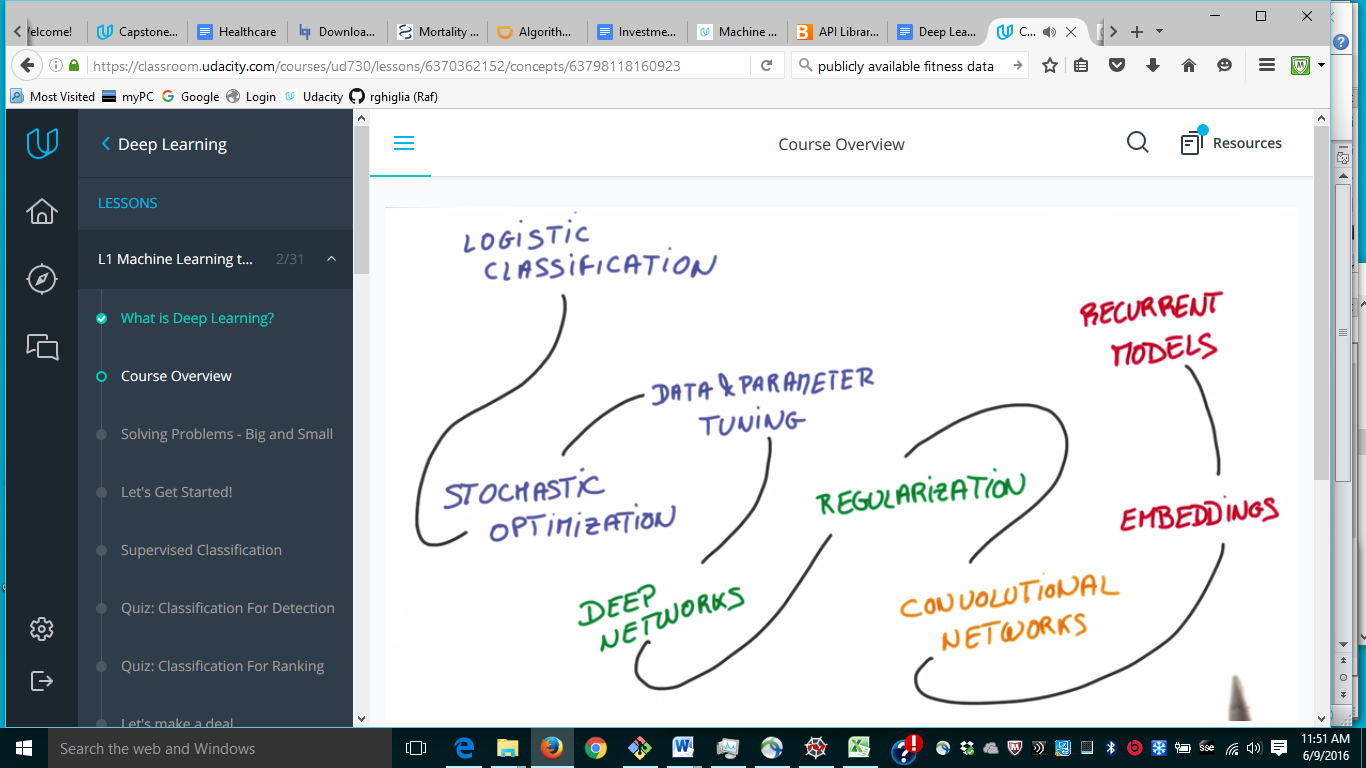
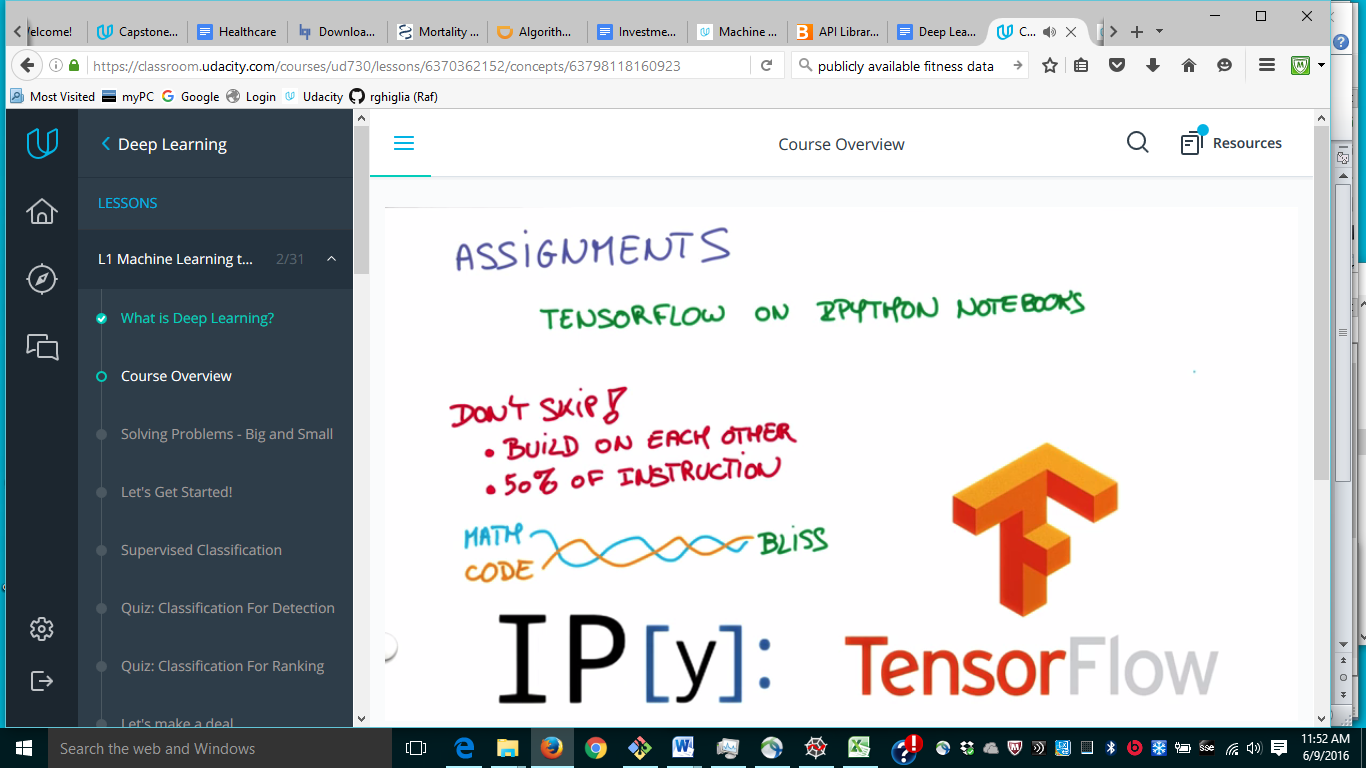
