CSE 512 Spring 2021 - Machine Learning - Homework 2

Your Name: Irfan Ahmed

Solar ID : 113166464

NetID Email: irfan.ahmed@stonybrook.edu

1.) $P(X=k|\lambda) = \dfrac{\lambda^{x} e^{-\lambda}}{x!}$ $k \in \{0, 1, 2, \ldots\}$

2.) Log-likelihood :

$$P(D|\theta) = \frac{\lambda^{x} \cdot e^{-\lambda}}{x!}$$

$\theta \to \lambda$

$\Updownarrow$

$$P(x_1, \ldots, x_n|\lambda) = P(x_1/\lambda) \cdot P(x_2|\lambda) \cdots P(x_n|\lambda)$$

Given x_i's are i.i.d

$$\log P(x_1 \ldots x_n|\lambda) = \sum_{i=1}^{n} \log P(x_i|\lambda) = \sum_{i=1}^{n} \log \frac{\lambda^{x_i} \cdot e^{-\lambda}}{x_i!}$$

$$= \sum_{i=1}^{n} -\lambda + \log \frac{\lambda^{x_i}}{x_i!} = -n\lambda + \sum_{i=1}^{n} \log \frac{\lambda^{x_i}}{x_i!}$$

$$= -n\lambda + \sum_{i=1}^{n} \left(\log \lambda^{x_i} - \log x_i!\right) = -n\lambda + \log\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} x_i!$$

1.1.b.) $\log(P(x_1 \dots x_n | \lambda)) = -n\lambda + \log\lambda \sum_{i=1}^{n} x_i - \sum_{i=1}^{n} \log x_i!$

MLE:

$$\frac{d}{d\lambda}(\cdot) = -n + \frac{1}{\lambda}\sum_{i=1}^{n} x_i - 0$$

$$0 = -n + 1/\lambda \sum_{i=1}^{n} x_i$$

$$\lambda = \frac{1}{n}\sum_{i=1}^{n} x_i$$

$$\frac{d^2}{d\lambda^2}(\cdot) = -\frac{1}{\lambda^2}\sum_{i=1}^{n} x_i < 0$$

[x_i's are in minutes]

$\sum_{i=1}^{n} x_i > 0$

c.)

i Trip	1	2	3	4	5	6	7
X Wait time	4	12	3	5	6	9	17

$$\lambda_{MLE} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

$$= \frac{1}{7}(4+12+3+5+6+9+17)$$

$$= \frac{1}{7}(56)$$

$$\lambda_{MLE} = 8$$

1.2)

$X|_\lambda \sim \text{Poisson}(\lambda)$

$\lambda \sim \text{Gamma}(\alpha,\beta) \Rightarrow P(\lambda|\alpha,\beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \lambda^{\alpha-1} e^{-\beta\lambda}, \lambda > 0$

1) Posterior distribution over λ

$$P(\lambda|x) = \frac{P(X|\lambda)\cdot P(\lambda)}{P(x)}$$

$P(x) = \int P(X|\lambda_i)\cdot P(\lambda_i)\,d\lambda$ ← does not depend on λ
↳ normalizing constant

$$P(X|\lambda)\cdot P(\lambda) = P(X_1,\dots X_n|\lambda)\cdot P(\lambda)$$

$$= \left(\prod_{i=1}^{n} \frac{e^{-\lambda}\cdot\lambda^{x_i}}{x_i!}\right) P(\lambda)$$

$$= e^{-n\lambda}\cdot\left(\prod_{i=1}^{n}\frac{\lambda^{x_i}}{x_i!}\right)\frac{\beta^\alpha\cdot\lambda^{\alpha-1}\cdot e^{-\beta\lambda}}{\Gamma(\alpha)}$$

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

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

$$= \underbrace{\frac{\beta^\alpha}{\Gamma(\alpha)\cdot\prod_{i=1}^{n}x_i!}}_{\text{constant for }\lambda}\cdot e^{-(n+\beta)\lambda}\cdot\lambda^{\sum_{i=1}^{n}x_i+\alpha-1}$$

$$\therefore P(\lambda|x) = \underbrace{\frac{\beta^\alpha}{\Gamma(\alpha)\left(\prod_{i=1}^{n}x_i!\right)P(x)}}_{\text{constant}}\cdot e^{-(n+\beta)\lambda}\cdot\lambda^{\sum_{i=1}^{n}x_i+\alpha-1} \quad \text{—— ①}$$

↓

Gamma distribution

with $\hat{\alpha} = \sum_{i=1}^{n}x_i + \alpha$

$\hat{\beta} = n+\beta$

Constant can be calculated by using Law of probability.

