MULTIMODAL MOOD DETECTION BASED ON LYRICS AND AUDIO

Kenil Shah
V00903842
University of Victoria

ABSTRACT

The atmosphere of a song is largely conveyed through musical elements, but the lyrics which correlate to a song contribute a significant amount of information and have a crucial role in dissecting the meaning of a song as well, and understanding the mood of music based on these factors has a lot of potential for solving problems like genre identification and suggestion. In this project we will conduct a study into multimodal music emotion recognition, and strive towards designing a program which can accurately classify the emotion based on a predetermined emotion paradigm. The data which this program will consider as input for classification mainly includes a given song and its lyrics, as well as the corresponding MIDI file, whose possible inclusion will be determined as we further investigate its benefits and disadvantages. Moreover, this exploration of the music emotion recognition (MER) problem will utilize the multi-modal Music Information Retrieval Evaluation eXchange (MIREX) like emotion dataset - based on the emotion tags used in the MIREX Mood Classification Task.

1. INTRODUCTION

Since its conception, music has been a prime tool used for emotional expression, comprehension, and navigation. In recent years, the surge of music platforms and the increasingly extensive content that has accumulated across these platforms has gifted people the freedom to explore, experiment with, and be inspired by sounds that suit any desired mood and environment. This abundance, while beneficial to listeners and artists alike, has posed challenges for those conducting research in the fields of Music Information Retrieval (MIR) and MER due to how expansive the range has become for musical mood classification. The challenge of detecting the mood created by a musical piece is rather simple for human listeners, but algorithms designed to carry out the same action have so far only been able to achieve satisfying results to a certain extent; such investigations are both costly and time-consuming, and do not scale to the volume of music required to

address real-world research questions. Many studies have solely considered the audio content of songs in order to classify music by mood (e.g. [3, 14, 16]), others have focussed on using song lyrics to achieve the same goal (e.g. [10, 12]), and some have done so with the use of MIDI files (e.g. [2]). Naturally, these studies have developed simultaneously emerging branches of research, where authors explore the intersection of two or three of these musical elements with multimodal approaches to music mood detection (e.g. [1, 2, 8, 9, 11]). In this project we will investigate this multimodal approach to design a program that considers audio, lyrics, and possibly MIDI file information in order to correctly classify the various moods of musical pieces.

2. PROCESS

Xiao, Kahyun & Downie in [1] presents a robust but flexible framework to undertake music mood classification using multiple information sources. We choose to apply this framework as it is flexible in that each component can be easily extended by changing the tools, algorithms and methods. The roadmap of this approach, which will be enumerated further in the next sections, is - Dataset construction → Feature Generation/Extraction → Feature Selection → Multimodal Combination → Classification → Evaluation and Analysis.

2.1 Dataset

For this project we are going to use the multi-modal MIREX like emotion dataset [2]. This dataset consists of 903 audio clips, 734 lyrics, 196 MIDIs. It uses MIREX mood tags. It contains 5 clusters with several emotional categories in each cluster.

2.2 Feature Generation

After choosing the dataset, the next step is to extract features from various modes of information (audio, lyrics, MIDI).

2.2.1 Audio

In the case of audio, we will generate standard audio features such as tempo, as well as spectral features such as MFCC's. To accomplish this task we may use Marsyas [17], a software framework developed for audio processing in the field of MIR. It is also widely used in this field to study this task [1, 2, 10, 15]. As a backup we will also look into echonest API[13], as it can also help extract spectral features in addition to other percussive, harmonic, and structural features; as well as JAudio[20].

2.2.2 Lyrics

Music mood classification has hit a glass ceiling using single mode approaches [14]. Using lyrical features to compensate for these shortcomings is a common method, as lyrical features have shown to have strong correlation to audio features [14]. They also in some cases outperform audio features [9, 10]. We plan to use various feature types to form a comprehensive set of lyrical features.

Starting with bag-of-words features evaluated on content words. Content words are all words except stop words. In particular, we are going to evaluate a combination of these bag-of-words features and their bigrams and trigrams, using two representational models: Boolean and TF-IDF weighting. Our decision to use these models is based on the excellent results they yield [8, 10]. As for using bigrams & trigrams, it is shown that bag-of-words using uni+bi+trigrams are very suitable for music emotion classification [13].