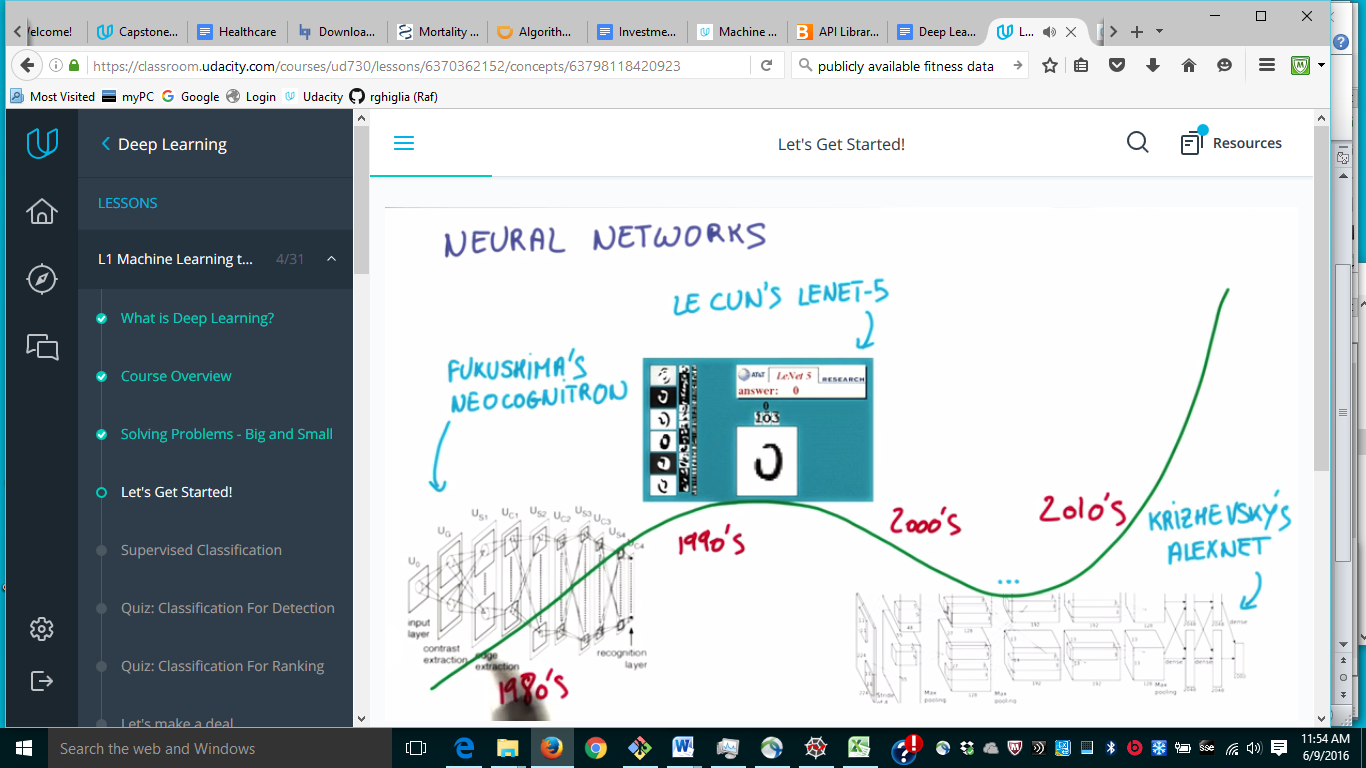
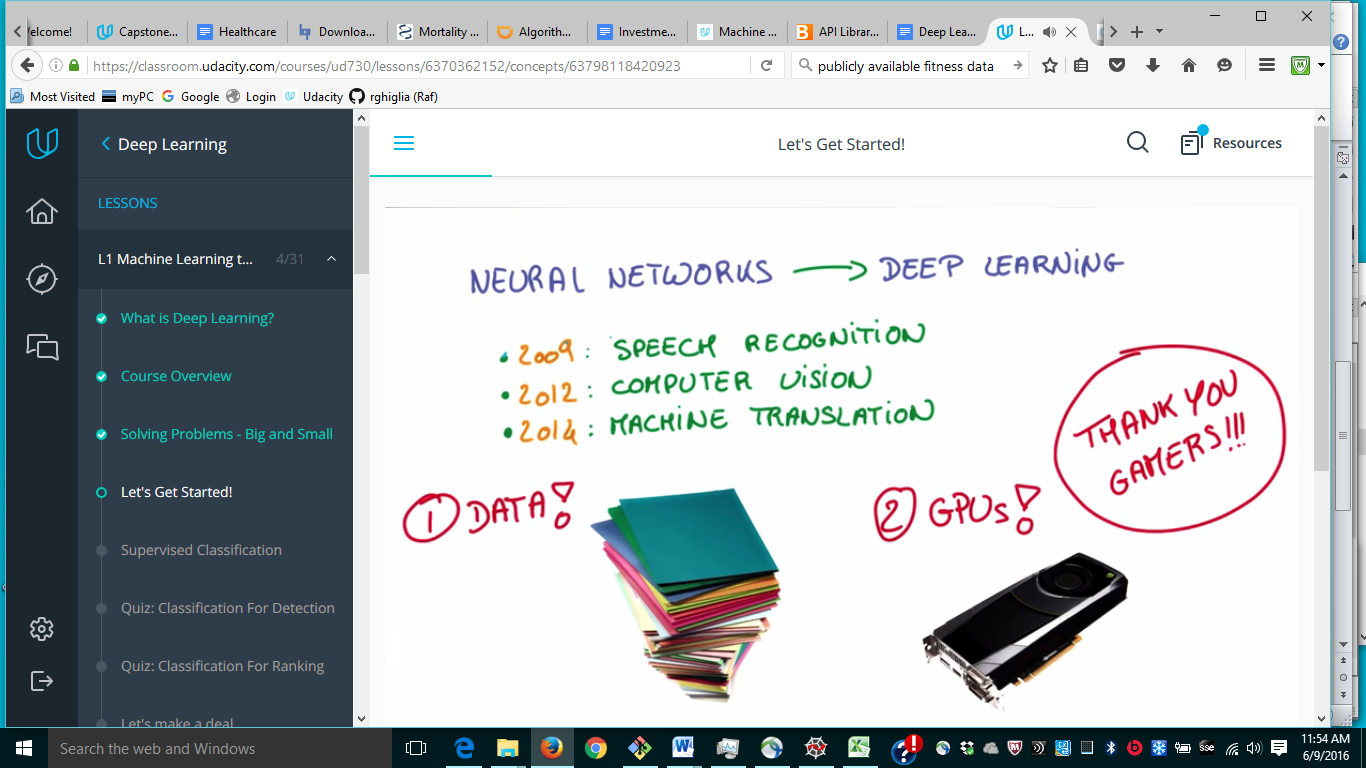
Deep Learning



