M1 Informatique Practice in Machine Learning

I Naive Bayes

During this practice, we are going to consider a dataset which is composed of course reviews together with associated grades, coming from Coursera. It is available on the Madoc platform (`reviews_by_course.csv`). An easy way to load this data is to use the pandas library, with `data = pd.read_csv('reviews_by_course.csv')`

1) With the help of `pandas` tools, identify the volume and the dimension of the data.

2) Target classes are the ratings, from 1 to 5. Determine the number of instances for each class. What could be the problem during the learning process ?

3) Each instance consists in a short paragraph of text, containing the review, and the course tag. Consequently, one first step is to transform the text into a description that we can use for classification. The following code, using the scikit library,

```
from sklearn.feature_extraction.text import CountVectorizer
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(corpus).toarray()
```

allows to create an array where each column represent a word of the `corpus` data, and each paragraph is then described by the count of each of its words (a.k.a. bag of words). Tranform the data so that each tag and each review is described by this approach. (In the case of the course tag, it will be a binary descprtion, not a count).

4) The volume of the data could be quite (too) large, depending on your computer. Together with your answer of Question 2), and using the functions `head()`, generate a meaningful subset of the data.

5) Using a naive Bayes approach, we are going to learn model whose objective is to predict the rating, given the tag and the review. What kind of probabilistic distribution should we use ?

6) Evaluate the performance of your model with cross validation (use `cross_val_score` from sklearn package).

7) The NB model has different parameters. Make a plot of the accuracy of the model as a function of α, the smoothing parameter.

II Going further ...

In the first part of this practice, we transformed the text with a rather naive bag of words approach. There are other solutions, such as TF-IDF [1], which is implemented in sklearn package. Is this description of the text improves the quality of the classifier ?

1. `https://scikit-learn.org/stable/modules/feature_extraction.html`