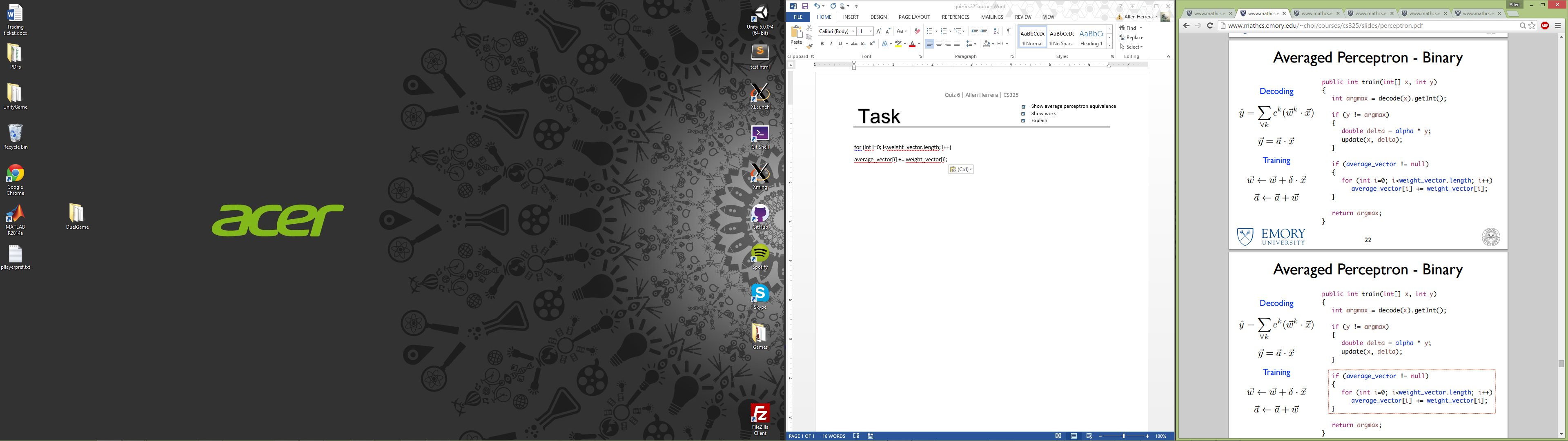
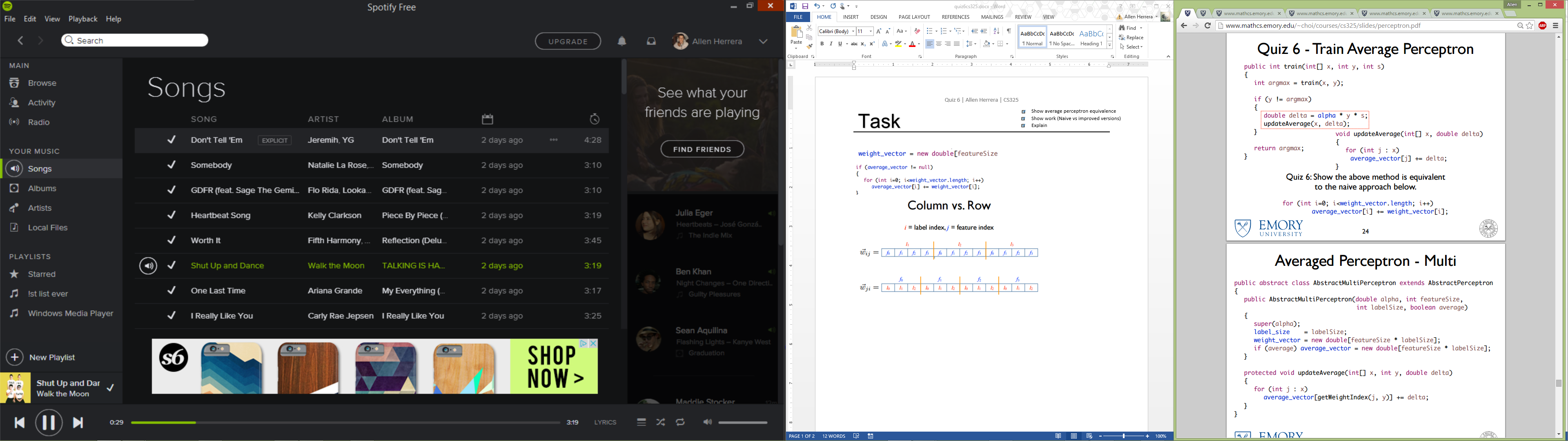
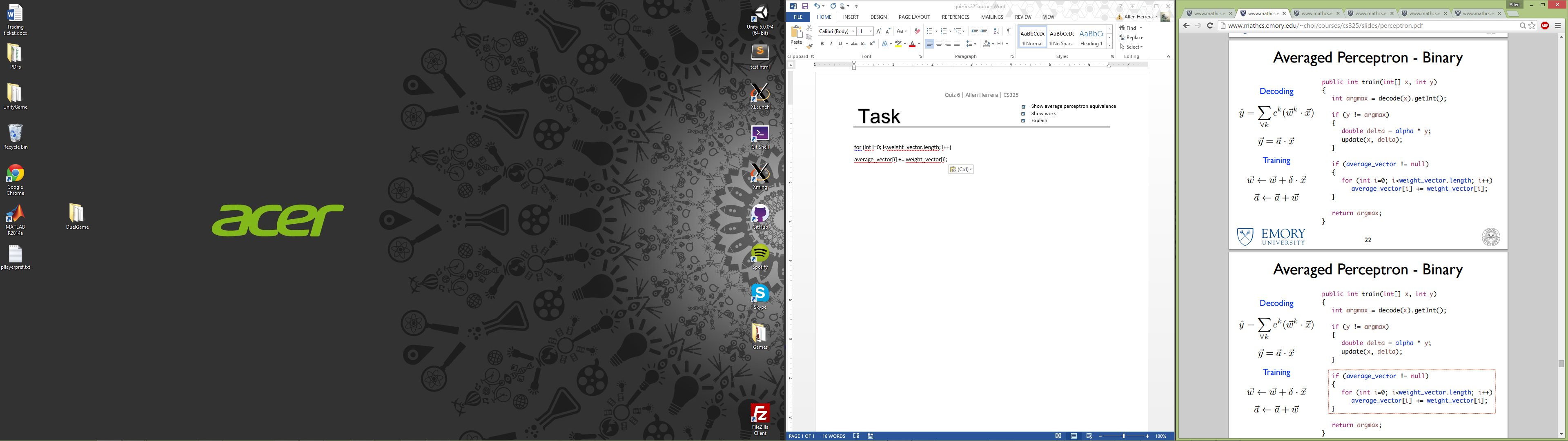