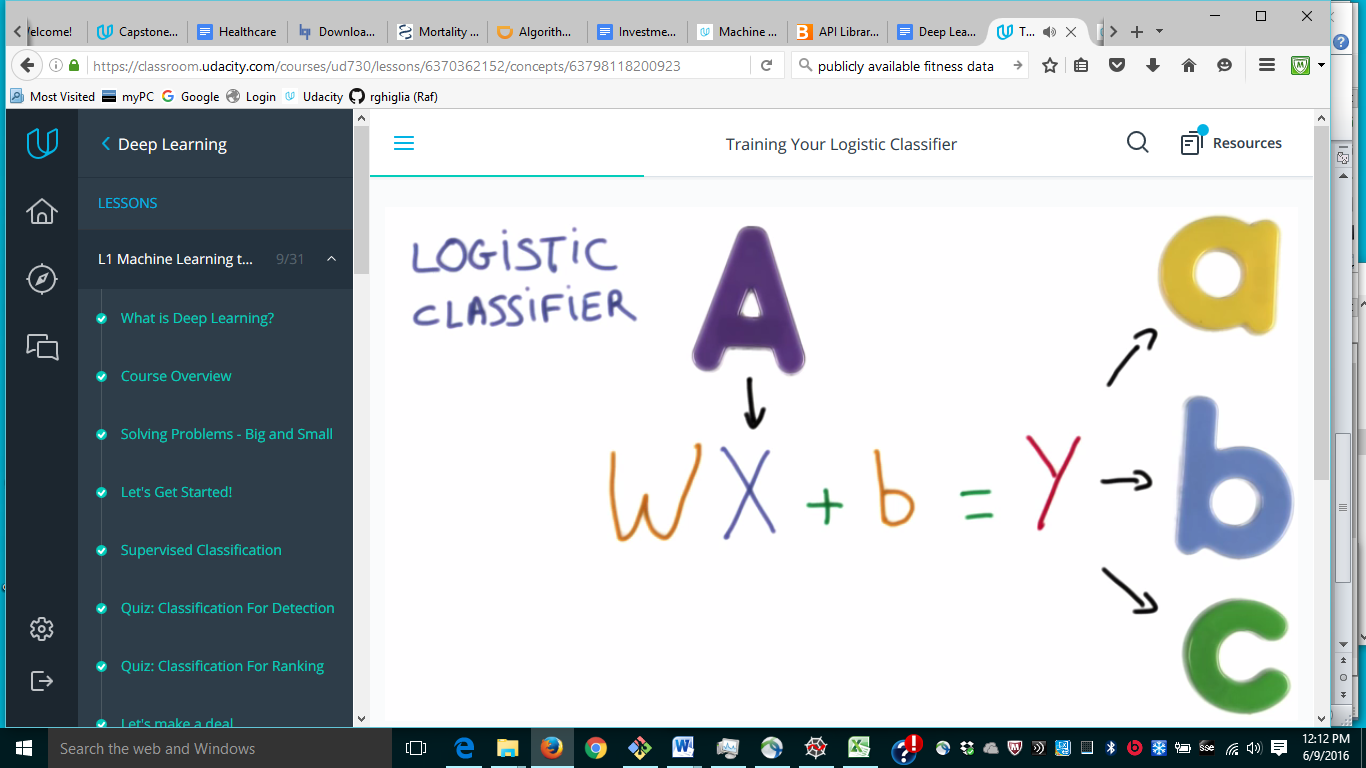


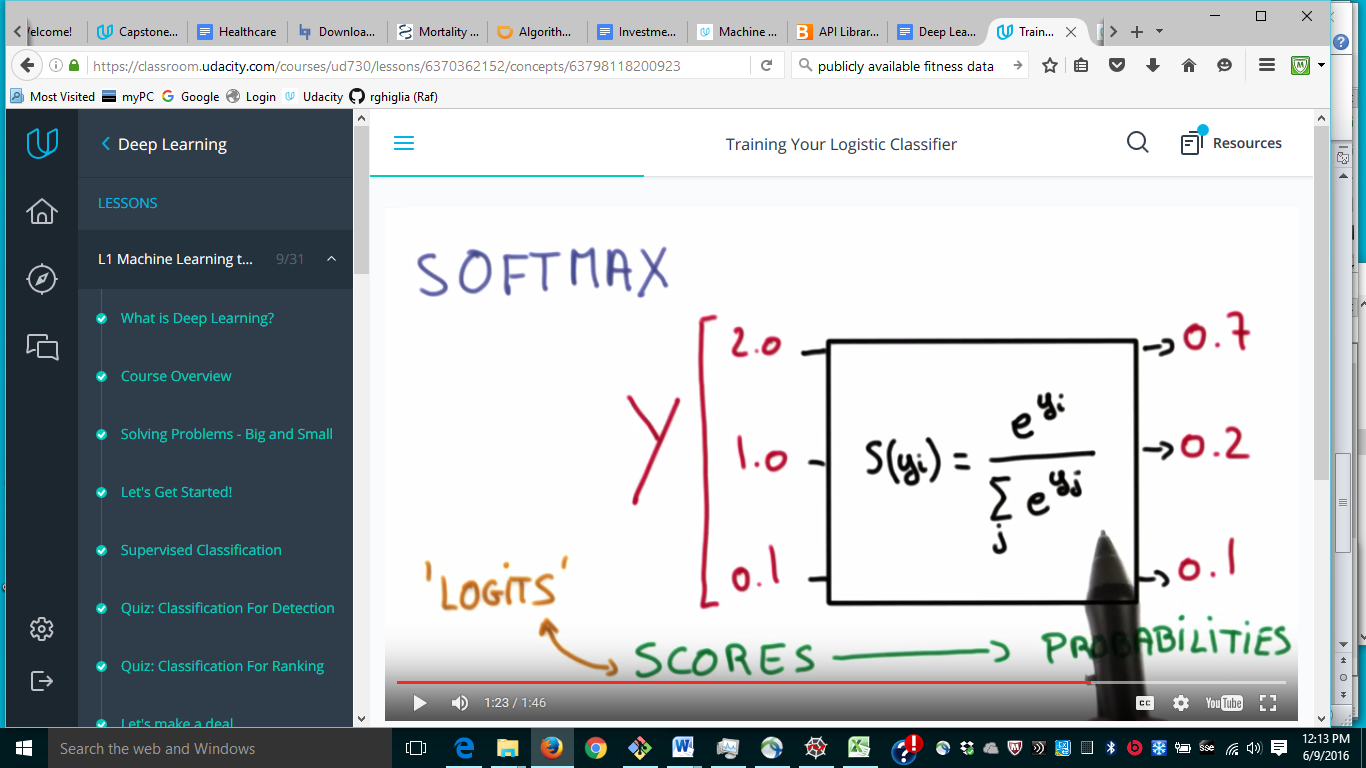


