Homework 11:
Naive Bayes

Robert Litschko*
Symbolische Programmiersprache

Due: Thursday, January 26, 2022, 12:00 (noon)

In this exercise we will implement a Multi-Class Naive Bayes Classifier that will be trained with the 20 Newsgroup Dataset to distinguish 20 different text categories. The dataset can be imported as part of the library sklearn, so you don't need to get the dataset somehow else. Take a look at the file `hw11_naive_bayes/naive_bayes_classifier.py`. In this exercise you will have to complete some methods to make the classification work. The implementation uses the external python module `sklearn`[1]. You should already have installed sklearn for previous exercises; if not, run: `pip3 install sklearn`

This homework will be graded using unit tests by running:

```
python3 -m unittest -v hw11_naive_bayes/test_naive_bayes.py
```

Exercise 1: Creating the instances [1 point]

Complete the @classmethod `DataInstance.from_list_of_feature_occurrences(cls, feature_list, label)`. The method should create a dictionary `feature_counts`, which contains all words of the feature_list and their frequency. To implement this dictionary, you can also use the Counter class we learnt in the last session. The created `feature_counts` is then passed to the cls() method, which calls the constructor of the class and constructs a `DataInstance` from a `feature_list`.

Exercise 2: Constructing/training the Classifier [3 points]

Complete the classmethod `NaiveBayesClassifier.for_dataset(cls, dataset, smoothing = 1.0)`. The method should go through each data instance of the `instance_list` and fill out the two dictionaries defined in the method. Then, the method can serve as a constructor to construct a `NaiveBayesClassifier` from a `Dataset`.

*Credit: Exercises are based on previous iterations from Katerina Kalouli.
[1] `https://sklearn.org/`

Exercise 3: Predicting [6 points]

Complete the method `prediction(self, feature_counts)`. This method should return the predicted class label (a string) for the given `feature_counts`, i.e., for the given data instance. You need to understand the method `log_probability` and `score_for_category` first. You will also need to use the method `score_for_category` to score each category and return the one with the highest score.

Exercise 4: Evaluating [4 points]

Complete the method `prediction_accuracy(self, dataset)`. This method should iterate over all data instances of a given `dataset`, predict the labels for each dataset (use the previously implemented method for that) 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: $\log\left(\frac{P(category|word)}{1-P(category|word)}\right)$. Have a look at the slides to get some help.

Exercise 6: Using the classifier [just for fun]

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. This may take a while depending on the power of your computer. Run the code with:
`python3 -m hw11_naive_bayes.text_categorization`