$ better, we can expand it as: $1 + exp(\boldsymbol{\theta}_1^T \bar{X}^i) + exp(\boldsymbol{\theta}_2^T \bar{X}^i) + ... + exp(\boldsymbol{\theta}_{k-1}^T \bar{X}^i)$
From this expansion, we can see that unless $\boldsymbol{\theta}_j = \boldsymbol{\theta}_c$, the values behave as constants since they don't depend on what we are taking the derivative in respect to.
Therefore, the derivative $\frac{\partial(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c} = \frac{\partial(exp(\boldsymbol{\theta}_c^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c}$
$= exp(\boldsymbol{\theta}_c^T \bar{X}^i)\bar{X}^i$

$\therefore \frac{\partial log(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c} = \frac{1}{1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i)} exp(\boldsymbol{\theta}_c^T \bar{X}^i)\bar{X}^i$
Since $P(c|\bar{X}^i;\boldsymbol{\theta}) = \frac{exp(\boldsymbol{\theta}_c^T \bar{X}^i)}{1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i)}$:
$\frac{\partial log(1+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X}^i))}{\partial \boldsymbol{\theta}_c} = P(c|\bar{X}^i;\boldsymbol{\theta})\bar{X}^i$

Step 3. Putting Steps 2a and 2b Together:

$\frac{\partial log(P(Y^i|\bar{X}^i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_c} = \delta(c = Y_i)\bar{X}^i - P(c|\bar{X}^i;\boldsymbol{\theta})\bar{X}^i$

$$\boxed{\frac{\partial log(P(Y^i|\bar{X}^i;\boldsymbol{\theta}))}{\partial \boldsymbol{\theta}_c} = (\delta(c = Y_i) - P(c|\bar{X}^i;\boldsymbol{\theta}))\bar{X}^i}$$

### 2.2

The code for this part of the assignment can be found in hw2.py.
The following alteration was made for the probability:

$P(Y = i|X;\boldsymbol{\theta}) = \frac{exp(\boldsymbol{\theta}_i^T \bar{X} - a)}{exp(-a)+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X} - a)}$
$P(Y = 0|X;\boldsymbol{\theta}) = \frac{exp(-a)}{exp(-a)+\sum_{j=1}^{k-1} exp(\boldsymbol{\theta}_j^T \bar{X} - a)}$

Where a = $\max(\boldsymbol{\theta}_1^T \bar{X}, \boldsymbol{\theta}_2^T \bar{X}, ...\boldsymbol{\theta}_{k-1}^T \bar{X})$

This alteration was done as to avoid overflow problems. However, they still occurred.

### 2.2.1

### 2.2.2

### 2.2.3

### 2.3

The code to answer these specific questions can be found in hw2.py.

### 2.3.1

To run the code that generates these specific values, write the command, "python -c 'import hw2; hw2.normalize_train()'" into the terminal. a.
In Fig 1 we see that the loss function converges quickly, only after 4 epochs, to a steady value.