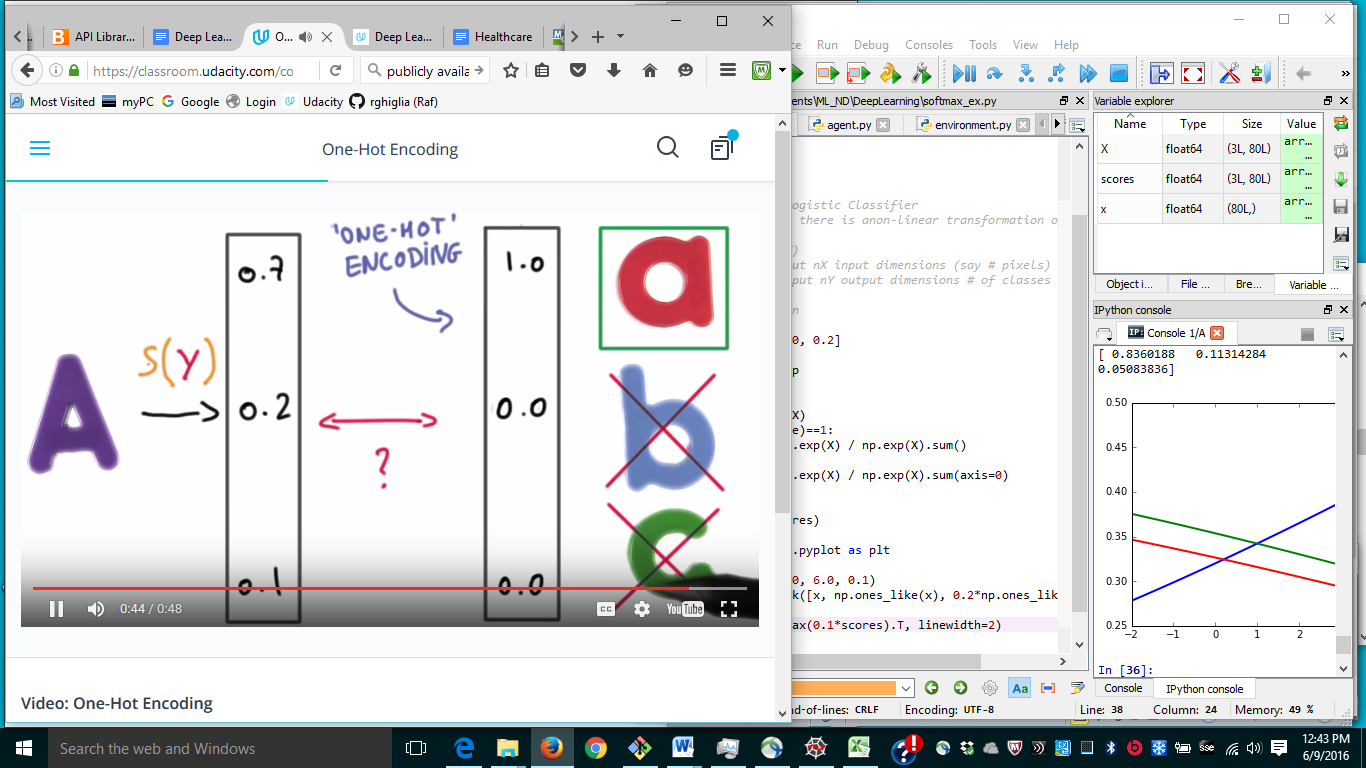


Logistic Classifier

Linear



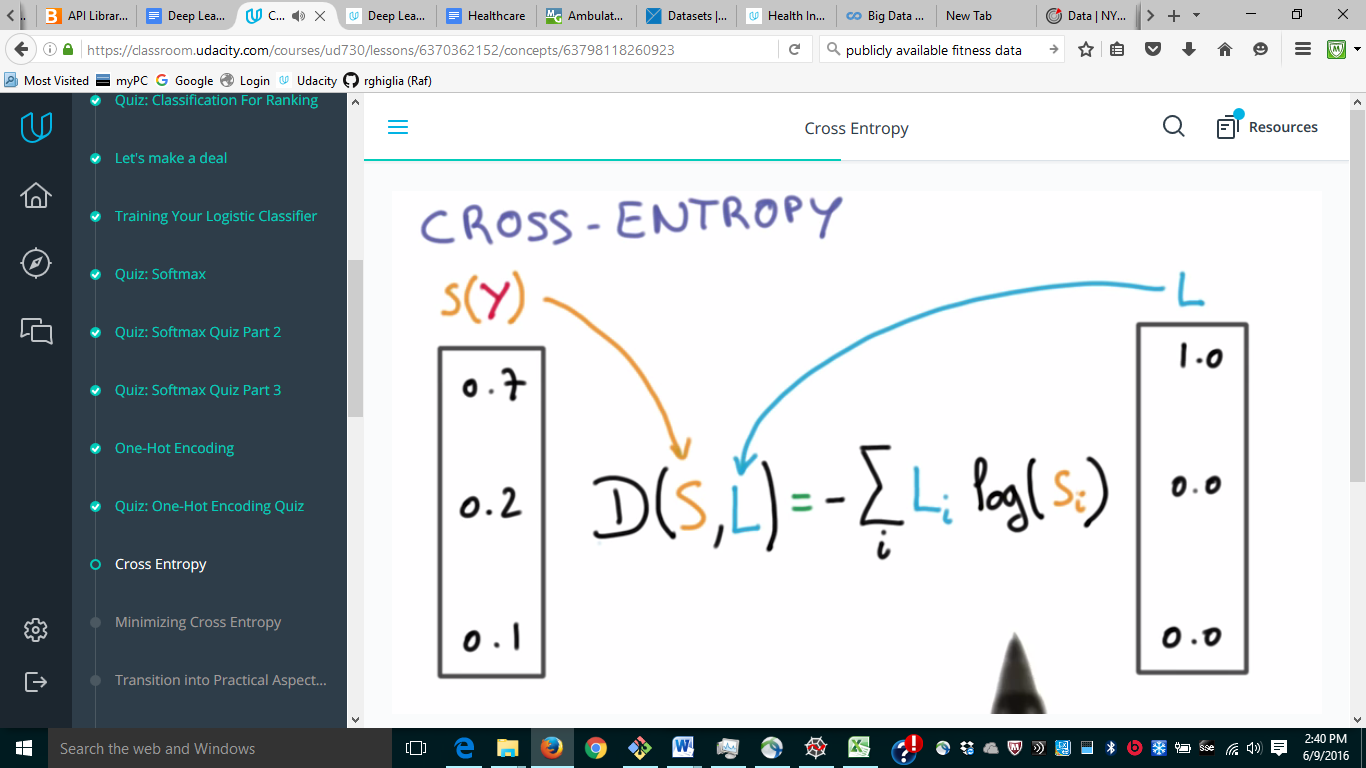


