Logistic Regression:

* The data can be linearly separable, which is how the data should be.

Perceptron Trick:

* It uses the trial-and-error method by taking random values for the variables. The equation of a line will be like ax1+bx2+cx3+d=0.
* We use a loop until we get the number of points that are misclassified, so that it is minimum.
* Wn=w0+n(yi-yi^)xi is
  1. The above is the summary of the perceptron trick

Sigmoid Function:

* σ(z)=1/1+e^(−z)
* We used to calculate the yi^ using the perceptron trick, rather than using this, we take this value as z and use the sigmoid function. If value >0.5 it is positive, so answer =1; else it is negative and answer = 0
* if sigma is 0.5, it will be plotted on the line.
* It may be understood as the probability.

Loss function:

* It is used to decide what model is good using the sigmoid function.
* We will multiply the probabilities of all the points in the graph and decide which model has the maximum product.
* The above process is called the maximum likelihood.
* It's difficult for the large dataset. We can convert this into a sum by using the log function (the negative of it)
* Its like adding the log values of it.
* Using the log, we should choose the model with the minimum. This is called the cross-entropy.
* The formula can be changed to

Lossi​=−[yi​log(y^​i​)+(1−yi​)log(1−y^​i​)]

Gradient descent:

* Y^= σ(xw)
* It is used to find the w values for the loss function such that it is minimum.

Soft-max regression:

* If there are many outputs than yes or no.
* It is also used in deep learning and is called multinomial regression.
* softmax(zi​)=e^zi/ ∑(j=1 to k) e^zj
* k is the number of classes.
* If the outputs are m we will train the m number of models separately with their datasets. For every model, we will get the coefficients of the line. We will take sigma values and find their probability and select the coefficients which has the highest probability