$$\sum_{\lambda}\text{constant}\cdot e^{-(n+\beta)\lambda}\cdot\lambda^{\sum_{i=1}^{n}x_i+\alpha-1} = 1$$

1.2.2) MAP estimate of λ

From ①,

$$P(\lambda|x) = K \cdot e^{-(n+\beta)\lambda} \cdot \lambda^{\sum_{i=1}^{n} x_i + \alpha - 1}, \quad K = \frac{\beta^\alpha}{\Gamma(\alpha) \cdot P(x) \cdot \prod_{i=1}^{n} x_i!} \text{ : constant for } \lambda$$

$$\log(P(\lambda|x)) = \log K - (n+\beta)\lambda + \left(\sum_{i=1}^{n} x_i + \alpha - 1\right) \log \lambda$$

Differentiating to get optimum

$$\frac{d \log(P(\lambda|x))}{d\lambda} = 0 - (n+\beta) + \frac{1}{\lambda}\left(\sum_{i=1}^{n} x_i + \alpha - 1\right) = 0$$

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

$$\frac{d^2}{d\lambda^2}(\cdot) = 0 - \frac{1}{\lambda^2}\left(\sum_{i=1}^{n} x_i + \alpha - 1\right)$$

$\sum_{i=1}^{n} x_i > 0$ [$\because$ x_i's are in minutes]

$\alpha - 1 > 0$ [$\because \alpha > 1$]

$$\therefore \frac{d^2}{d\lambda^2}(\cdot) < 0 \Rightarrow \lambda_{MAP} = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{n+\beta}$$

λ_{MAP} is mode of the gamma distribution

$$\text{Mode of Gamma}(\alpha, \beta) = \frac{\alpha - 1}{\beta}$$

$$\text{Here } \lambda_{MAP} = \frac{\sum_{i=1}^{n} x_i + \alpha - 1}{n+\beta}$$

1.3.1) $X \sim \text{Poisson}(\lambda) \; : \; P(X|\lambda) = \dfrac{e^{-\lambda} \cdot \lambda^{x}}{x!}$

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

$$\ln \eta = -2\lambda$$

$$\lambda(\eta) = \frac{-1}{2} \ln \eta$$

MLE:

$$P(X|\lambda(\eta)) = \frac{e^{-\lambda(\eta)} \cdot (\lambda(\eta))^{x}}{x!}$$

$$= \frac{e^{-(\frac{-1}{2}\ln\eta)} \cdot (\frac{-1}{2}\ln\eta)^{x}}{x!}$$

$$= \frac{\eta^{1/2} \cdot (\frac{-1}{2}\ln\eta)^{x}}{x!}$$

$$\log(\cdot) = \frac{1}{2}\log\eta + \log(-\tfrac{1}{2}\ln\eta)^{x} - \log x!$$

$$= \frac{1}{2}\log\eta + x\log(\tfrac{-1}{2}\ln\eta) - \log x!$$

Differentiating

$$\frac{d}{d\eta}(\cdot) = \frac{1}{2\eta} + \frac{x}{\frac{-1}{2}\ln\eta} \cdot \left(\frac{-1}{2\eta}\right) - 0 = 0$$

$$\frac{x}{\eta\ln\eta} = \frac{-1}{2\eta} \qquad [\because \eta \neq 0]$$

$$-2x = \ln\eta$$

$$\eta_{MLE} = e^{-2x}$$

$\therefore$ If $\hat{\eta} = e^{-2x}$ then $\hat{\eta}$ is MLE estimate of η

1.3.2) Bias of $\hat{\eta} = E(\hat{\eta}) - \eta$

$$E(\hat{\eta}) = E(e^{-2X}) = \sum_{x=0}^{\infty} e^{-2x} \cdot Pr(X=x)$$

$$= \sum_{x=0}^{\infty} e^{-2x} \cdot \frac{\lambda^{x} \cdot e^{-\lambda}}{x!} \quad [\because X \sim \text{Poisson}(\lambda)]$$

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

Using Taylor's expansion

$$e^{x} = \sum_{k=0}^{\infty} \frac{x^{k}}{k!} \rightarrow \quad = e^{-\lambda} \left[e^{e^{-2}\lambda} \right]$$

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

$$\text{Bias} = E(\hat{\eta}) - \eta = e^{-(1 - 1/e^{2})\lambda} - e^{-2\lambda} \quad [\because \eta = e^{-2\lambda}]$$

1.3.3) $(-1)^x$ is unbiased estimate of η

$$E((-1)^x) = \sum_{x=0}^{\infty} (-1)^x \cdot P_x(X=x)$$

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

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

Taylor series:

$$e^x = \sum_{k=0}^{\infty} \frac{x^k}{k!} \rightarrow \quad = e^{-\lambda} \cdot e^{-\lambda}$$

