**Summary**

We were interested in seeing how we could use the power of machine learning to sift through CTECs. Currently, there is an enormous corpus of data from CTECs, which include review scores, actual reviews, and metadata. We hope that by using the correct machine learning algorithms, we can spot trends predicative of high CTEC scores.

Our attributes come from a bag of words of size 1000, where half the words comprise the most common words found in CTECs, and the other half comprise names of professors. Our input is a list of 1000 attributes, where the value is 1 if it exists and 0 otherwise. For our learner, we used a Bayesian Logistic Regression algorithm. We got the best result from this, likely because this learner works well for problems involving bags of words and situations where each attribute may or may not contribute a small amount in predicting the final output (but many attributes may be accurate in aggregate).

We tested and trained our data using Weka. Using a data scraper created by another classmate, Al Johri, we scraped Northwestern’s CTEC database for all CTECs related to the EECS department. We used that data to create a dictionary mapping to the aforementioned bag of words, and used that to generate a bag for each CTEC, outputting our final results in Weka’s ARFF format. Our “class” was whether or not the instructor score for the CTEC was above or below a 5 (we will elaborate more on that later). Finally, we used Weka to try different algorithms until we found one that worked the best. We measured our success in terms of maximizing accuracy and f-score, using the ZeroR classifier as a baseline.

Our final classifier had both an accuracy and f-score of 77.7%, which is not too bad compared to our ZeroR baseline accuracy of 55%. Furthermore, we were able to create a list of words and Professor names, as well as their correlations with higher CTEC scores (both positive and negative). Although this is far from conclusive, we believe that a critical analysis of our results will show that this warrants a more detailed look.

Bag of words – Input 5000 words – Output CTEC Instructor Score > 5 (1) else (0)

* Unable to run Weka without running out of heap space

Bag of words – Input 2500 words – Output CTEC Instructor Score > 5 (1) else (0)

* ZeroR gives 55% accuracy
* Neural net with 0 hidden layers – Couldn’t get it to run in a reasonable amount of time

Bag of words – Input 1000 words – Output CTEC Instructor Score > 5 (1) else (0)

* Naïve Bayes
* Once again, neural nets simply take too long

Bag of words – Input 100 words – Output CTEC Instructor Score > 5 (1) else (0)

* ZeroR – 55 percent
* Naïve Bayes
* BayesianLogisticRegression did pretty well, with a 70 percent classification instance
  + Interesting coefficients: great (0.452), professor (0.009), fun (0.286), learned (0.190), difficult (-0.239), best (0.599), boring (-0.796), awesome (0.910), could (-0.485), didn (-0.606), amazing (0.940), bad (-0.308), project (-0.460), homework (0.004), subject (0.004)
* LogisticRegression, 71 percent
  + Tells us roughly the same thing

Bag of words – Input 100 words – Output CTEC Instructor Score > 4 (1) else (0)

* Concern is that there are not enough negative variables to make it worthwhile
* ZeroR is around 85 percent, and logistic and bayeslogistic perform around the same (naïve bayes does worse)
  + coefficients don’t really yield any new information

Pretty cool so far. It’s clear the Bayesian Logistic Regression is the best performer here, so we are going to sort of build on top of that, so let’s revisit this:

Bag of words – Input 500 words – Output CTEC Instructor Score > 5 (1) else (0)

* + 75 percent accuracy
  + entertaining (0.244),