**“LYRICS CLASSIFICATION”**

**PROJECT REPORT**

Submitted for CAL in B. Tech Natural Language Processing (CSE4022)

By

**MIRTUNJAY KUMAR (16BCE1179)**

**ADARSH DUBEY (16BCE1338)**

**Slot: A2**

**Name of the faculty: Dr. BHARADWAJA KUMAR**

**(SCHOOL OF COMPUTING SCIENCES AND ENGINEERING)**

**(SCSE)**



**ABSTRACT**

Sentiment prediction of contemporary music can have a wide-range of applications in modern society, for instance, selecting music for public institutions such as hospitals or restaurants to potentially improve the emotional well-being of personnel, patients, and customers, respectively.

In this project, music classification system built upon on a machine and deep learning based classifier, trained to predict the sentiment of songs based on song lyrics alone. The experimental results show that music corresponding to a happy mood can be detected with high precision based on text features obtained from song lyrics.

**INTRODUCTION**

With the rapid growth of digital music libraries as well as advancements in technology, music classiﬁcation and recommendation has gained increased popularity in the music industry and among listeners. Many applications using machine learning algorithms have been developed to categorize music by instruments artist similarity, emotion, or genre. Psychological studies have shown that listening to music is one of the most popular activities in leisure time and that it has an enhancing effect on the social cohesion, emotional state, and mood of the listeners. The increasing number of song lyrics that are freely available on the Internet allow the effective training of machine learning algorithms to perform mood prediction and ﬁltering for music that can be associated with positive or negative emotions. The aim of this project was to build a recommendation system that is able to predict whether a song is happy or sad, which can be applied to song databases in order to select music by sentiment in different social contexts.

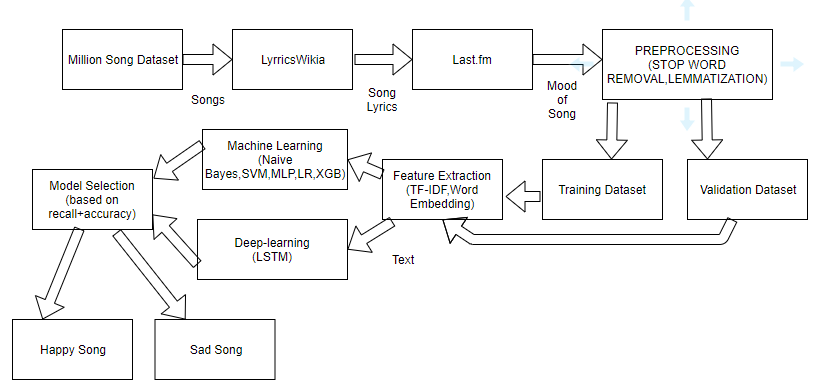


Figure ARCHITECTURE DIAGRAM OF CLASSIFICATION SYSTEM

**DATASET**

A random subsample of 10,000 songs was downloaded from the Million Song Dataset format. Using the provided song title and artist information from these HDF5 ﬁles, custom code was written to download the corresponding lyrics from LyricWikia. Songs for which lyrics were not available — songs that are either instrumental or not deposited in the LyricWikia database — were removed from the dataset. Custom code based on the Python NLTK library was written to identify non-English lyrics andremovethesesongsfromthedatasetusingmajoritysupportbasedonthecountsofEnglishwords vs. non-English words in the lyrics. After applying those ﬁltering rules, the remaining dataset of 2,773 songs was randomly partitioned into a training dataset (1,000 songs) and a validation dataset (200 songs). Music labels were automatically collected from user-provided content on the music database Last.fm [1]. However, due to the nonexistence of mood-related tags for a majority of songs in the ﬁltered dataset, the two mood labels (happy and sad) were manually assigned based on human interpretation of the lyrics and listening tests. Happy music was deﬁned as music that could be associated with upbeat sounds and positive themes. Sad music was deﬁned as music that the author related to a negative, dark, or violent theme.



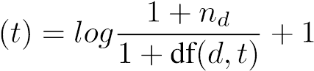
**FEATURE EXTRACTION**

Prior to the tokenization of the lyrics, a bag of words model [8] (a fixed-size multiset where the order of words has no significance) was used to transform the lyrics into feature vectors. Further processing of the feature vectors includes the choice of different n-gram sequences (*n* ∈ {1*,*2*,*3}), stop word removal based on a stop word list from the Python NLTK library [5], and usage of the Porter stemming algorithm [20] for suffix stripping. Also, different representations of the word count in the feature vectors for each song text were used, such as binarization, term frequency (*tf*) computation, and term frequency-inverse document frequency (*tf-idf*) computation.