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

$$\text{Bias} = E((-1)^x) - e^{-2\lambda} = e^{-2\lambda} - e^{-2\lambda} = 0$$

Hence $(-1)^x$ is unbiased estimate of η

MSE of $\hat{\theta}$

$$\hookrightarrow E((\hat{\theta}-\theta)^2)$$

$$= E((\hat{\theta})^2) - 2E(\hat{\theta}\cdot\theta) + E(\theta^2)$$

$$= E(\hat{\theta}^2) - 2\theta E(\hat{\theta}) + \theta^2 E(1)$$

$$= E(\hat{\theta}^2) - 2\theta E(\hat{\theta}) + \theta^2$$

MSE for e^{-2X} (Biased), $\theta = e^{-2\lambda}$

$$= E(e^{-4X}) - 2e^{-2\lambda} E(e^{-2X}) + e^{-4\lambda}$$

$$= e^{-[1-e^{-4}]\lambda} - 2e^{-2\lambda}(e^{-(1-e^{-2})\lambda}) + e^{-4\lambda}$$

$$= e^{\lambda(e^{-4}-1)} - 2e^{\lambda(e^{-2}-3)} + e^{-4\lambda}$$

MSE for $(-1)^x$ (Unbiased)

$$= E(((-1)^x)^2) - 2e^{-2\lambda} E((-1)^x) + e^{-4\lambda}$$

$$= E(1^x) - 2e^{-2\lambda} e^{-2\lambda} + e^{-4\lambda}$$

$$= 1 - e^{-4\lambda}$$

$$E(e^{tx}) = \sum_{x=0}^{\infty} e^{tx} P_x(X=x)$$

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

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

$$= e^{-\lambda} \cdot e^{e^t \cdot \lambda}$$

$$E[e^{tx}] = e^{-[1-e^t]\lambda}$$

The probabilities of unbiased estimator can be negative and also MSE of biased estimator is far better than MSE of unbiased estimator.

Probabilities are negative when X is odd.

2.1) Given:

$$P(Y=i \mid x;\theta) = \frac{\exp(\theta_i^T \bar{x})}{1+\sum_{j=1}^{K-1} \exp(\theta_j^T \bar{x})} \quad i=1,2,\dots,K-1$$

LHS

$$\frac{\partial \log(P(y^i \mid \bar{x}^i;\theta))}{\partial \theta_c} = \frac{\partial}{\partial \theta_c}\left(\log \exp(\theta_i^T \bar{x}^i) - \log\left(1+\sum_{j=1}^{K-1} \exp(\theta_j^T \bar{x}^i)\right)\right) \quad \forall i=1,\dots,K-1$$

$$= \frac{\partial}{\partial \theta_c}\left(\theta_i^T \bar{x}^i - \log\left(1+\sum_{j=1}^{K-1} \exp(\theta_j^T \bar{x}^i)\right)\right)$$

Given $\delta(c=y^i)$ is indicator function. which gives 1 when $c=y^i$, '0' otherwise

$\frac{\partial}{\partial \theta_c} \theta_i^T \bar{x}^i$

This will be $\bar{x}^i$ only when $i=c$. for all other $i \neq c$

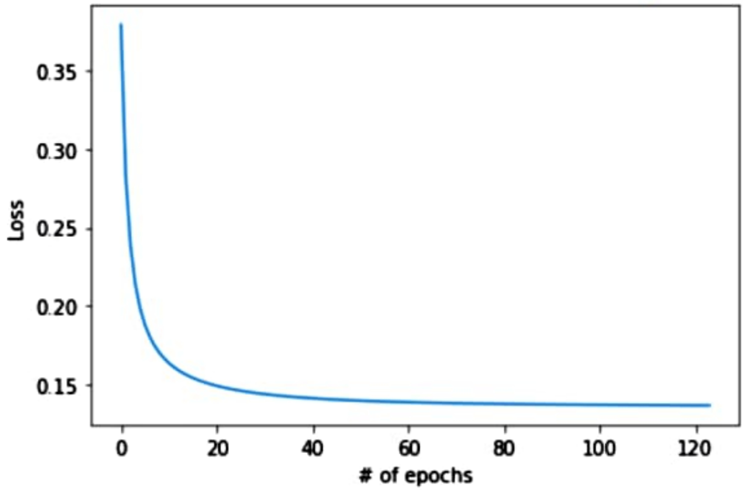
$\frac{\partial}{\partial \theta_c} \theta_i^T \bar{x}^i = 0 \quad (i \neq c)$