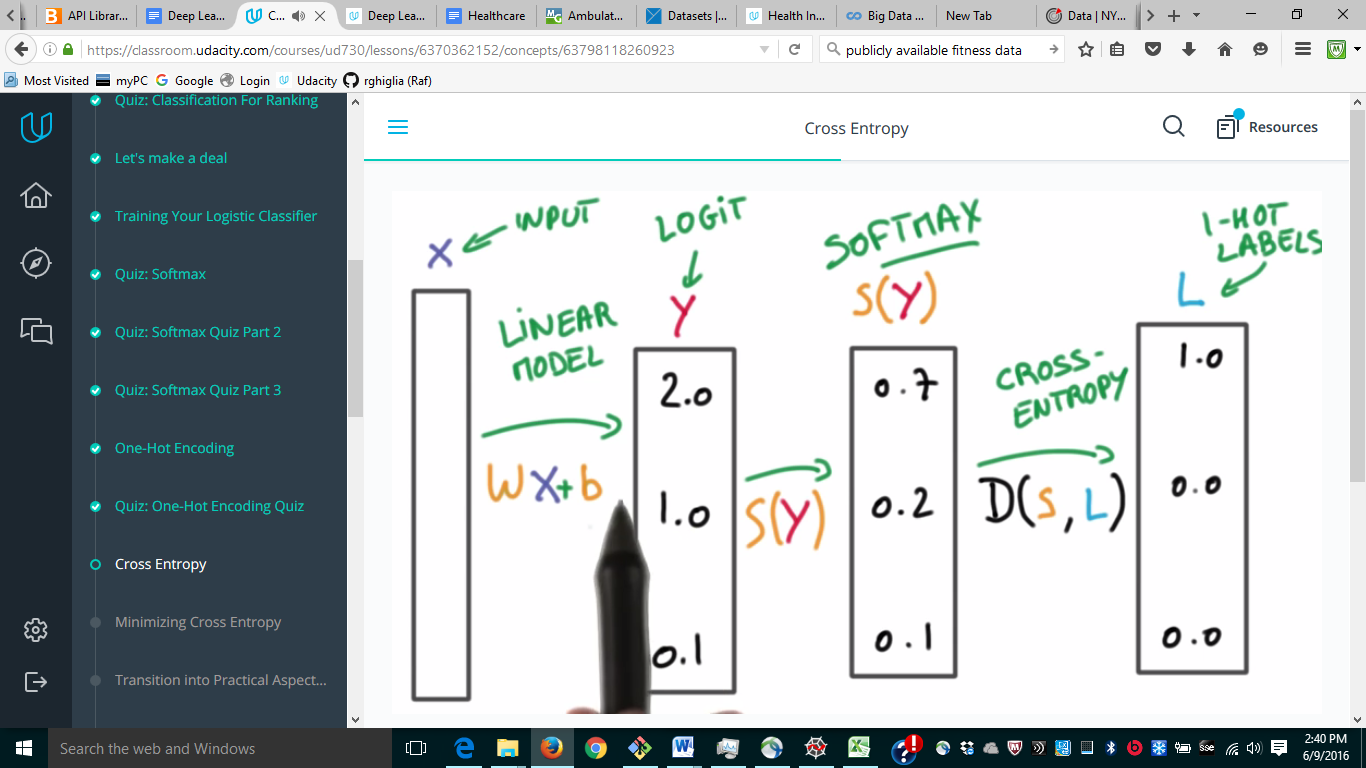


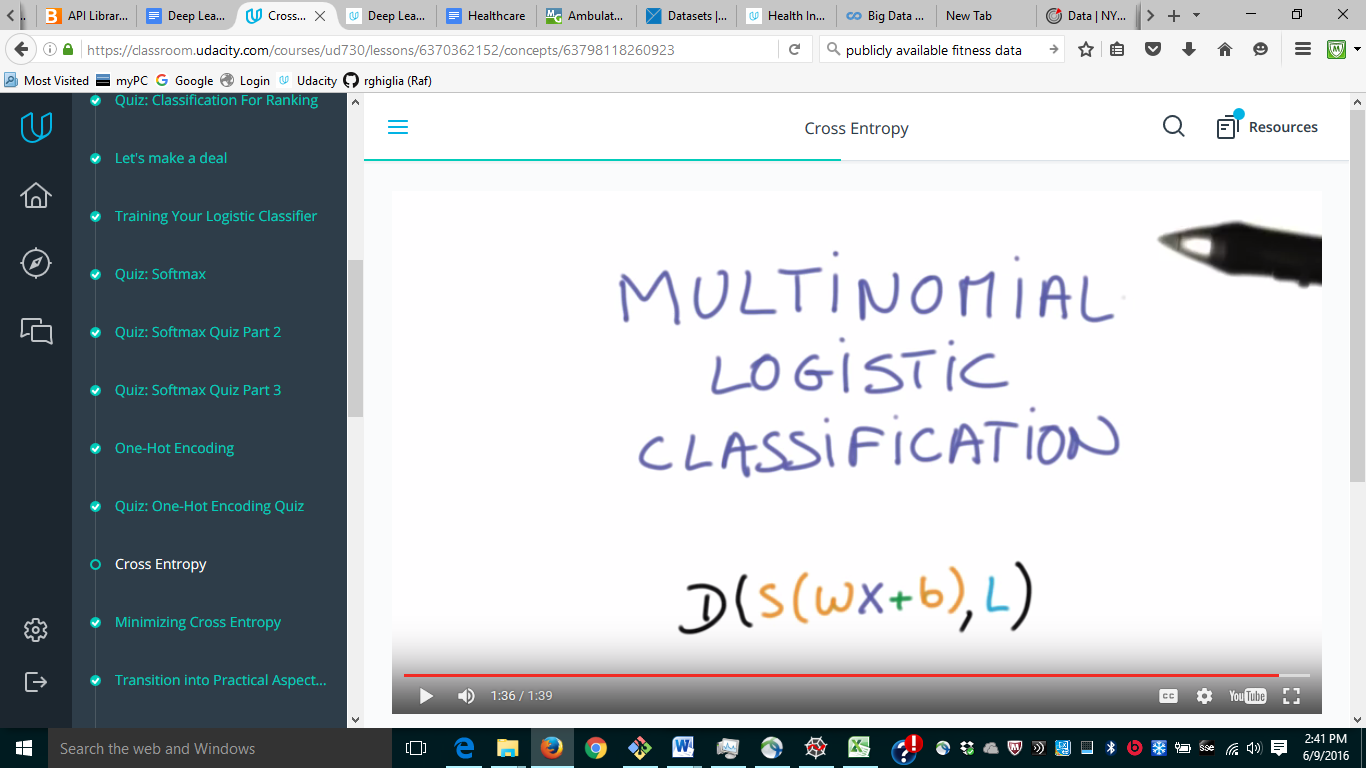
Each label is represented by a vector as long as # of labels, so a = [1, 0, 0], b = [0, 1, 0], c = [0, 0, 1]

