Loan Default Prediction:

Model: The first draft of the model involved splitting the training data into two subsets: no loss, and total loss. For each sub set the extremes (top and bottom 5%) were removed and the remainder of data was averaged into an average data point. We did not use some of the data in the training data (to be used for a testing set). A data point would be compared to each average, for each of the roughly 800 features; if it was closer to the no loss it received one point in that feature, if it was closer to the complete loss it lost one point for that feature. If the final tally was above zero, predict no loss; if it were below zero predict complete loss.

1. Ignoring the percentage of loss, the algorithm correctly predicted whether the loan would default or not sixty percent of the time. The numbers produced [-400, 400] seemed to be random and could not be used to determine the percentage of default. Using this model also crashed our computers multiple times (and we have a Haswell processor with 16 GBs of RAM).
2. A lot of the features seem random and appear to not have any correlation with the amount of default. Using less features would make the algorithm faster (and easier to run) but (currently) we do not know of a good way of eliminating features with unsupervised learning.

NCAA Basketball Prediction:

Model: The first draft of the model consisted of using a perceptron and running the seasons A-Q through it to train the perceptron. From there, we randomized the remaining season and ran that through as testing data. Over the course of several iterations, the average and median accuracy was 67- 68%.

1. The model worked decently well, 67-68% of the time. We could not see any reason for when it produced outliers, but the model needs refinement so we can focus on that later. It was quick to perform, taking about a couple seconds to read in the .csv and getting the data processed and producing the result.
2. Feature generation was not too terrible, we used the win percentages of teams as well as the seed they teams were places in for the actual tournament for now. Generating the features took some time as we used Microsoft Excel to calculate win rates when they were not readily available. However, it worked well enough, only resulting in some out of memory problems with Excel.