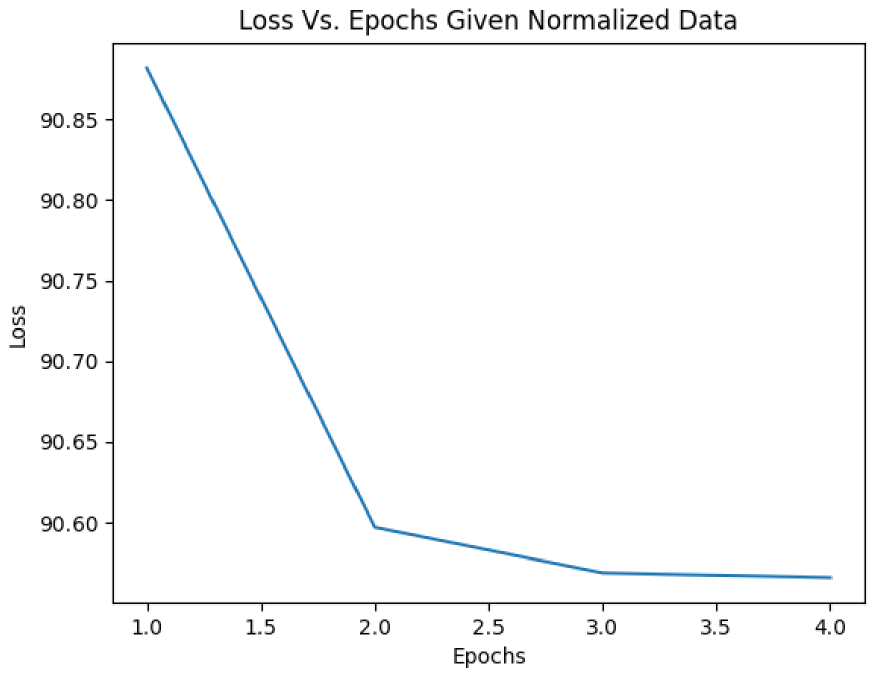


Figure 1

b.
1. Accuracy:
Using Training Data: 60.27496382054993%
Using Test Data: 72.06946454413892%
2. Confusion Matrix:
We are able to view the confusion matrices produced in Fig. 2, and Fig. 3. From these matrices we are able to see that for the training data, the diagonal represents the majority. The classification of 1 labeled as 1 is at .84, while the classification for 0 labeled as 0 is at .58, barely beating it's predicted 1 label, at .42.
On the other hand, using test data, we now see that the classification of true data 0 as 0 leads. We also observe that true values of 1 were classified more often as 0 than they were as 1. It is possible that the algorithm over fit this 1 classification or that more data for true values of 1 could be needed for training.

3. Accuracy Based on the Diagonal of The Confusion Matrix:
Using Training Data: 71.15098895366936%
Using Test Data: 57.93728908886389%

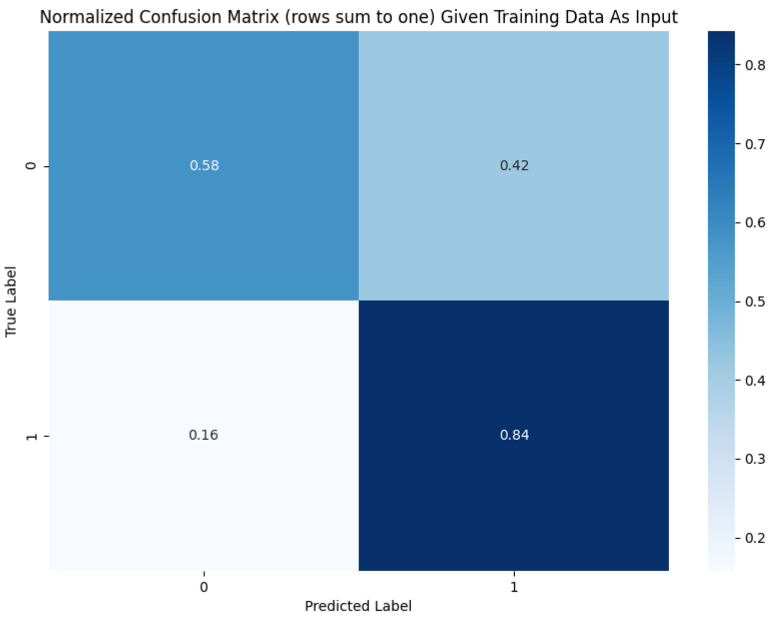


Figure 2: Training Data (0 is survival and 1 is death)

