CSC475/575 MIR Assignment 3, Spring 2017 (10 pts)

The goal of this assignment is to familiarize you with data mining/machine learning in the context of music information retrieval.

Hope you find it interesting, George Tzanetakis

1 Classification using Audio Features (4 points)

Download the 1.2 GB genre classification dataset from:
`http://marsyas.info/downloads/datasets.html`
You will only need 1.2 GB of space for download but after that you can pick any three genres out of the 10 genres for your experiments. Alternatively if you don't have enough space you can download individual files for 3 genres (at least 20 tracks for each genre) from:
`http://marsyas.cs.uvic.ca/sound/genres/`
Read the instructions in Chapter 3 of the Marsyas User Manual (Tour - Command Line Tools) and use the **bextract** command-line program to extract features for the 3 genres you selected. Load the extracted .arff file into Weka and report on the classification accuracy of the following classifiers: ZeroR, NaiveBayesSimple, J48, and SMO.

Your deliverable will be the list of command you used and the classification accuracy + confusion matrix for each classifier for the 3-genre experiment. (**) **(2 points)**

Now use Weka to convert the .arff to the .libsvm format that is supported by scikit-learn. Do a similar experiment using scikit-learn i.e 3 classifiers and report accuracy and confusion matrix. Provide a listing of the relevant code (**) **(2 points)**.

2 Music Genre Classification Via Naive Bayes and Lyrics Text (6pts/3pts)

Text categorization is the task of assigning a given document to one of a fixed set of categories, on the basis of text it contains. Naive Bayes models are often used for this task. In these models, the query variable is the document category, and the "effect" variables are the presence/absence of each word in the language; the assumption is that words occur independently in documents within a given category (conditional independence), with frequencies determined by document category.

Our goal will be to build a simple Naive Bayes classifier for the MSD dataset, which uses lyrics to classify music into genres. More complicated approaches using term frequency and inverse document frequency weighting and many more words are possible but the basic concepts are the same. The goal is to understand the whole process, so do not use existing machine learning packages but rather build the classifier from "scratch".

We are going to use the musicXmatch [1] dataset which is a large collection of song lyrics in bag-of-words format for some of the trak IDS contained in the Million Song dataset (MSD). The correspondent genre annotations, for some of the song in the musicXmatch dataset, is provided by the MSD Allmusic Genre Dataset [2]. For the purpose of this course, in order to simplify the problem, we are going to use a reduced version of the musicXmatch dataset. Three genres are considered, namely: "Rap", "Pop_Rock", and "Country". The resulting genre annotated dataset is obtained by an intersection of musicXmatch and MAGD, where we select 1000 instances of each genre, such that the three classes are balanced and easy to handle. In addition, we also reduce the cardinality of the dictionary of words used for the bag-of-words lyrics representation (originally equal to 5000), to the 10 *best* words for each genre. Intuitively, the best words are the most frequent words for a particular genre that are not frequent among all the genres [3].

The resulting dictionary of the three genres is:

```
['de',   'niggaz', 'ya',    'und',   'yall',  # rap
 'ich',  'fuck',   'shit',  'yo',    'bitch', # rap
 'end',  'wait',   'again', 'light', 'eye',   # rock
```

[1] https://labrosa.ee.columbia.edu/millionsong/musixmatch

[2] http://www.ifs.tuwien.ac.at/mir/msd/partitions/msd-MAGD-genreAssignment.cls

[3] The best genre words maximize the Term Frequency (TF) and Inverse Document Frequency (IDF) product. More details available at https://en.wikipedia.org/wiki/Tf-idf

```
'noth',   'lie',    'fall',   'our',    'away',   # rock
'gone',   'good',   'night',  'blue',   'home',   # country
'long',   'littl',  'well',   'heart',  'old']    # country
```

An additional simplification of the problem is to consider just the presence or absence of a particular word, instead of the frequency count. Therefore according to this problem setup, the feature vector of the song `TRAAAHZ128E0799171` is [0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 1, 0]

For answering this question we provide you with:

- `data.npz` – the three genres dataset (not binarized)
- `labels.npz` – the genre labels where Rap=12, Pop_Rock=1, and Country=3)
- `dictionary.pck` – the full 5000 words dictionary
- `words.pck` – the 30 best word indexes with respect to the full dictionary
- `tracks.pck` – the track IDs of songs used.

. These data is available in either Python pickle format (`*.pck`), or NumPy format (`*.npz`) and can be found at: `http://marsyas.cs.uvic.ca/csc475_asn3_data.tar.gz`

- (**) Write code that calculates the probabilities for each dictionary word given the genre. For the purposes of this assignment we are considering only the tracks belonging to the three genres: Rap, Rock pop, Country (1pt, 0.5pt)
- (**) Explain how these probability estimates can be combined to form a Naive Bayes classifier. Calculate the classification accuracy and confusion matrix that you would obtain using the whole data set for both training and testing partitions. (1pt, 0.5pt)
- (**) Read the Wikipedia page about cross-validation in statistics [4]. Calculate the classification accuracy and confusion matrix using the $k-$fold cross-validation, where $k = 10$. Note that you would use both the training and testing data and generate your own splits. (2pt, 1pt)

[4] `https://en.wikipedia.org/wiki/Cross-validation\_(statistics)`

- (***) One can consider the Naive Bayes classifier a generative model that can generate binary feature vectors using the associated probabilities from the training data. The idea is similar to how we do direct sampling in Bayesian Networks and depends on generating random number from a discrete distribution (the unifying underlying theme of this assignment). Describe how you would generate random genre "lyrics" consisting solely of the words from the dictionary using your model. Show 5 examples of randomly generated tracks for each of the three genres: Rap, Rock pop, and Country; each example should consist of a subset of the words in the dictionary. (2pt, 1pt)

3 Reading (ONLY FOR CSC575 STUDENTS) (3 points)

Read the paper "Semantic Annotation and Retrieval of Music and Sound Effects"
(You can find it on Google Scholar) which is one of the first published works exploring automatic music tagging and answer the following questions:

1. (**) The authors describe three parameter estimation techniques: direct estimation, model averaging, and mixture hierachies estimation. For which category of tags does model averaging work better than mixture hiearchies ? Which category was modeled most successfully using the proposed approach ?

2. (**) Describe the differences between the music data set and the sound effect dataset. For which one does auto-tagging seem to work better ? Speculate why this is the case.

3. (***) Read about the EM-algorithm and how it can be used for estimating the parameters of a Gaussian mixture model. Prepare a short (2-3 paragraphs) description targeted to undergraduate students for helping them understand how it works. Try to go beyond a dry mathematical description using one or more of the following techniques: 1) code example, 2) animation, 3) static visualization, 4) concrete specific example.