Question 1:

Loss function is used to find the error between the output of our model and the given value. In other words, it tells how far off the mark our output is. As of now, there are multiple ways to determine loss. Most popular ones in Ml are 0-1 loss function and the quadratic loss function.

Talking about 0-1 loss function, it is an indicator function which returns 1 when the target and the output are not equal and 0 other wise. The quadratic loss function is a commonly used symmetric loss function. The output is identical with relation to targets that differ that differ by some value x in any direction.

Loss function is used in optimization problems with the goal of minimizing the loss. Loss functions are used in regression when finding a line of best fit minimizing the overall loss of all the points with the prediction from the line. Loss function are used while training neural networks by influencing their weights are updated. Larger the loss, larger the update. By minimizing loss, the model’s accuracy us maximized. Now that’s exactly what we want 😉

Question 2:

We’ve previously dealt with the loss function, which is a mathematical way of measuring how wrong your predictions are. While training data, we change parameters of our model to try minimize that loss function, and make our predictions as correct and optimized as possible. Optimizers tie together the loss function. It shapes and mold your model into its’s most accurate form by futzing with the weights. The loss function is the guide to the terrain, telling the optimizer when it’s moving in the right or wrong direction.

Advantages are that it’s easy to implement, easy to compute and understand. But on the other hand, it may trap at local minima, weights are calculating gradient on the whole dataset. So, if the dataset is too large than this may take years to coverage to the minima. Also, it requires large memory to calculate gradient on the whole dataset.