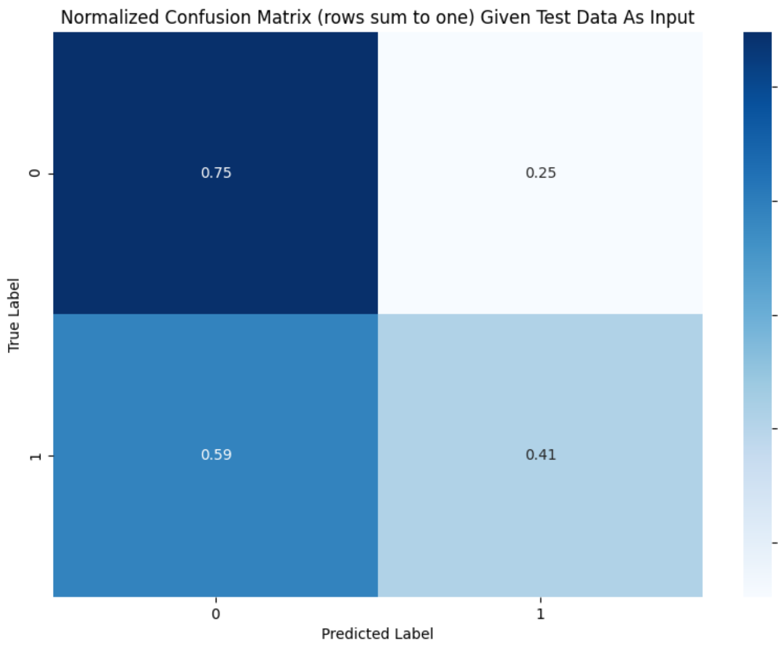


Figure 3: Test Data (0 is survival and 1 is death)

c.
1. Average Precision: 0.15614612031488634
2. Precision Recall Curve:
The precision recall curve produced can be seen in Fig. 4.

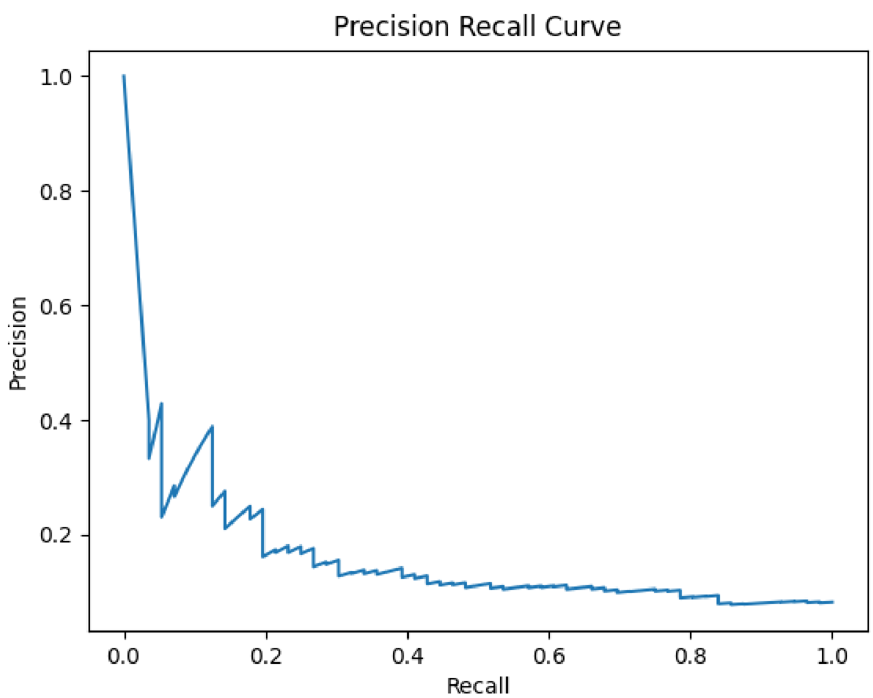


Figure 4

### 2.3.2

a.
The resulting loss function is what's shown in Fig. 5 . As you can see, there is nothing on this graph. Even though I had put in place metrics in order to prevent overflowing, the exp(-a) part quickly reached infinity and overflowed the problem. The number of epochs used was the max number of epochs since the relationship: $L(\theta^{old}) - L(\theta) < \epsilon L(\theta^{old})$ was never satisfied (the loss was always inf). The problem was able to reach some sort of convergence, but it took a significantly longer time to reach (max epoch). Overall, the performance of this algorithm without normalization is much worse and should be avoided.