Next, we extract text stylistic features, which are compiled in [9]. This will be done in part using the jLyrics suites in jMIR [2, 7]. We will also employ Synesketch, a Java API, which uses NLP for textual emotion recognition [2]. We also hoped to use features generated from using English lexicons such as ANEW, but it was deemed to be impractical as lexicons such as ANEW are not readily available to use.

2.2.3 MIDI

We consider the methods used in [2] as this seems to be the only study in multimodal MER that employs features extracted from MIDI files. We employ the tools used in this research. Particularly, jSymbolic software [4], MIR MATLAB MIDI toolbox [5] and jMusic [6].

2.3 Feature Selection

Feature selection is an important component of hybrid MER approaches, as aggressive reduction in lyrical features have proven to give improved results [11]. Tentatively, we are going to employ SVM feature weighting. Other methods in consideration are F-scores, Chi-Square [1] and the relief algorithm [2]. The relief algorithm outputs a weight for each feature which is then ranked and optimal number of features determined experimentally by adding one feature at a time. As one would assume, SVM feature weighting gives the best results when using SVM classifiers.

2.4 Multimodal Combination

Hybrid approaches like ours need a method to fuse the multiple sources of information. We use two methods, namely EFFC (early fusion with feature concatenation), and LFLC (late fusion by linear combination) [15]. We decided to compare these two methods as studies employing them have shown very mixed results [1, 15].

2.5 Classification

We intend to explore and extend this section as we continue our efforts into this task. As a starting point we will start with SVM with grid parameter search as this classifier is shown to have the best accuracy in multiple studies employing the same approach as ours [1, 2, 10, 18, 19]. We will use Python's scikit-learn library.

3. TIMELINE

Feb 28 - Mar 6: Familiarize ourselves with the tools to use, requirement gathering for assigned sections and clean up dataset.

Mar 7 - 13: Complete code for feature generation.

Mar 14 - 20: Complete code for feature selection and multimodal combination

Mar 21 - 27: Complete code for classification.

Mar 28 - end of class: Evaluation and Analysis.

4. INITIAL RESULTS

For our initial experiments we decided to work with each modality separately, so that we could gain familiarity with the dataset as well as machine learning methods we are planning to apply; to find what methods work best and what features in each mode are important to the bigger picture.

4.1 Midi Features

Using jSymbolic [4], we extracted 278 features from 196 Midi files in our dataset, with a feature vector dimension of 1495. Due to the high dimensionality of our feature vector we decided to employ the following feature selection methods :-

1) Tree based (Random Forests) feature selection (TB)
2) Recursive Feature Elimination with SVM estimator (RFE)
3) Anova F-test
4) Chi-squared test

We ran an SVM classifier on the above 4 feature selection methods alongside a baseline. The accuracies are presented in table 1 and figure 1 All experiments were conducted with stratified 10-fold cross validation with 20 repetitions, on a 70-30 train-test dataset split. The parameters for SVM (C value) were selected based on performing a grid search on every step of the feature selection and baseline process. We also normalized the dataset using the sklearn MinmaxScaler so that our SVM would perform better and also to remove negative values as the chi-squared test requires positive feature values.

For tree based feature selection, we used a random forest classifier to assign importance scores to our features. Then by taking a mean of all the importance scores, we discarded the features with scores less than the mean. This left us with a feature vector dimension of 518.

In the case of RFE, we first employed recursive feature elimination with cross validation to find the optimal number of features. Then we employed feature elimination until we were left with only the optimal number of features. In our case, our feature vector was pruned from 1495 to only 15 optimal features. Top features selected by RFE were : *Mean Tempo, presence of pitched instruments, Amount of Staccato, Number of Relatively Strong Rhythmic Pulses, Importance of Middle Register.*
The low number of features combined with high training accuracy compared to test accuracy, leads us to believe that the model might be overfitting in this case and a more in-depth look is necessary.

As a separate experiment, we calculated the ANOVA F-scores and Chi-squares scores of all the features, ranked them, and recursively added the top features one at a time to check the optimal number of features which gives us the best training accuracy. In this case, we found the optimal number of features to be 351 and 590 for F-test and Chi2 respectively

The best performing methods ended up being tree based feature selection and ANOVA F-test. So moving forward for our final experiments we are going to stick to these methods.