The term frequency-inverse document frequency was calculated based on the normalized term frequency tf-idf(*t,d*), which is computed as the number of occurrences of a term *t* in a song text *d* divided by the total number of lyrics that contain term *t*

tf-idf(*t,d*) = tf(*t,d*) × idf(*t*)*.* (1)

Let tf-idf(*t,d*) be the normalized term frequency and idf(*t*) be the inverse document frequency

idf*,*

where *nd* is the total number of lyrics and *df*(*d,t*) the number of lyrics that contain the term *t*.

**Word Embedding:**

Word embeddings are in fact a class of techniques where individual words are represented as real-valued vectors in a predefined vector space. Each word is mapped to one vector and the vector values are learned in a way that resembles a neural network, and hence the technique is often lumped into the field of deep learning. Key to the approach is the idea of using a dense distributed representation for each word. Each word is represented by a real-valued vector, often tens or hundreds of dimensions. This is contrasted to the thousands or millions of dimensions required for sparse word representations, such as a one-hot encoding.

**MODEL SELECTION**

Model performances using different combinations of the feature, as mentioned earlier, preprocessing techniques including hyper parameter optimization of the naive Bayes models were evaluated using grid search and 10-fold cross-validation on the 1000-song training set to optimize the F1-score. Defining the mood label *happy* as the *positive* class, the F1-score was computed as the harmonic mean of precision and recall

precision × recall

F1 = 2 × *,* (2)

precision + recall

where

TP

precision = (3)

TP + FP

and

TP

recall = *.* (4)

TP + FN

(TP = number of true positives, FP = number of false negatives, and FN = number of false negatives.)

**RESULT AND DISCUSSION**

After manual assignment of the mood labels and random sampling, the training dataset consisted of happy (44.6%) and sad (55.4%) songs; the number of happy and sad songs in the validation dataset was equal (Table 1). The model selection was performed via grid search and 10-fold cross- validation on the 1000-song training dataset to optimize the performance measured via F1-score. The final model was trained on the entire training dataset, the performance was evaluated on the 200-song validation dataset by measuring the receiver operating characteristic area under the curve (ROC auc), accuracy, precision, recall, and F1-score.

For initial model selection, grid search was performed on three separate naive Bayes models to select the best performing combination of feature extraction and selection approaches and parameters for each model. These three models were Multi-variate Bernoulli Bayes with binary word counts as feature vectors, multinomial Bayes with term frequency features, and multinomial naive Bayes with tf-idf features. After the three models had been individually optimized via grid search, the performance of the best performing model, from each of the three categories, was evaluated via ROC auc. The best performing model was then chosen for a more thorough optimization via grid search. During the grid search optimization, the following settings and parameters were optimized: n-gram rangefortokenization, stopwordsremoval, Porterstemming, themaximumnumberoffeaturesinthe vocabulary (based on the k most frequent tokens), a cut-off for minimum term frequency, and the α smoothing parameter.

Table 1: Mood label distribution in the training and validation datasets.

|  |  |  |  |
| --- | --- | --- | --- |
| Mood | Training | Validation | Total |
| happy | 446 | 95 | 541 |
| sad | 554 | 95 | 649 |

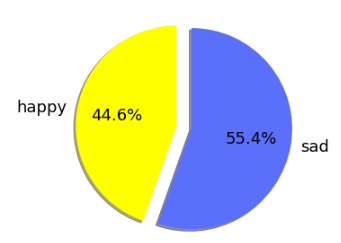


Figure : SONG DISTRIBUTION IN DATASET

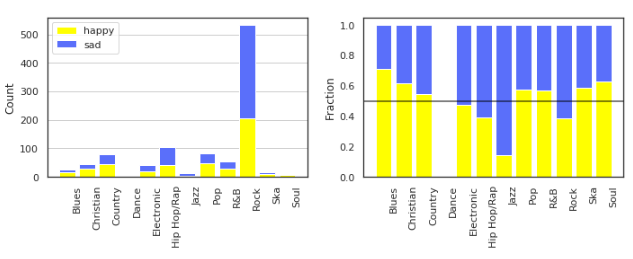


Figure : YEARWISE HAPPY SAD DISTRIBUTION OF SONG



Figure :positive WORD DISTRIBUTION WORD -CLOUD

MACHINE LEARNING CLASSIFIER RESULT ON DIFFERENT FEATURE VECTOR:

1. COUNTVECTORIZER

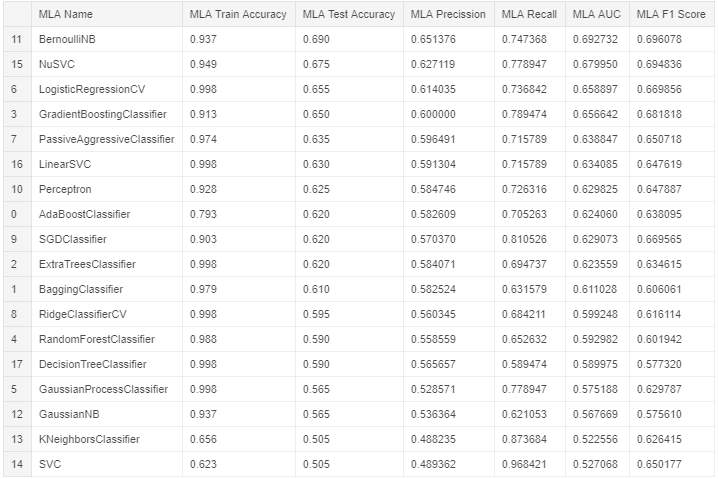
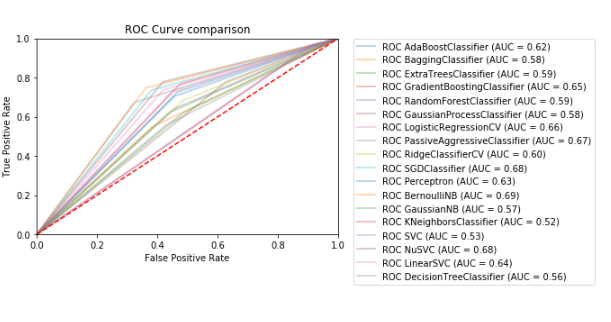


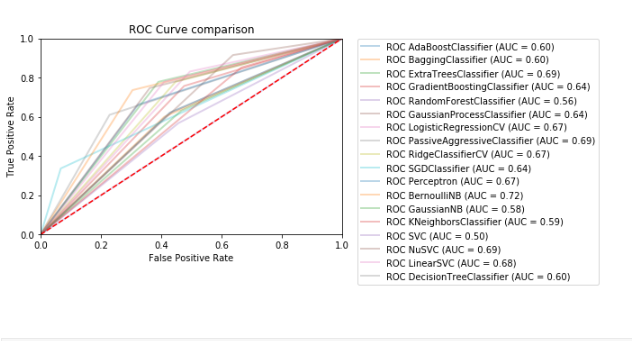
Figure : ACCURACY FOR COUNTVECTORIZER



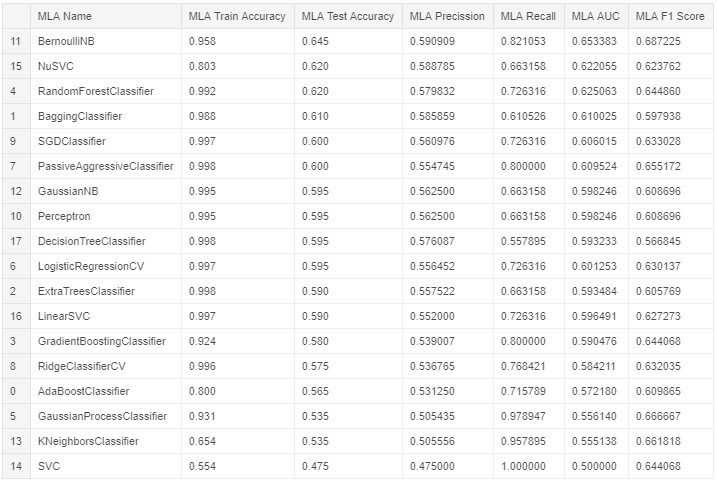
TF-IDF AS FEATURE VECTOR:



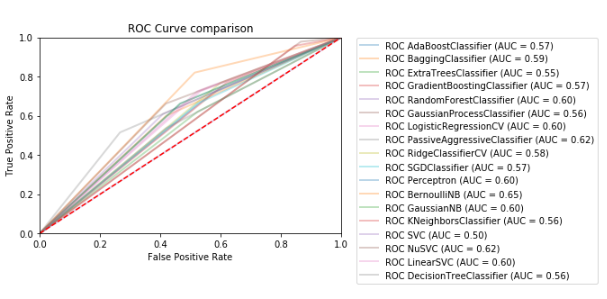
ROC CURVE:



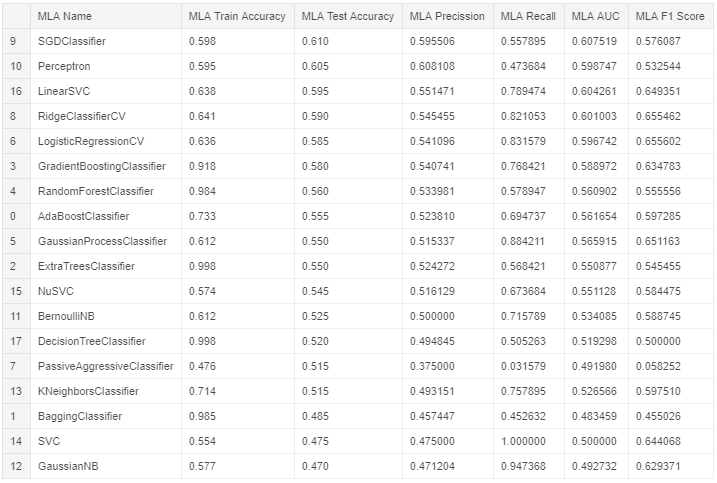
TF-IDF WITH NGRAM AS FEATURE VECTOR



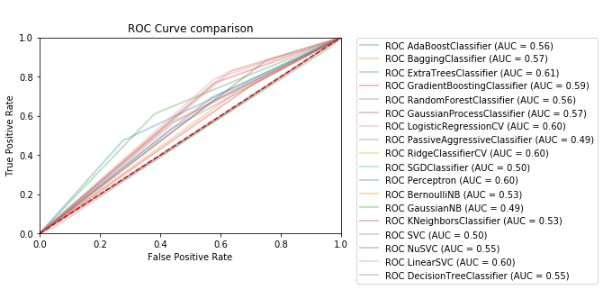
ROC CURVE



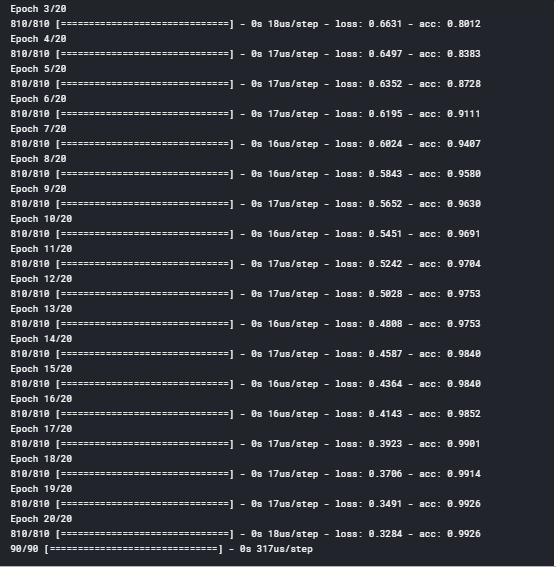
TF-IDF NGRAM WITH CHARACTER LEVEL



ROC CURVE



WORD MEBEDDING AS FEATURE USING KERAS SEQUNTIAL MODEL



**CONCLUSION**

The results have shown that a naive Bayes model applied to mood classiﬁcation based lyrics can predict the positive class (happy) with high precision, which can be useful to ﬁlter a large music library for happy music with a low false positive rate. A music library ﬁltered in this manner could further be used as input for genre classiﬁcation to ﬁlter music according to different tastes. Planned future work will include extensions to the mood classiﬁcation web application to incorporate more. The results have shown that a naive Bayes model applied to mood classiﬁcation based lyrics can predict the positive class (happy) with high precision, which can be useful to ﬁlter a large music library for happy music with a low false positive rate. A music library ﬁltered in this manner could further be used as input for genre classiﬁcation to ﬁlter music according to different tastes. Planned future work will include extensions to the mood classiﬁcation web application to incorporate more lyrics to evaluate if the predictive performance of the classiﬁer can be improved given a larger dataset. The extensions will include feedback about the prediction. In one extension, online learning will be implemented to update the hypothesis incrementally.

**REFERENCES**

1. musicXmatch - \url{http://labrosa.ee.columbia.edu/millionsong/musixmatch}.
2. Steven Bird. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.
3. Pedro Domingos and Michael Pazzani. On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, 29(2-3):103–130, 1997.
4. Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, pages 1–12, 2009.
5. Zellig S Harris. Distributional structure. *Word*, 1954.
6. Sundus Hassan, Muhammad Rafi, and Muhammad Shahid Shaikh. Comparing SVM and Naive Bayes classifiers for text categorization with Wikitology as knowledge enrichment. In *Multitopic Conference (INMIC), 2011 IEEE 14th International*, pages 31–34. IEEE, 2011.
7. Perfecto Herrera, X. Amatriain, E. Batlle, and Xavier Serra. Towards Instrument Segmentation for Music Content Description a Critical Review of Instrument Classification Techniques. *International Conference on Music Information Retrieval*, 2000.
8. Yajie Hu and Mitsunori Ogihara. Genre classification for million song dataset using confidence-based classifiers combination. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1083–1084. ACM, 2012.