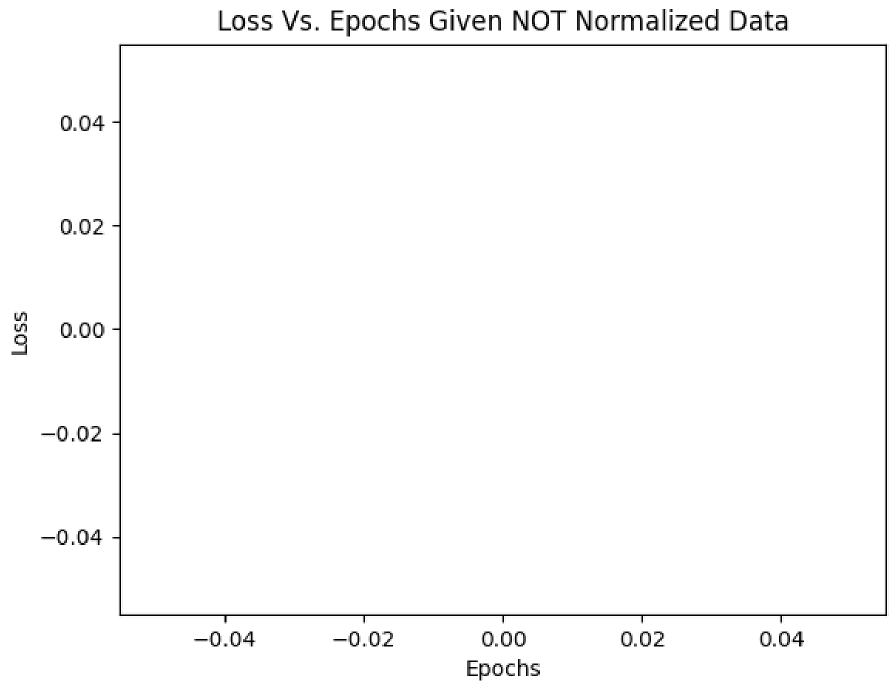


Figure 5

b. The function that used to make this data is called: normalized_train_dif_param()
1. Baseline Data: Figure 6

2. Eta Start:

Small: Fig. 7 with eta start = .00001
Accuracy: 68.01736613603472%
Large: Fig. 8 with eta start = 10
Accuracy: 77.85817655571635%
From this, we can see an improvement in accuracy when there is a larger eta start. We also see that with small enough eta start, the loss function doesn't work properly.

3. Eta End:

Small: Fig. 9 with eta end = .00000001
Accuracy: 75.39797395079594%
Large: Fig. 10 with eta end = .01

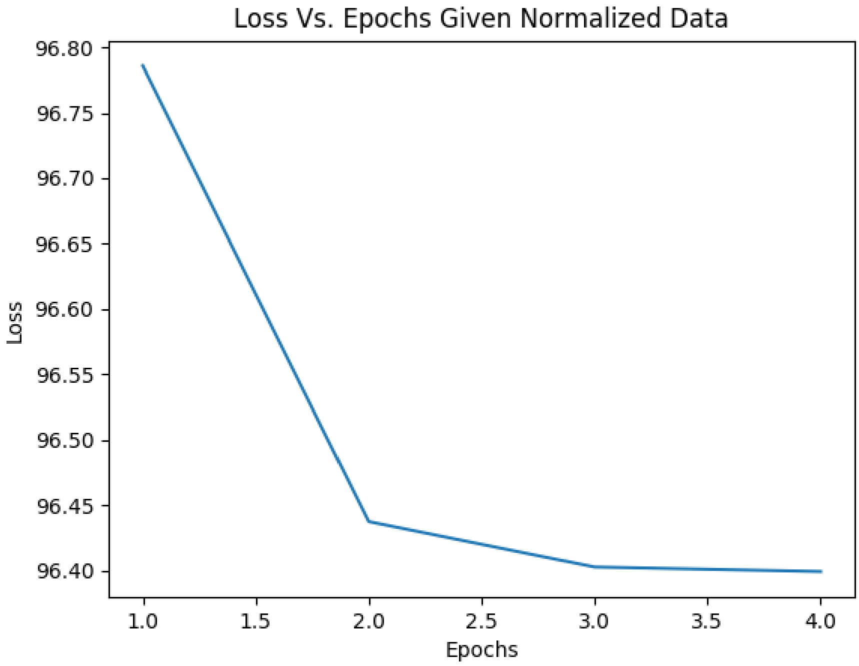


Figure 6: Baseline

Accuracy: 72.21418234442837%

We see a decrease in accuracy as eta end increases. We once again see that the loss function doesn't work properly in this region.