Accuracy	Base	TB	RFE	F-test	Chi2
Train	36.77	44.84	48.11	54.25	46
Test	32.20	40.67	33.89	38.98	33.89

Table 1: Training and Test accuracies initial midi

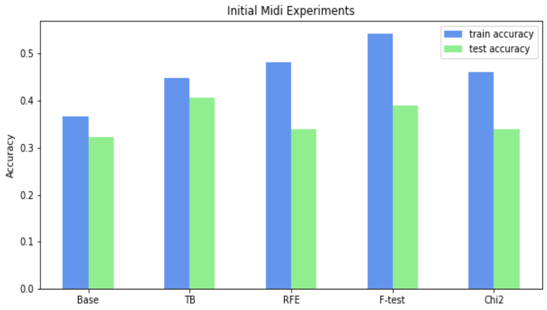

Figure 1: Initial Midi Experiments

4.2 Lyric Features

For the textual processing portion of our program, we used a bag-of-words approach with Tf-Idf and boolean weighting. Before we applied BOW we removed stop words from the text, and applied stemming and lemmatization. Now, using sklearn's CountVectorizer and TfidfVectorizer yielded us with a feature matrix of 764 rows, corresponding to the number of lyric text files given in the MIREX dataset, and 95,979 columns corresponding to the feature words that were extracted using uni+bi+trigrams of text. We used uni+bi+trigrams to capture as much meaning from the lyrics as possible.

Since we used the term frequency-inverse inverse document frequency (Tf-Idf) approach, the values of words increased proportionally to count, but they were inversely proportional to the frequencies of the words in the corpus.

We were not worried about the high dimensionality of our feature vector as SVM with bag of words features requires high feature vector length so as to not lose meaning in the classification process. Still we decided to apply the chi-squared test as this is a known feature selection method used in NLP that not only reduces the feature vector dimension but increases performance for bag-of-words features. Applying the chi-squared test on tf idf features left us with a feature vector dimension of 55,000, while chi-squared on boolean weighted features left us with a feature vector dimension of only 5300.
The accuracies for these methods are presented in table 2 and figure 2.

Accuracy	**Tfidf**	**Tfidf+chi2**	**Bool**	**Bool+chi2**
Train	37.23	40.93	36.83	39.93
Test	39.56	40.93	34.78	39.93

Table 2: Training and Test accuracies initial lyrics

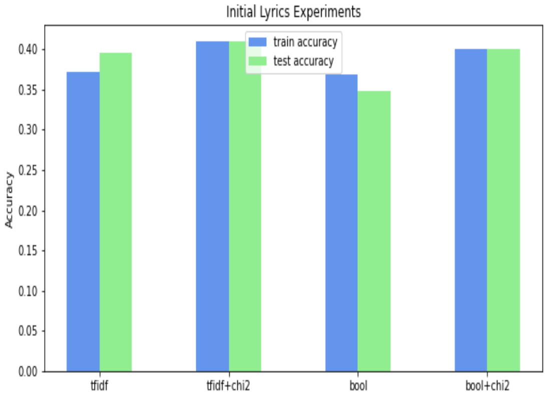

Figure 2: Initial Lyrics Experiments

4.3 Audio Features

Our audio dataset was given to us in a compressed mp3 form. Thus, firstly using Audacity we encoded each audio file in wav format with sampling rate of 22KHz, sample size of 16 bit and a mono channel. Then we moved on to feature generation.
Using JAudio[20], we extracted 44 standard audio features as well as their statistics, for a feature vector dimension of 108. Similar to midi features, we normalized the dataset using sklearn's MinmaxScaler. We first checked the baseline performance on a 70-30 train-test split and using a linear SVM with parameters selected after performing a grid search. This gave us a baseline test accuracy of 95.94%, highest we have seen so far. After this we applied recursive feature elimination, which reduced our feature vector from 108 to 7, giving us a perfect test accuracy of 100%.
All 7 features considered optimal by RFE corresponded to the 2D Method of moments statistic of the magnitude spectrum. From what we could gather, the method of moments statistic is an alternative to maximum likelihood estimate (MLE) as a computationally less expensive way to estimate parameters. The accuracies are presented in table 3 and figure 3.