$$\rightarrow = \delta(c=y^i)\,\bar{x}^i - \frac{\exp(\theta_c^T \bar{x}^i)\,\bar{x}^i}{1+\sum_{j=1}^{K-1} \exp(\theta_j^T \bar{x}^i)}$$

$$= \delta(c=y^i)\,\bar{x}^i - P(Y=c \mid \bar{x}^i;\theta)\cdot \bar{x}^i$$

$$= \left[\delta(c=y^i) - P(c \mid \bar{x}^i;\theta)\right] \bar{x}^i$$

//

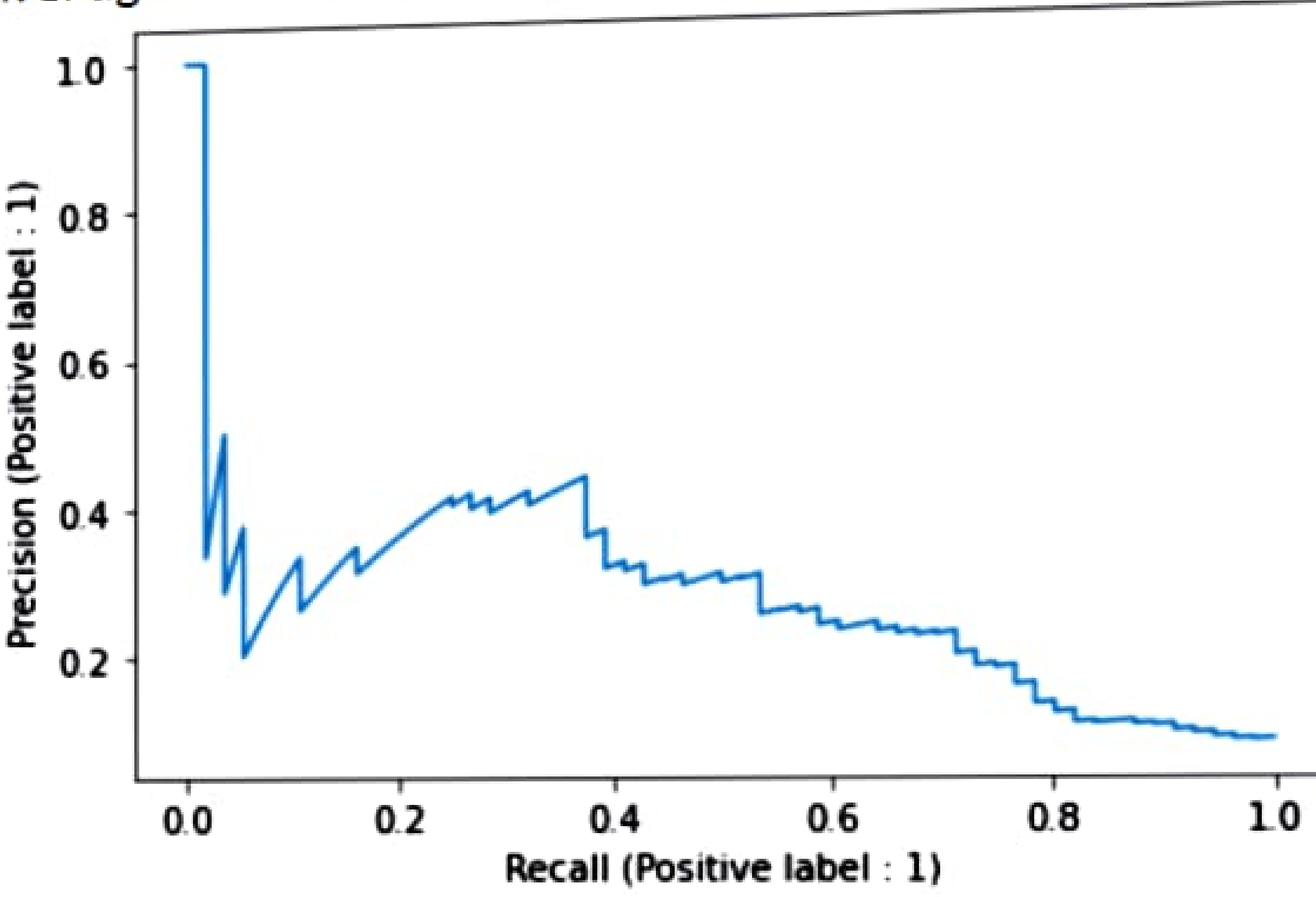


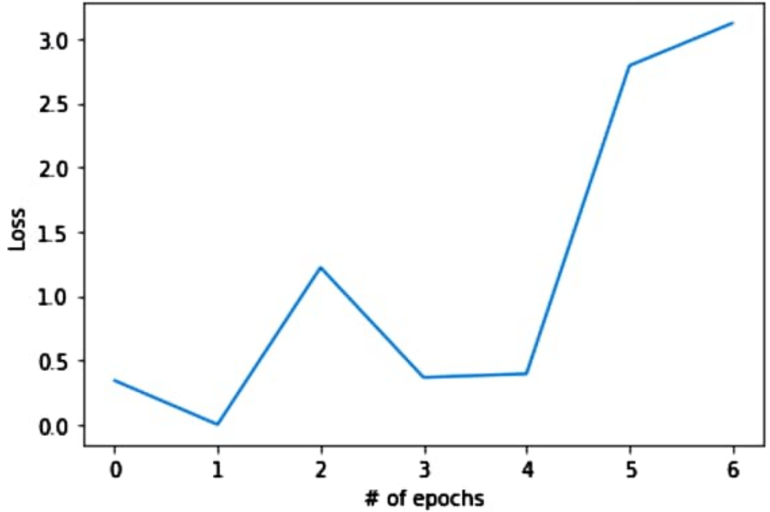
With increasing epochs, the loss function converges to a constant value and does not increase or decrease much from the previous value. From the graph it can be observed that beyond 40 epochs, there is no use training the model further as the loss has converged.

```
----------Training Data----------



Performance Metrics:


Accuracy Score:0.942836468885673

 Confusion Matrix :
[[0.98299723 0.01700277]
 [0.4893617  0.5106383 ]]

 Accuracy from confusion matrix:0.7468177649899463
----------Testing Data----------



Performance Metrics:


Accuracy Score:0.9059334298118669

 Confusion Matrix :
[[0.95748031 0.04251969]
 [0.67857143 0.32142857]]

 Accuracy from confusion matrix:0.6394544431946007
```

```
avg_precision_and_precision_recall_curve(X_test,y
```

```
Average Precision Score: 0.2811980458553681
```





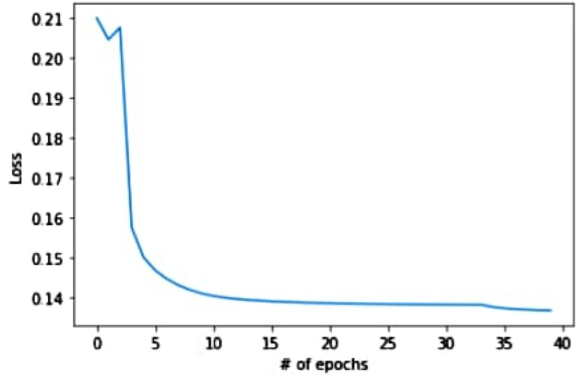
Without feature normalization, the model is increasing in loss and then converging. The number of epochs have reduced are now being dependent on eta_start and eta_end. It implies that the loss is not converging but decreasing the learning rate in this regard. Changing eta_end to 10^-12 shows loss has increased to a certain level and converged there. The final performance of the model has been decreased but nothing significant.