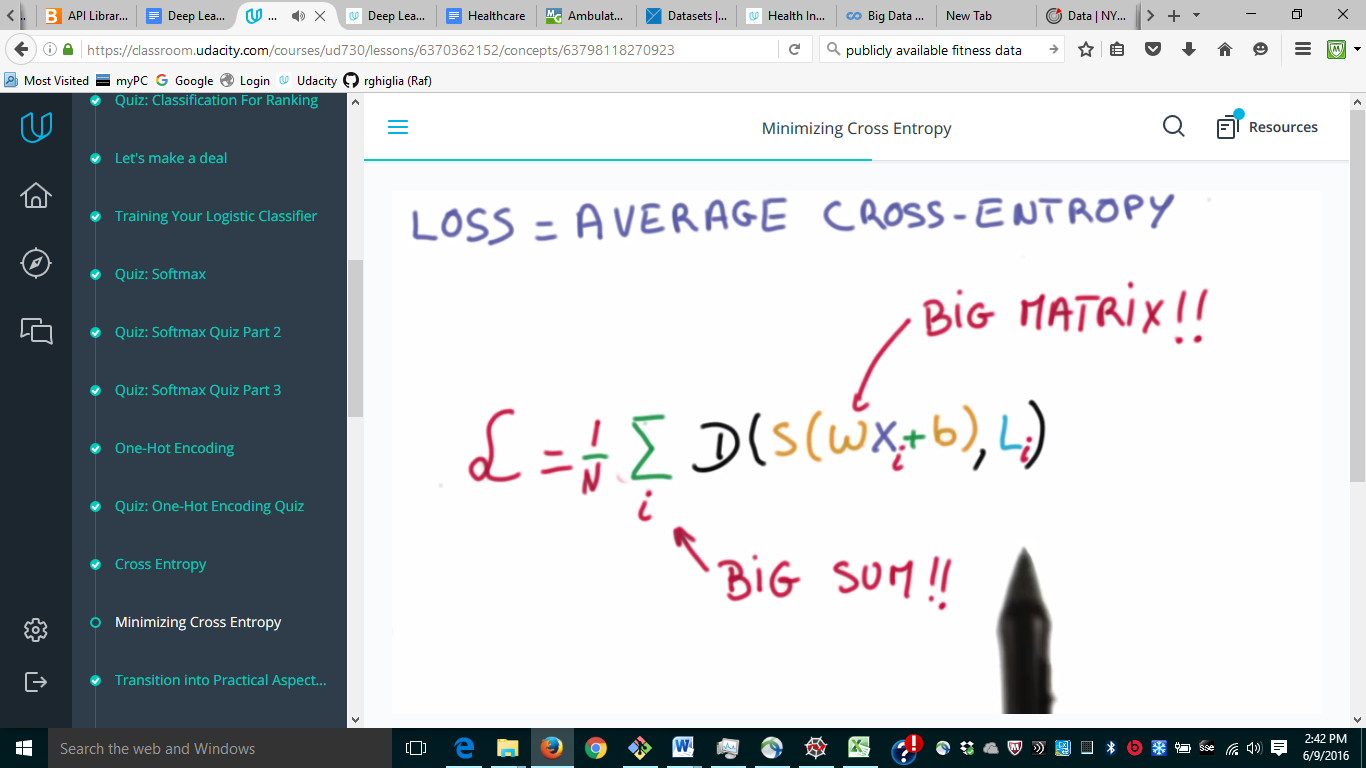
I thought the one-hot encoding was a transformation of S(Y) instead is an independent representation of a label! Thus you want to compare [0.7, 0.2, 0.1] with [1.0, 0.0, 0.0]. Bah, that seams to me artificial extra complication, why not simply take the max of the prob? Assuming each output is identifying a class. If we encoded the output in binary then it would no longer be true. On the other hand we would need to map the scores into an output. Say you have 0.7 prob for output 1 = 2^0 and 0.2 prob for output 2 = 2^1, what binary number does that represent?

Ok, I see you need to be more sophisticated.