Accuracy	Baseline	RFE
Train	95.40	99.33
Test	95.94	100

Table 3: Training and Test accuracies initial audio

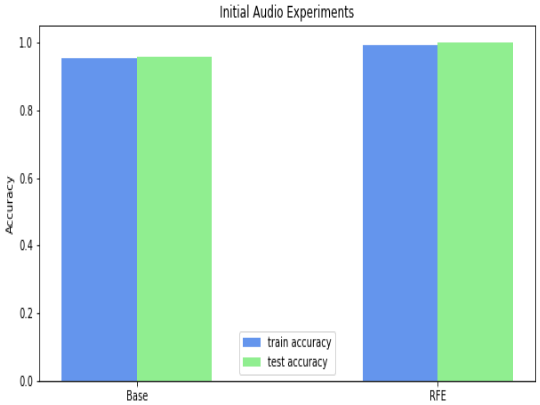

Figure 3: Initial Audio Experiments

Compared to the performance of the audio dataset, the performance of midi and lyrics look mediocre at best. We attribute this to the smaller dataset size of these modes compared to audio.This brought us to a conclusion that given a sufficiently large dataset of audio files, we can accurately classify the mood of the songs into the 5 predetermined clusters.

But can we offset the disadvantage of having a small dataset by using a multimodal approach?

Thus in the next part of our experiments we are going to test the hypothesis that we can still achieve a perfect classification accuracy on an extremely small dataset of 193 entries by combining feature sets of multiple modalities.

5. DIFFICULTIES IN INITIAL EXPERIMENTS

The difficulties we encountered corresponded to model evaluation. In our submitted progress report, we reported midi accuracy with ANOVA F-test to be 65%. This was incorrect as during that time we were using test accuracy as a measure for models performance and not the cross validated training accuracy as we were supposed to. The accuracies measured after using the correct methodology are reported in this final report.

6. SCENARIOS FOR COMPLETION

6.1 Basic/minimum scenario

For our basic scenario we believe to have the models trained for individual modes of information (Midi, lyrics, audio).

6.2 Expected Scenario

Our expected outcome for this project is to have tested a multimodal model trained from features extracted from all 3 modes of information (Midi, lyrics, audio) alongside tests conducted not only on the dataset but random songs introduced into the dataset.

6.3 Stretch Goal

The stretch goal for us is to create a simple app, where any user can input a combination of midi, lyrics and audio files, and the app will automatically detect the mood based on the five clusters we are experimenting with.

7. EXPERIMENTS

We first begin by finding matching entries in our 3 datasets and combining it into a new reduced dataset. These matching entries corresponded to 193 entries in each dataset.

Generating features on these reduced datasets gives us feature vectors of dimensions 1495 for midi, 108 for audio and 24,502 for lyrics. The size of the lyrics feature set has changed as there are fewer words to convert to bag-of-words uni+bi+trigrams. We used Tfidf weighting as it performed best in our initial experiments.

Now first we calculate individual performance of each modality with and without feature selection methods. This will serve as a baseline to compare our multimodal approach.

Next we look at a multimodal combination of these feature sets. We use the best performing feature sets after applying feature selection on individual datasets. As the feature vector dimension of each modality is vastly different, we decided to use 3 different approaches:-

1) Early Fusion
2) Early Fusion with mode specific dimensionality reduction
3) Early Fusion - dimensionality reduction after fusion

Early fusion is simply the method of concatenating feature vectors of each modality and training one classifier with the combined feature set.

For dimensionality reduction we used PCA. We were still new to the idea of PCA, but this tutorial[21] helped demystify this concept and gave us an intuitive idea on how it works. Our goal with PCA was not to improve performance as this would have been accomplished in our feature selection section, but to reexpress our feature vector into a lower dimension and eliminate features that do not contribute at all. Applying PCA before fusion gives us the benefit that the influence of each modality is not affected by the number of dimensions it has. Another benefit of PCA is that it is completely non-parametric, that is we can simply plug our data in and an answer comes out.

For all our experiments we used a linear SVM classifier with 10-fold stratified cross validation with 20 repetitions. As well as using grid search to find the optimal parameters.

8. RESULTS

The results of baseline accuracies are presented in table 4 and figure 4. As expected we see reduced performance across the board.

In case of midi we see the best results for midi features selected after applying the ANOVA F-test. This left us with a midi feature vector of 218.

For lyrics we see that using the chi-squared test to select optimal features performed better than working on the entire feature set. This left us with 5000 features from the original 24502.