4. Max Epoch:

Small: Fig. 11 with max epoch = 100
Accuracy: 69.46454413892909%
Large: Fig. 12 with max epoch = 10000
Accuracy: 71.34587554269174%
We see that as the max epoch's increases, the accuracy increase. There is not any significant difference in the loss versus epoch curve.

5 M:

Small: Fig. 13 with m = 100
Accuracy: 70.04341534008684%
Large: Fig. 14 with m = 500
Accuracy: 73.51664254703329%
We see that as m increases, the accuracy increase. There is once again no significant difference in the loss versus accuracy curve.

c. The function used to generate this answer is called: my_own_values()
I use the parameters: m= 500, eta start= 10, eta end= .00000001, and max epoch= 10000. I chose these parameters because from the previous part they seemed to indicate an improvement of accuracy would happen. Doing such, I was able to improve the accuracy to: 81.33140376266282%. I produced the loss curve shown in Fig. 15. In this, you can see that the number of epochs used increased, and the convergence is more steadily flat.

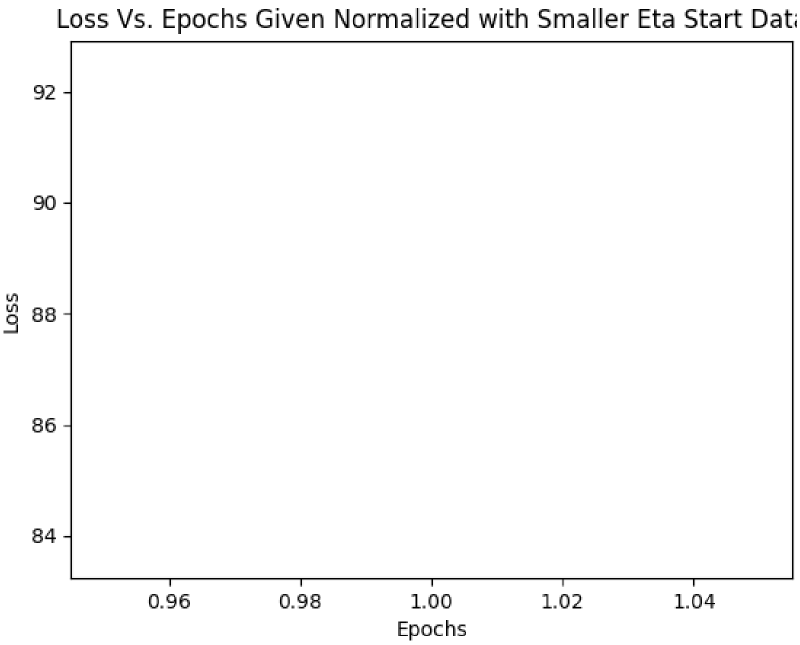


Figure 7

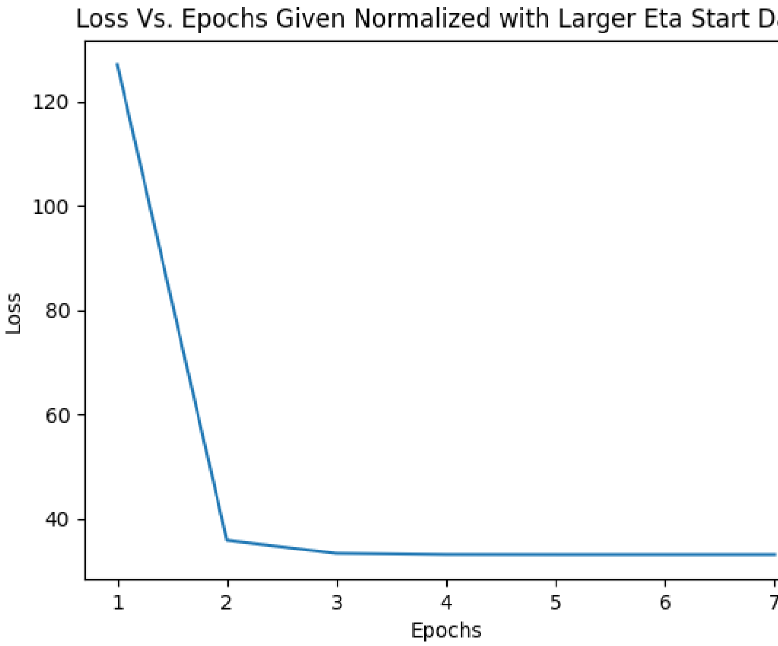


Figure 8

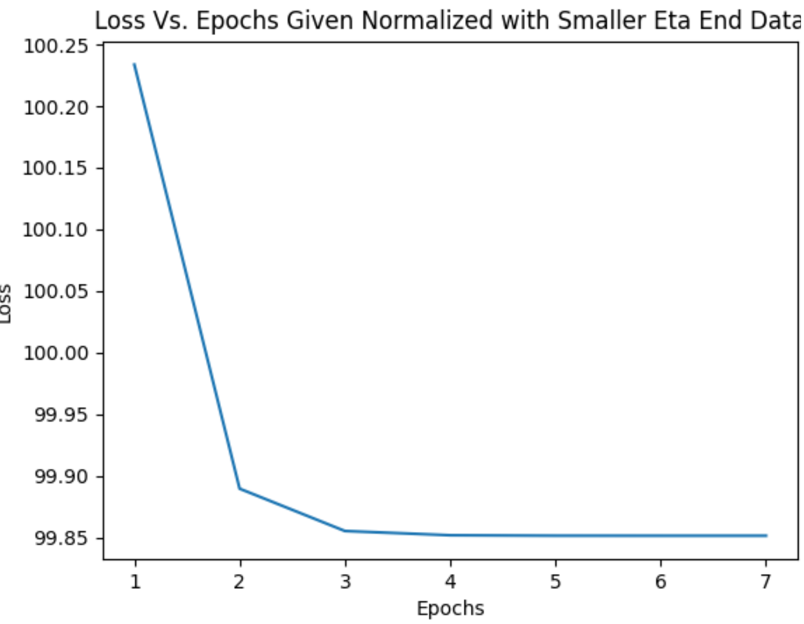


Figure 9

