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) = $\frac{exp(w_0 + \sum_i w_i X_i)}{exp(w_0 + \sum_i w_i X_i) + 1}$ 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) = $\frac{exp(w_0 + \sum_i w_i X_i)}{exp(w_0 + \sum_i w_i X_i) + 1}$ and we have P(Y=0 /X) = $\frac{1}{exp(w_0 + \sum_i w_i X_i) + 1}$

for P(Y=1 / X) > P(Y=0 / X)
We need to have $exp(w_0 + \sum_i w_i X_i) > 1$, this happens when $(w_0 + \sum_i w_i X_i) > 0$.

The equation $(w_0 + \sum_i w_i X_i) > 0$ is a liner decision boundary.

1.3.

1.3a. l(w) = ln $\prod_{j=1}^{n} p(y^j / x^j, \text{w})$, we know that log(ab) = log(a) + log(b)

= $\sum_{j=1}^{n} ln(p(y^j/x^j, \text{w}))$, as P(Y=y /X) = $P(Y = 1/X)^y P(Y = 0/X)^{1-y}$, here
If y =0 we will have P(Y=0/X) = $P(Y = 1/X)^0 P(Y = 0/X)^1$ = P(Y=0/X)
If y =1 we will have P(Y=1/X) = $P(Y = 1/X)^1 P(Y = 0/X)^0$ = P(Y=0/X)

= $\sum_{j=1}^{n} ln(p(y^j = 1/x^j, \text{w})^{y_j} p(y^j = 0/x^j, \text{w})^{1-y_j})$, and as we know that logx^a = a*log(x) and also log(ab) = log(a) + log(b)

= $\sum_{j=1}^{n} y_j ln(p(y^j = 1/x^j, \text{w})) + (1 - y_j)\, ln(p(y^j = 0/x^j, \text{w}))$, and we are given
$p(y^j = 1/x^j, \text{w}) = \frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1}$ and $p(y^j = 0/x^j, \text{w}) = 1 - p(y^j = 1/x^j, \text{w})$
$p(y^j = 1/x^j, \text{w}) = \frac{1}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1}$, substituting it you get

= $\sum_{j=1}^{n} y_j ln(\frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1}) + (1 - y_j)\, ln(\frac{1}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1})$, expanding terms and simplifying it we get,

= $\sum_{j=1}^{n} y_j ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j)) + ln(\frac{1}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1})$, we know that ln(1 /a) = -ln(a) using that we get

= $\sum_{j=1}^{n} y_j ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j)) - ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1)$

Applying partial derivative on both sides.

$\frac{\partial(l(w))}{\partial w_i} = \frac{\partial}{\partial w_i} \sum_{j=1}^{n} y_j ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j)) - \frac{\partial}{\partial w_i} \sum_{j=1}^{n} ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1)$

The blue part can be simplified down to $\sum_{j=1}^{n} y_j x_i^j$

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

$$\frac{\partial(\text{ log(f(x))})}{\partial x} = \frac{1}{f(x)} * \frac{\partial(\text{ f(x)})}{\partial x}$$

Now if we apply the above rule to $\frac{\partial}{\partial w_i} \sum_{j=1}^{n} ln(exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1)$ we get

$= \frac{1}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1} * \frac{\partial}{\partial w_i}(exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1)$, we know that $\frac{\partial(e^{f(x)})}{\partial x} = \frac{\partial(f(x))}{\partial x}.e^{f(x)}$, Applying it we get.

$$= \frac{1}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1} * exp(w_0 + w_1 x_1^j + w_2 x_2^j) * \frac{\partial}{\partial w_i}(w_0 + w_1 x_1^j + w_2 x_2^j)$$

$\frac{\partial}{\partial w_i}(w_0 + w_1 x_1^j + w_2 x_2^j)$ will be equal to x_i^j .

$= \frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1} * x_i^j$, if we substitute this in the red part of the above equation we get.

1.3b.