Finally for audio we get a high base accuracy of 79.31%, but after using RFE to whittle down the features to only 11, the test accuracy shoots up to 96.55%. The 11 features once again corresponded to the 2D Method of moments statistic of the magnitude spectrum, similar to our initial results.

Methods	Train	Test
Base Midi	39.40%	31.03%
Midi + TB	44.53%	31.03%
Midi + F-test	60.73%	**36.21%**
Base Lyrics	28.76%	36.20%
Lyrics + Chi2	42.79%	**42.79%**
Base Audio	91.79%	79.31%
Audio + RFE	97.02%	**96.55%**

Table 4: Accuracy with feature selection as a baseline

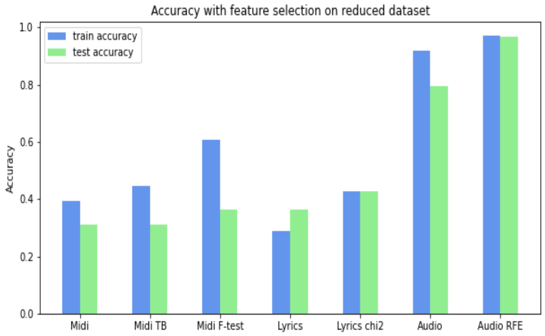

Figure 4: Accuracy with feature selection on reduced datasets.

Using the best performing results above for midi (f-test 218 features), lyrics (tfidf+chi2 5000 features) and audio (rfe 11 features) we try the 3 methods of multimodal combinations mentioned in the previous section and compare their performances. The accuracies for each method can be seen in table 5 and figure 9.

For early fusion with mode specific dimensionality reduction, we first apply PCA to each individual mode. This led to the feature vectors to be reexpressed in 132, 127, 3 components each for midi, lyrics and audio respectively. These components were responsible for the majority of variance in their respective datasets. The visualization for explained variance in each case is presented in figure 5,6,7.

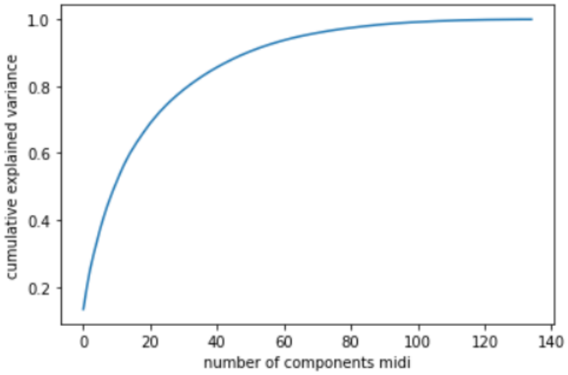

Figure 5: Variance vs No. of components Midi

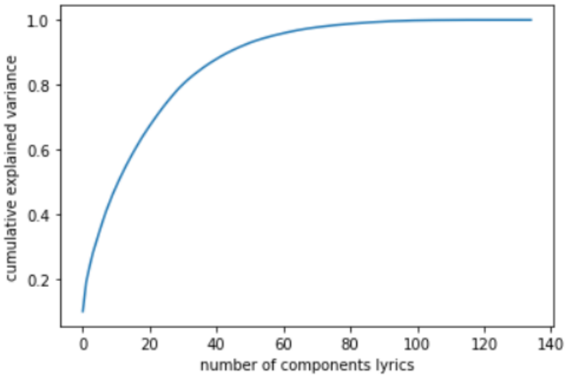

Figure 6: Variance vs No. of components Lyrics

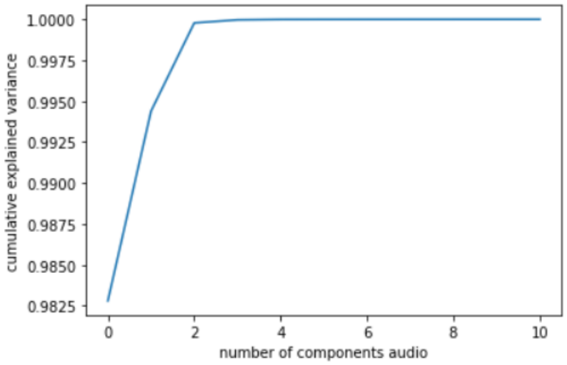

Figure 7: Variance vs No. of components Audio

In case of dimensionality reduction after fusion, applying PCA re-expresses the concatenated feature set into 133 components corresponding to the majority of variance. The visualization for explained variance is presented in figure 8.

