1.1.

In Generative classifiers we assume some functional form for P(X / Y) and P(Y). We estimate the parameters of P(X /Y) and P(Y) directly from the training data and then calculate P(Y /X) based on this. This is not what we do in logistic regression.

In Discriminative classifiers we assume some form for P(Y / X) and estimate the parameters of P(Y / X) directly from the training data. We are following the same thing in Logistic regression we are having some form for P(Y=1 /X) = and we estimate w’s based on training data.

So for the above reasons we can say logistic regression is a Discriminative classifier rather than generative classifier.

1.2.

The decision boundary of a logistic regression is a straight line. Let us assume

P(Y=1 /X) = and we have P(Y=0 /X) =

for P(Y=1 / X) > P(Y=0 / X)

We need to have > 1 , this happens when > 0 .

The equation > 0 is a liner decision boundary.

1.3.

1.3a. l(w) = ln ,w) , we know that log(ab) = log(a) + log(b)

= , as P(Y=y /X) = , here

If y =0 we will have P(Y=0/X) = = P(Y=0/X)

If y =1 we will have P(Y=1/X) = = P(Y=0/X)

= , and as we know that log = a\*log(x) and also log(ab) = log(a) + log(b)

= , and we are given = and

= , substituting it you get

= , expanding terms and simplifying it we get,

= , we know that ln(1 /a) = -ln(a) using that we get

=

Applying partial derivative on both sides.

=

The blue part can be simplified down to

For the red part we can apply the chain rule. We know that partial derivative of log(f(x)) with respect to x is

=

Now if we apply the above rule to we get

= \* () , we know that

= . , Applying it we get.

= \* \* ()

() will be equal to .

= \* , if we substitute this in the red part of the above equation we get.

1.3b.

= - , taking common we get

= and we know that, =

=

We know that

= + , substituting it we get

= + (

For initial the equation will be

= + ( , for the ith w we will have the update rule as ,

**= + (**

2.

I have written the code for logistic regression and attached the folder. I got

optimal learning\_rate = 0.028 and optimal epochs = 260.

Training Accuracy:



Validation Accuracy:



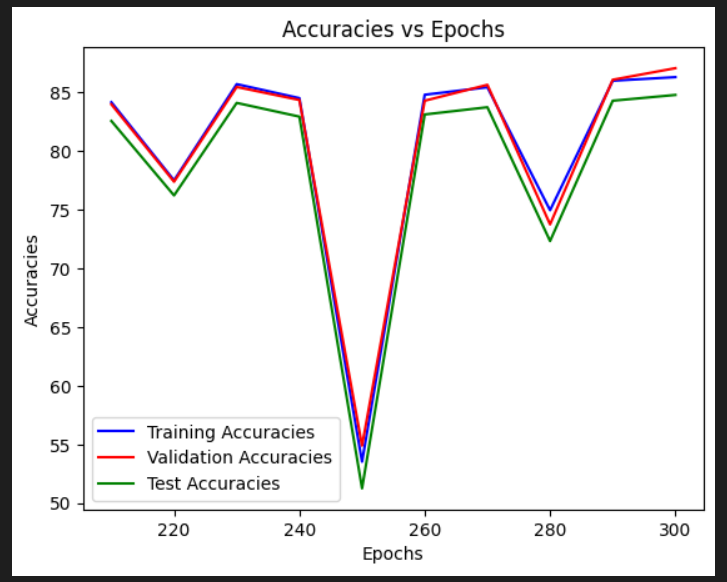
Testing Accuracy:



While training the model I have considered the whole dataset as a batch, I believe training the model by splitting training data into batches will give better outputs. I have taken random values for hyper parameters initially and tried to observe the trend how the accuracies are changing. I tried to check for different ranges and for the optimal hyper parameters I found I have drawn the plots which help me analyze better values for hyper parameters and then I got my optimal hyper parameters.

The plots asked in the question are as follows.

1. For this graph I have taken the learning rate at 0.0678 and varied epochs.



1. For this graph I have taken epochs = 90 and varied the learning rate.