$\frac{\partial(l(w))}{\partial w_i} = \sum_{j=1}^{n} y_j x_i^j - \sum_{j=1}^{n} \frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1} * x_i^j$, taking x_i^j common we get

$\frac{\partial(l(w))}{\partial w_i} = \sum_{j=1}^{n} x_i^j (y_j - \frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1})$ and we know that, $p(y^j = 1/x^j, \text{w}) = \frac{exp(w_0 + w_1 x_1^j + w_2 x_2^j)}{exp(w_0 + w_1 x_1^j + w_2 x_2^j) + 1}$

$$\frac{\partial(l(w))}{\partial w_i} = \sum_{j=1}^{n} x_i^j (y_j - p(y^j = 1/x^j, \text{w}))$$

We know that

$W^{t+1} = W^t + \eta \frac{\partial(l(w))}{\partial w_i}$, substituting it we get

$$W^{t+1} = W^t + \eta (\sum_{j=1}^{n} x_i^j (y_j - p(y^j = 1/x^j, \text{w})))$$

For initial W_0 the equation will be

$W_0^{t+1} = W_0^t + \eta (\sum_{j=1}^{n} x_0^j (y_j - p(y^j = 1/x^j, \text{w})))$, for the ith w we will have the update rule as ,

$$\boldsymbol{W_i^{t+1} = W_i^t + \eta (\sum_{j=1}^{n} x_i^j (y_j - p(y^j = 1/x^j, \text{w})))}$$

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:

```
The training accuracy is given by: 90.439064
```

Validation Accuracy:

```
The validation accuracy is given by: 89.846154
```

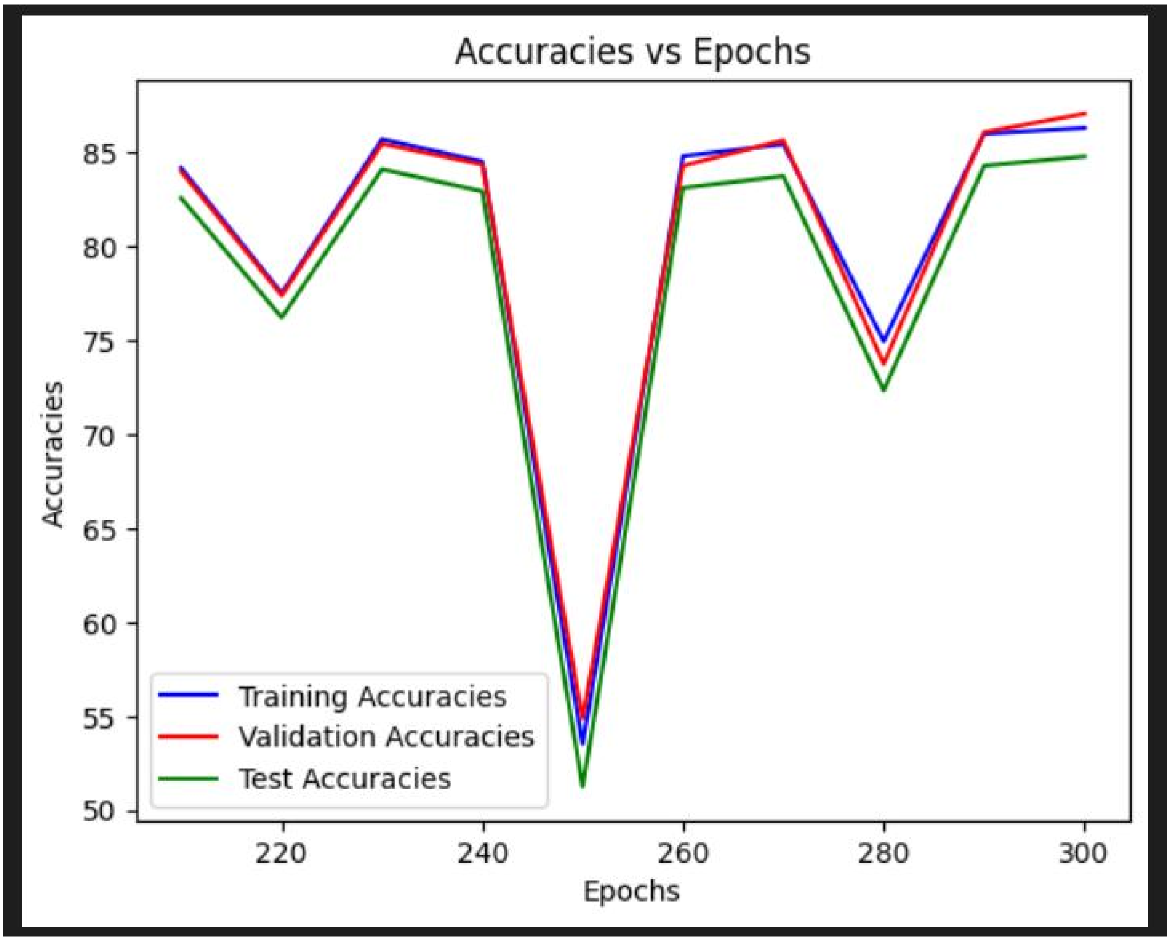
Testing Accuracy:

```
The testing accuracy is given by: 89.415385
```

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.

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



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