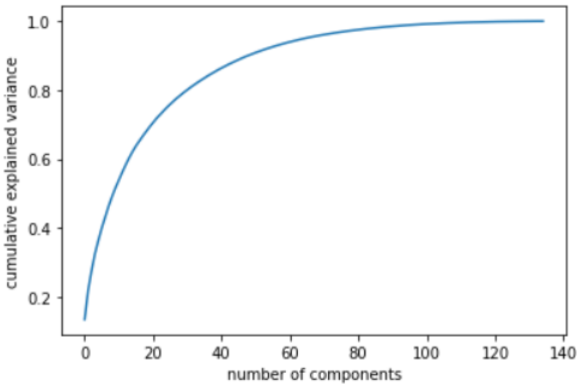

Figure 8: Variance vs No. of components After fusion

Methods	Train	Test
Early Fusion	84.29%	72.41%
EF mode specific	84.29%	72.41%
EF after fusion reduction	84.75%	72.41%

Table 5: Accuracies for multimodal combinations

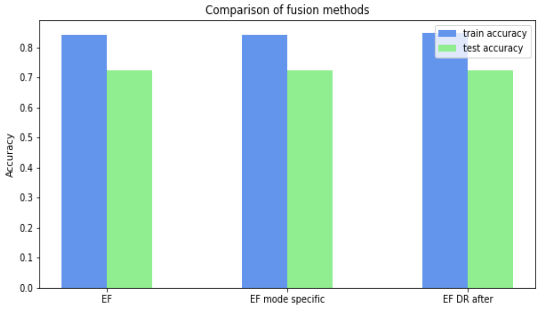

Figure 9: Comparison of fusion methods

Performance of all fusion methods are the same across the board, but we were still able to drastically reduce the feature vector dimensions from 5229 in plain early fusion, down to 133 after using PCA.

9. DISCUSSION

Looking at the baseline accuracy of the reduced datasets with and without feature selection, we see that audio features perform the best with 96.55% accuracy on our validation set, when used in conjunction with recursive feature elimination. This is lower than the perfect 100% accuracy we got on the full dataset, but is leaps and bounds better than the corresponding midi and lyrics feature sets which showed best accuracies of 36.21% and 42.79% respectively.

We hoped that using multimodal fusion we would be able to bridge the gap towards a 100% accuracy on a reduced dataset of only 193 entries. But alas we were wrong as using multimodal fusion drastically reduced our performance to only 72.41% accuracy on the validation set.

As for comparing different fusion methods, they all performed the same but using early fusion with dimensionality reduction after fusion, converted our high dimensional feature vector into only 133 components. This helped significantly cut down computational expenses while having no effect on performance.

Thus, our hypothesis that we can achieve a perfect classification accuracy on an extremely small dataset by combining features from multiple modalities proved to be wrong.

10. CONCLUSION

We started this project with the goal of measuring the effects of combining features from multiple music modes in detecting the mood of a song. Our initial experiments showed that audio features alone can gain 100% classification accuracy on the mood of our songs. We attributed the poor performance of our other modes to the small size of their datasets. Next we shifted our focus to see if we could achieve this same perfect classification accuracy on a reduced dataset by combining the generated features on each modality. This experiment again showed that audio features working on 1/5th the dataset still gave a near perfect classification accuracy. Additionally, multimodal fusion not only failed to get us perfect accuracy, but drastically decreased our performance. Thus, by the work we have done in this project we conclude that given a sufficiently large dataset of audio files, standard audio features alone are enough to detect the mood of a song.

11. FUTURE WORK

For future works regarding this project we would have hoped to work on a larger dataset of midi files. Also we would have liked to work on the lyrics dataset more by not sticking to BOW features alone. We initially planned to extract text stylistic features as well as features based on lexicons such as ANEW or Word Net, but time constraints as well availability of the lexicons stopped us from doing so. In hindsight we should have also tested BOW features before applying stemming and lemmatization. Also, we really wanted to try late fusion as we had drastically different performing modalities and we believe that appropriately weighting outputs of their classifier before training a separate classifier on these outputs may have led us to a perfect multimodal mood detection model. Finally, we hoped to create a simple app which, given an audio file, would generate standard audio features on the fly, run the mood classification model and output the result.

