The overall goal of the exercise is to get hands-on experience with the implementation of a popular machine learning scheme and to work on a real-world task. The task is to implement an improved version of the Naive Bayes algorithm that is able to predict the domain - one of Archaea, Bacteria, Eukaryota or Virus - from the abstract of research papers about proteins taken from the MEDLINE database. You will then apply your implementation on a test set without class labels and hand in the predictions of your implementation.

The challenge

The ultimate goal of this programming project is to come up with an implementation of a (possibly extended or modified) Naive Bayes algorithm, that achieves a high predictive accuracy on the test data. As a minimum requirement your implementation should be better than a vanilla-plain version of Naive Bayes as explained in the lecture or any standard textbook. Your algorithm should still be "Naive Bayes" in the sense that it makes the assumption that all attributes are conditionally independent of each other given the class. You might, however, change any other assumptions, representations or models in your implementation. Basically, there are two parts to be solved:

1. First of all, you need to decide about a suitable representation for the text in the abstract. An easy way to obtain an attribute-value representation is to identify the 1000 most frequently occurring words and generate 0-1 attributes stating whether or not the word occurs in the corresponding abstract. There are other possible representations, e.g. one could take the occurrence frequency of a word in the abstract into account.
2. The standard Naive Bayes algorithm as outlined for example in Mitchell's "Machine Learning" will probably yield comparably poor predictive accuracy, so you need to improve it in order to obtain good predictive accuracy.

You have to implement from scratch the data preprocessing, the Naive Bayes algorithm, model validation (cross-validaton), and model performance metrics (you can implement the over-all accuracy as default metric). This means you are not allowed to use already existing Naive Bayes implementations or use packages for data preprocessing, ML or NLP tasks.

-
  - This task has to be implemented with the core functionality of Python. You can use numpy to represent data as arrays.
  - Prepossessing is just basic string manipulation (Python has great core function for this), so you do not need any NLP or other advanced packages/libraries for your extensions. The task is designed for you to stay within the scope of the material covered in the lectures, but still be able to propose some extensions. For motivation, you can also look at Rennie at al. (2003) "Tackling the Poor Assumptions of Naive Bayes Text

Classifiers" (Links to an external site.) that lists several extensions.