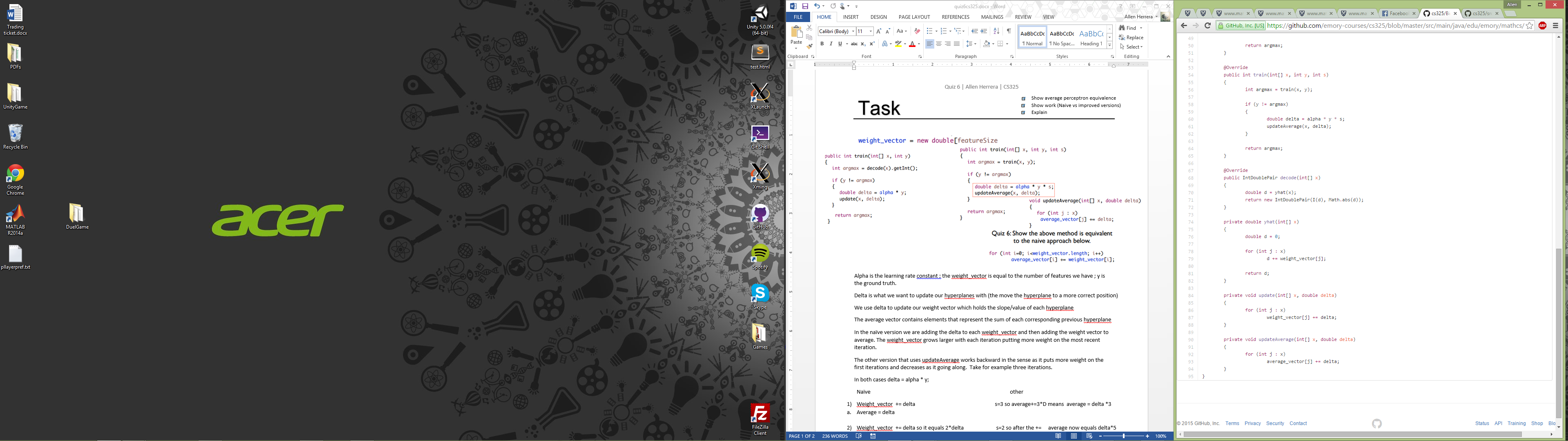
* Show average perceptron equivalence
* Show work (Naive vs improved versions)
* Explain



Task







Alpha is the learning rate constant; the weight\_vector is equal to the number of features we have; y is the ground truth.

Delta is what we want to update our hyperplanes with (the move the hyperplane to a more correct position)

We use delta to update our weight vector which holds the current slope/value of each hyperplane

The average vector contains elements that represent the sum of each corresponding previous hyperplane

In the naïve version we are adding the delta to each weight\_vector and then adding the weight vector to average. The weight\_vector grows larger with each iteration putting more weight on the most recent iteration.

The other version that uses updateAverage works backward in the sense as it puts more weight on the first iterations and decreases as it’s going along. Take for example three iterations.

In both cases delta = alpha \* y;

Naïve other average perceptron

1. Weight\_vector += delta s=3 so average+=3\*D means average = delta \*3
   1. Average = delta
2. Weight\_vector += delta so it equals 2\*delta s=2 so after the += average now equals delta\*5
   1. Average = 3\* delta
3. Weight\_vector += delta again so it equals 3\*D s=1 so after the += average now equals delta\*6
   1. After Average+= Weight vector ; Average = 6\*delta

In the end both come to the same average of delta\*6