3.
In the end, I never got my SGD algorithm to work. I suspect it might have something to do with how I derived/implemented my gradient, but after hours of debugging and checking my work, I'm stumped as to where my mistake might be.
Some peculiar behavior:

- My logistic loss actually INCREASES as I run SGD.
- If the step size is too small, then the gradient is always positive
- My test score sometimes dips BELOW 0.5

Anyways, I'm submitting my code anyways because I still spent a fuckload of time implementing it. If you want to take a look, and tell me where I went wrong, or give me pity marks for the scaffolding stuff I did implement, I'd appreciate it?
To run, make sure the training and testing datasets (csv) are in the same directory as **Exercise1.py**, and run **python Exercise1.py**

Some results from briefly letting it run 50 minutes before the deadline (I doubt I have enough time to let it completely finish running against all kernels, lambdas, and sigmas):
linear kernel
lambda = 0
Accuracy: 0.502
lambda = 10
Accuracy: 0.4965
lambda = 20
Accuracy: 0.5
lambda = 30
Accuracy: 0.5
lambda = 40
Accuracy: 0.5
lambda = 50
Accuracy: 0.5
lambda = 60
Accuracy: 0.5