Homework 2:
Naive Bayes

Dr. Benjamin Roth
Computerlinguistische Anwendungen

Due: Wednesday May 15, 2019, 14:00

In this exercise we will implement a Multi-Class Naive Bayes Classifier that will be trained with the 20 Newsgroup Dataset to disinguish 20 different text categories.
Take a look at the file `hw02_naive_bayes/text_categorization.py`. In this exercise you will have to complete some methods to make the classification work. Get the code for this exercise from your team git project (use `git pull`).

To install sklearn: `pip3 install sklearn`
To test your code: `python3 -m unittest -v hw02_naive_bayes/test_naive_bayes.py`

Exercise 1: Creating the instances [0 points]

Complete the method `DataInstance.from_list_of_feature_occurrences(...)`. This is the same as from the last homework.

Exercise 2: Constructing/training the Classifier [4 points]

Complete the classmethod `NaiveBayesClassifier.for_dataset(cls, dataset, smoothing = 1.0)`. To do so, you should be familiar with the python `@classmethod` idea. The method should serve as a constructor to construct a `NaiveBayesClassifier` from a `Dataset`.

Exercise 3: Predicting [6 points]

Complete the method `prediction(self, feature_counts)`. This method should return the predicted class label (a string). You need to understand the method `log_probability` first.

Exercise 4: Evaluating [4 points]

Complete the method `prediction_accuracy(self, dataset)`. This method should iterate over a labelled `Dataset`, predict labels for all samples and return the *Accuracy*.

Exercise 5: Finding the best features [6 points]

Complete the method `log_odds_for_word(self, word, category)` that computes the log-odds $\log\left(\frac{P(category|word)}{1-P(category|word)}\right)$.

Exercise 6: Using the classifier [bonus]

Once you have implemented all missing functionality, you can have a look at `text_categorization.py` to see how to use naive bayes in practice. Run the code with:
`python3 -m hw02_naive_bayes.text_categorization`
Info: Download server might be slow.