**PROPOSAL**

**Project Title:** Affective Music Information Retrieval

**Project Summary:** In our project we categorize songs according to their moods and emotions. Therefore, we use the lyrics of a song. With different term-weighting schemes we classify songs of a public available dataset (the million songs dataset) into different moods. At the end we test the accuracy of the classification.

In our term project we will replicate the paper "An Exploration of Mood Classification in the Million Songs Dataset", Humberto Corona and Michael P. O'Mahony, In proceedings of the Sound and Music Computing conference, Maynooth, Ireland, 2015.

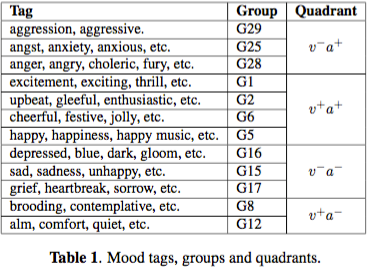
**Prepared by:** Büşra Demir and Alena Beyer

**Project Contacts:**

[bsrnrdmr@gmail.com](mailto:bsrnrdmr@gmail.com): Büşra Demir

[alena.beyer@gmail.com](mailto:alena.beyer@gmail.com): Alena Beyer

Project Methodology:

* In this project we are going to use Java as the programming language.
* First we build our dataset out of three different public available datasets *(the MIREX 2009 mood multi-tag dataset, the LastFm dataset and the MusixMatch dataset)* using the same criteria as in the MusixMatch dataset. Therefore three levels of granularity are defined: tags, mood groups and mood quadrant. We take songs into our dataset if a song is tagged at least twice with one tag or at least tagged with two tags of the same group (there exists different mood groups). In this step we already delete duplicates.  
  Moreover we only consider english lyric and erase all non-english lyric in this step.
* Next we do some statistics by calculating the term distribution for each document and classes (mood quadrants) by using five different term weighting schemes: a binary term weighting, the term-frequency (tf), inverted document frequency (tf-idf), BM25, and delta tf-idf. Therefore we use the vector-space-model and match every document to a vector d=(t1…tn). Moreover we use stemmed terms. Annotation: we don’t analyze the distribution over tag groups and mood tags as it is not done in the paper.
* Our goal is to get similar results concerning distribution to the analysis in the paper. Therefore, we compute the number of distinct terms, unique terms and number of unique terms per song and compute the standard deviation for them.
* In the next part we do the mood classification by comparing the performance of the five different term weighting schemes (see above) for all the three granularities (see above). With the Weka machine learning framework[[1]](#footnote-2) and the LIBLinear L2-SVM classification algorithm[[2]](#footnote-3) (both libraries available in Java), as Corona and O’Mahony did, we create 1000 positive and 1000 negative songs as training sets for each class (either mood group or tag or mood quadrant). With Weka and LibLinear we classify them.
* By using accuracy metric:

(TP: true positives, TN: true negatives, FP: false positives, FN, false negatives) we evaluate the classification performance

* And by making a Kruskal-Wallis[[3]](#footnote-4) test at the level 0.05 we evaluate whether there are significant statistical differences between the term evaluation scheme.
* Finally we compare our results with the results of Corona and O’Mahony and will conclude whether their paper was replicable or not.

1. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reute- mann, and I. H. Witten, “The WEKA Data Mining Software: an Update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009. [↑](#footnote-ref-2)
2. R. Fan, K. Chang, and C. Hsieh, “LIBLINEAR: A Li- brary for Large Linear Classification,” *The Journal of Machine Learning*, vol. 9, pp. 1871–1874, 2008 [↑](#footnote-ref-3)
3. E. Theodorsson-Norheim, “Kruskal-Wallis test: BA- SIC computer program to perform nonparametric one- way analysis of variance and multiple comparisons on ranks of several independent samples.” Computer Methods and Programs in Biomedicine, vol. 23, no. 1, pp. 57–62, 1986. [↑](#footnote-ref-4)