# Naive Bayes Classifier

Cherubin Manokaran

**Introduction:**

The large amount of user-generated data on social media present researchers with the opportunity to discover patterns in informal language to understand social opinions and gain insight on issues such as cybercrime. An example of this would be predicting gender based on individual posts. Such information would certainly be useful for marketing and legal investigation. For categorical classification purposes, a Naive Bayes approach provides a simple and efficient multiplicative formula for the conditional and class probabilities.

**Methods:**

A classifier was built by constructing features as a matrix with a feature per column and the corresponding labels as a vector for the training data set. The feature values per class were computed by finding the count and then relative occurrence of each. For the text classifier, however, a “Bag of Words” approach was taken where the features were the words in the text. For this reason a list of lists was implemented and the relative occurrence of the unique words in the text was computed.This data was then used to classify a test data set. Again feature matrices were constructed to compute the relative occurrence of each value. The prediction was made by finding the probability for each class,  computed by finding the product of the class probability and each conditional probability. The class with the highest probability of occurring was returned as the prediction. The F1 score, implemented in the *test naive bayes* module, was calculated as a metric of accuracy. With such a prediction task, the probability of false negatives is high. Considering both precision and recall, the score was computed.

To improve classifier performance, feature engineering and smoothing techniques were considered. The text was tokenized by concatenating each list corresponding to a post, removing punctuation, converting to lowercase and removing stop words, such as “a” and “able,” that are generally poor predictors. This enables enrichment of the correct prediction. In addition additive smoothing was performed by increasing every feature value count by one and adding the number of feature values to the denominator.This prevents overfitting and zero probabilities for words absent in the training set from decreasing confidence or accuracy. In addition, the average of this probability and the probability prior to smoothing was taken for values closer to the empirical probabilities. The improved classifier and test module are denoted as *improved* and a csv file containing stop words is also included.

**Results:**
The test for predicting even and odd numbers gave 100% accuracy and F1 scores of 1 for each class, or even and odd. Predictions on the sets of names for each gender based on the supplied features consistently gave >70% accuracy. The confidence score for female predictions was 0.83 while the confidence score for male predictions was 0.69. Finally gender prediction on the set of blogs with a balanced label distribution gave 52-58% accuracy with low confidence for female predictions. With the unbalanced set, accuracy reached 90% but with 0 confidence in the female predictions.

After smoothing and tokenization, accuracy remained in the same range, maybe reaching 64% accuracy at the most for the balanced set. Confidence increased to 0.70 for male predictions and 0.30 for female predictions. Taking the average as described in methods gave more consistent accuracy above 60% and increased confidence to 0.70 and 0.41. With the unbalanced data set, accuracy understably decreased while confidence increased after smoothing and tokenization. With additive smoothing the accuracy dropped to 77% with 0.20 confidence for female predictions and 0.86 confidence for male predictions. Taking the mean dropped accuracy to 64% but increased confidence to 0.27 and 0.77. In the future, supplementing smoothing with bigrams will most likely increase accuracy and confidence.