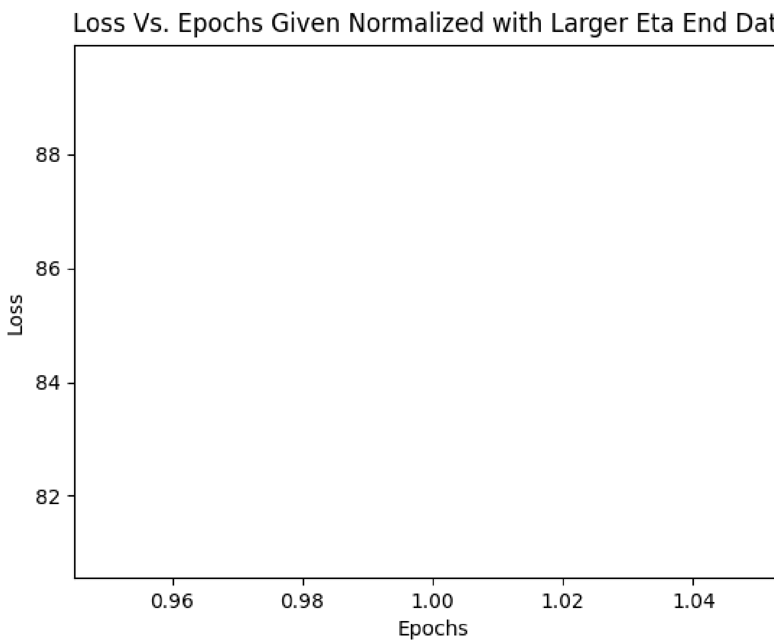


Figure 10

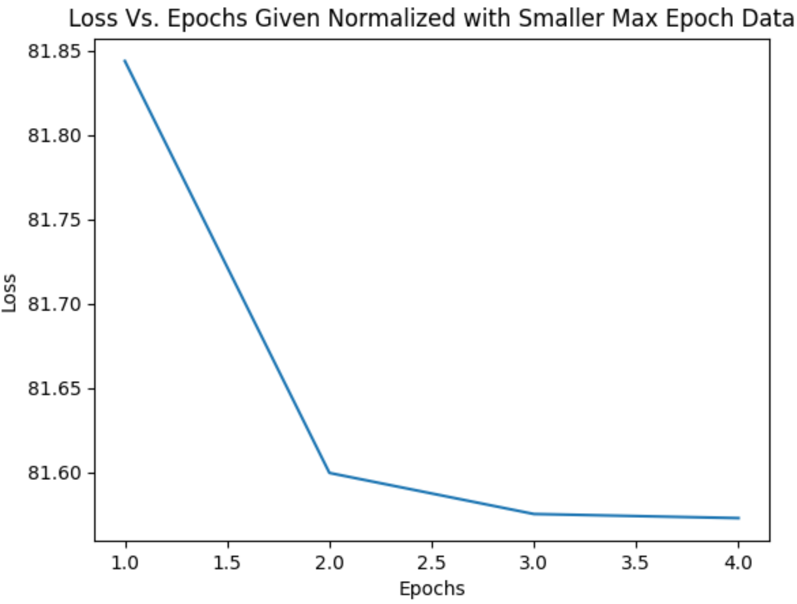


Figure 11

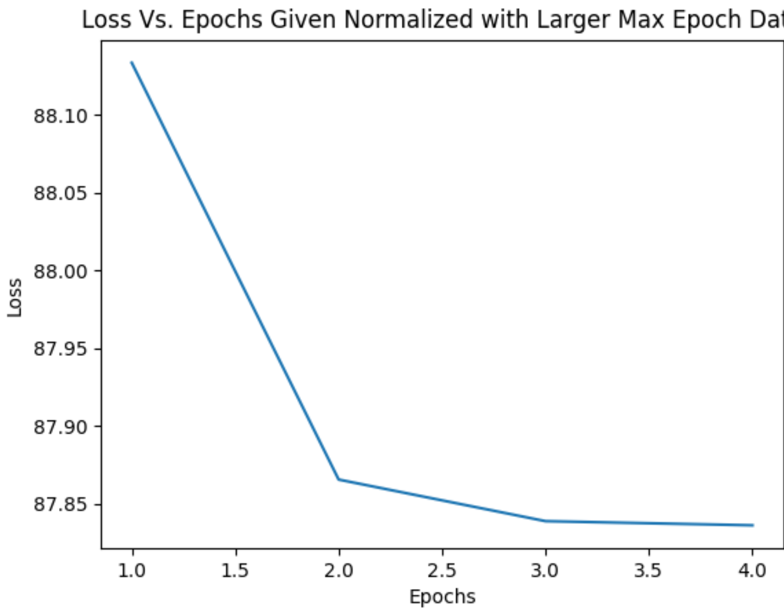


Figure 12

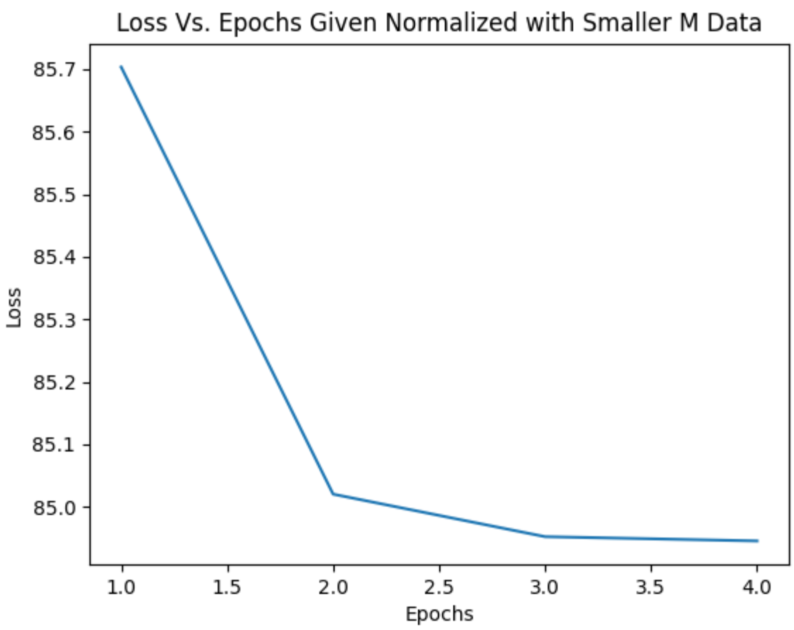


Figure 13

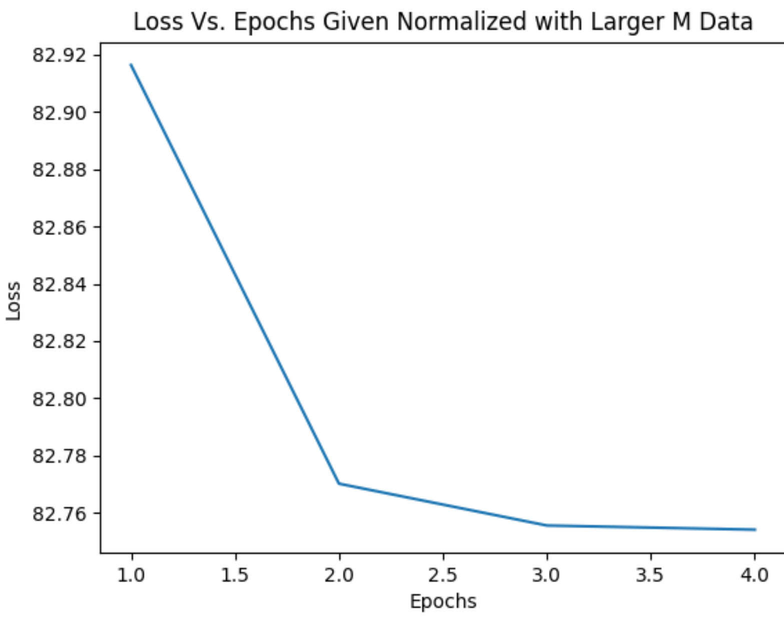


Figure 14

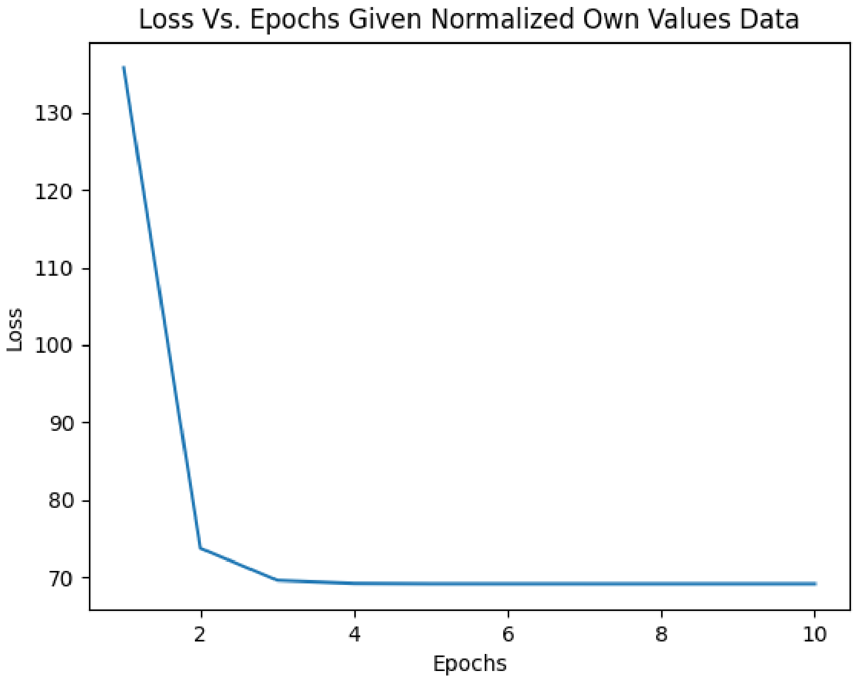
Loss Vs. Epochs Given Normalized Own Values Data
130
120
110
100
90
80
70
Loss
2
4
6
8
10
Epochs

Figure 15