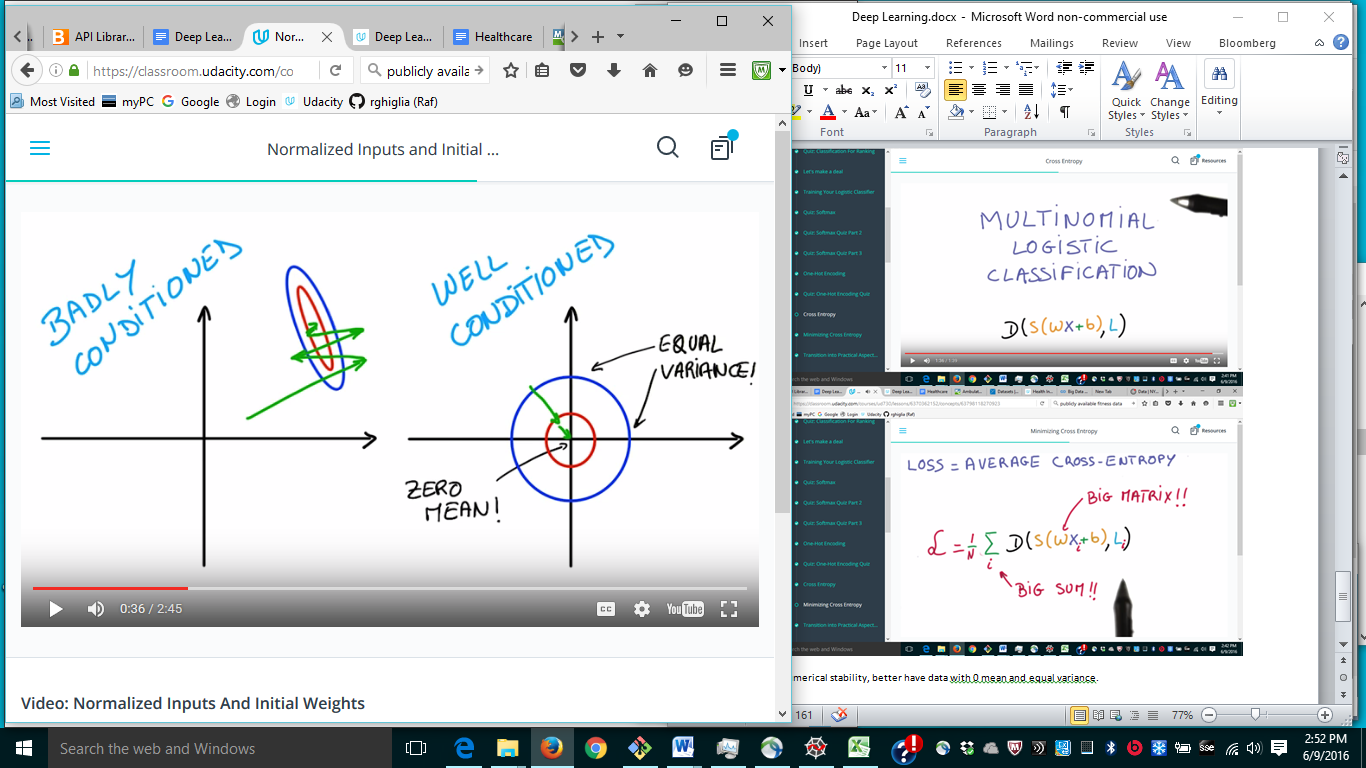




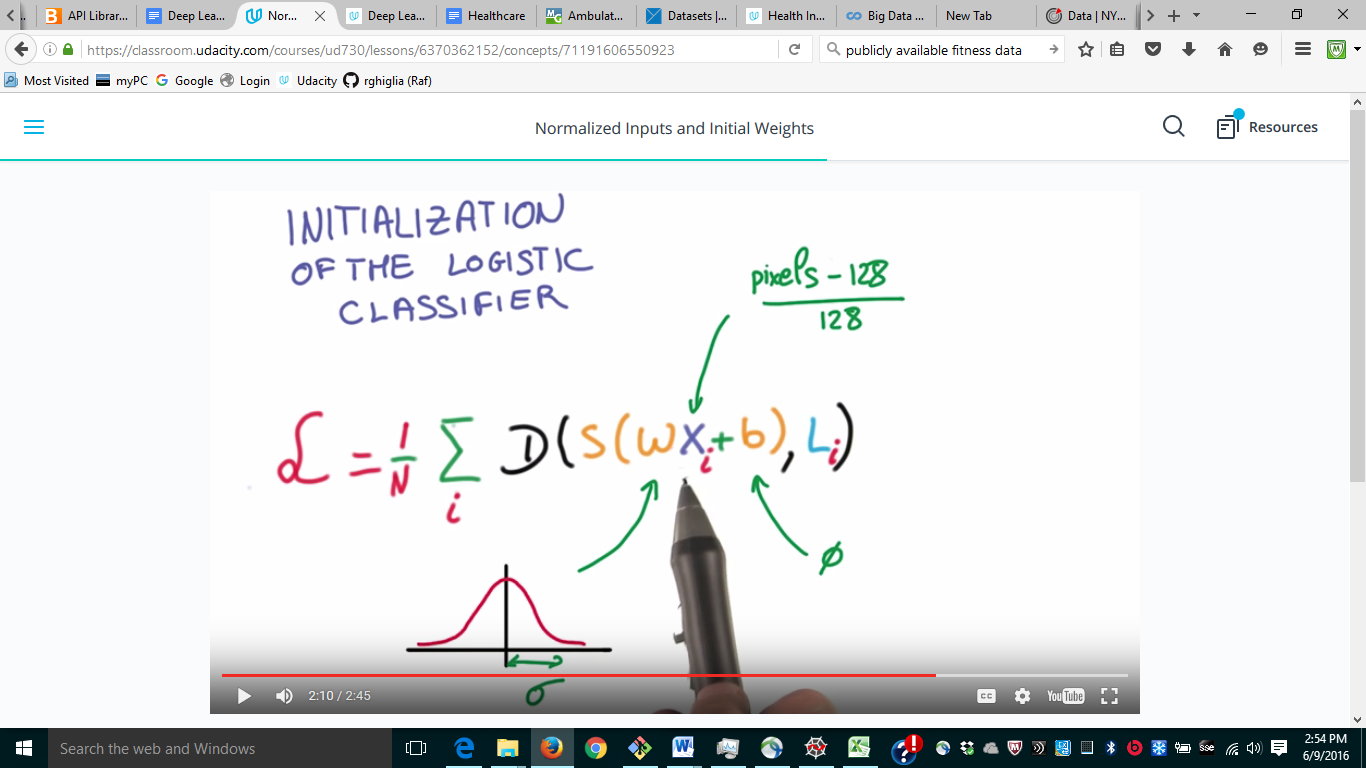




For numerical stability, better have data with 0 mean and equal variance.



Initialize weights at random with small variance so starting distribution is quite agnostic.



I have to get tensor flow … .

This seems particularly complicated … does it even work in windows or it needs to emulate a different OS?

<http://www.netinstructions.com/how-to-install-and-run-tensorflow-on-a-windows-pc/>

<https://github.com/dimchansky/tensorflow-udacity-vagrant>