2.3.2.b)Increasing (eta_start)learning rate(0.1), decreases my loss function faster and converges earlier (at 30). The accuracy on testing data increased to 91.7% (earlier 90.5%).
Decreasing learning rate(0.001), increases no of epochs to converge. The accuracy remains same with slight difference.

Decreasing batch size(100) decreases convergence rate of my loss function and the epochs have increased. Final performance is lowered a little bit.
Increasing batch size(doubled now) maintained the same final performance but the loss converged at 30 epochs(earlier).

Increasing eta_end, increases the no of epochs keeping the performance nearly same as original params. It looks like the model reaches the same performance but slowly.

Decreasing max epochs (<20) does not coverge the loss function. Increasing the max epochs gives no meaning as loss converges and no point in training further.



2.3.c) The values of hyperparameters chosen are eta_start = 0.1(Increased), m = 512(Doubled) eta_end = 0.00001, max_epochs = 40.

Training Data :

- Accuracy Score: 0.9439218523878437
- Accuracy from confusion matrix: 0.7551306966844182

Testing Data :

- Accuracy Score: 0.9117221418234442
- Accuracy from confusion matrix: 0.6426040494938133

With these new values, the number of epochs taken to converge is around 40 and same accuracy with original values. Reasoning behind chosing these values comes from 2.3.b,where increasing eta and increasing m maintained same accuracy but converged faster. Hence the early stopping was identified and used in this hyperparameters.

Other approaches such as Grid search and Random search can be used for hyperparameter optimisation but the computations are taking a toll.