In my Naive Bayes algorithm, by counting the frequency of each feature in the training data and calculating the percentage rate, I get 100% accuracy in classifying integers, 60-65% accuracy in classifying names, 55%-70% accuracy in classifying the blogs corpus with bag-of-words as features, and 90% accuracy in classifying the imbalanced data set

In order to better classifying the names, I created a specified algorithm that record each feature's position (for example, "Danika" would be recorded as ['d', 'a', 2, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, True, False, False, True, False, False, False, False, True, False, True, False, False, True, False, False, False, False, False, False, False, False, False, False, False, False], and the position of each feature is taken consideration in the algorithm). The accuracy in classifying names thus improved to 70-80%. However, this method is not universal to other tests since in the other tests the position of features is irrelevant.

By using tokenization (a.k.a removing non-alphanumeric characters), the accuracy slightly improved. (60%-70% for classifying the blogs corpus with bag-of-words as features)