12. REFERENCES

[1] Hu, Xiao & Choi, Kahyun & Downie, J.. (2016). A framework for evaluating multimodal music mood classification. Journal of the Association for Information Science and Technology. 68. n/a-n/a. 10.1002/asi.23649.

[2] Panda R., Malheiro R., Rocha B., Oliveira A. & Paiva R. P. (2013). “”. 10th International Symposium on Computer Music Multidisciplinary Research – CMMR'2013, Marseille, France.

[3] Salamon, Justin & Gómez, Emilia. (2011). Melody Extraction from Polyphonic Music: MIREX 2011.

[4] Mckay, Cory & Fujinaga, Ichiro. (2006). JSymbolic: A feature extractor for MIDI files. International Computer Music Conference, ICMC 2006.

[5] Eerola, Tuomas & Toiviainen, Petri. (2004). Mir in matlab: The MIDI toolbox.
[6] Brown, Andrew R. & Sorensen, Andrew C.(2000) Introducing jMusic. In Australasian Computer Music Conference, 2000-07-05 - 2000-07-08.

[7] Cory Mckay. 2010. Automatic music classification with jmir. Ph.D. Dissertation. McGill University, CAN. Order Number: AAINR68492.

[8] Xiao Hu and J. Stephen Downie. 2010. Improving mood classification in music digital libraries by combining lyrics and audio. In Proceedings of the 10th annual joint conference on Digital libraries (JCDL '10). Association for Computing Machinery, New York, NY, USA, 159–168. DOI:https://doi.org/10.1145/1816123.1816146

[9] Hu, X., & Downie, J.S. (2010). When Lyrics Outperform Audio for Music Mood Classification: A Feature Analysis. ISMIR.

[10] Hu, Xiao et al. "Lyric Text Mining in Music Mood Classification." ISMIR (2009).

[11] C. Laurier, J. Grivolla and P. Herrera, "Multimodal Music Mood Classification Using Audio and Lyrics," *2008 Seventh International Conference on Machine Learning and Applications*, 2008, pp. 688-693, doi: 10.1109/ICMLA.2008.96.

[12] He, Hui & Jin, Jianming & Xiong, Yuhong & Chen, Bo & Sun, Wu & Zhao, Ling. (2008). Language Feature Mining for Music Emotion Classification via Supervised Learning from Lyrics. 426-435. 10.1007/978-3-540-92137-0_47.

[13] Mcvicar, Matt & Freeman, Tim & Bie, Tijl. (2011). Mining the Correlation between Lyrical and Audio Features and the Emergence of Mood.. Proceedings of the 12th International Society for Music Information Retrieval Conference, ISMIR 2011. 783-788.

[14] Aucouturier, Jean-julien & Pachet, Francois. (2004). Improving Timbre Similarity: How high's the sky?. J. Negat. Results Speech Audio Sci. 1.

[15] Yang, yi-hsuan & Lin, Yu-Ching & Cheng, Heng-Tze & Liao, I-Bin & Ho, Yeh-Chin & Chen, Homer. (2008). Toward Multi-modal Music Emotion Classification. 70-79. 10.1007/978-3-540-89796-5_8.

[16] Hu, Xiao & Downie, J. & Laurier, Cyril & Bay, Mert & Ehmann, Andreas. (2008). The 2007 MIREX audio mood classification task: Lessons learned. ISMIR 2008 - 9th International Conference on Music Information Retrieval. 462-467.

[17] Tzanetakis, G., & Lemstrom, K. (2007). Marsyas-0.2: a case study in implementing music information retrieval systems. Intelligent Music Information Systems.

[18] Yang, Y.-H. et al: A regression approach to music emotion recognition. IEEE Trans.
Audio, Speech and Language Processing, Vol. 16, No. 2 (2008) 448–4

[19] Cheng, H.-T. et al: Automatic chord recognition for music classification and retrieval. Proc. ICME (2008) 1505–150

[20] McEnnis, Daniel & McKay, Cory & Fujinaga, Ichiro & Depalle, Philippe. (2005). jAudio: An Feature Extraction Library.. Proceedings of the International Conference on Music Information Retrieval. 600-603.

[21] Shlens, Jonathon. "A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS Derivation , Discussion and Singular Value Decomposition." (2003).

Appendix A : Graphs

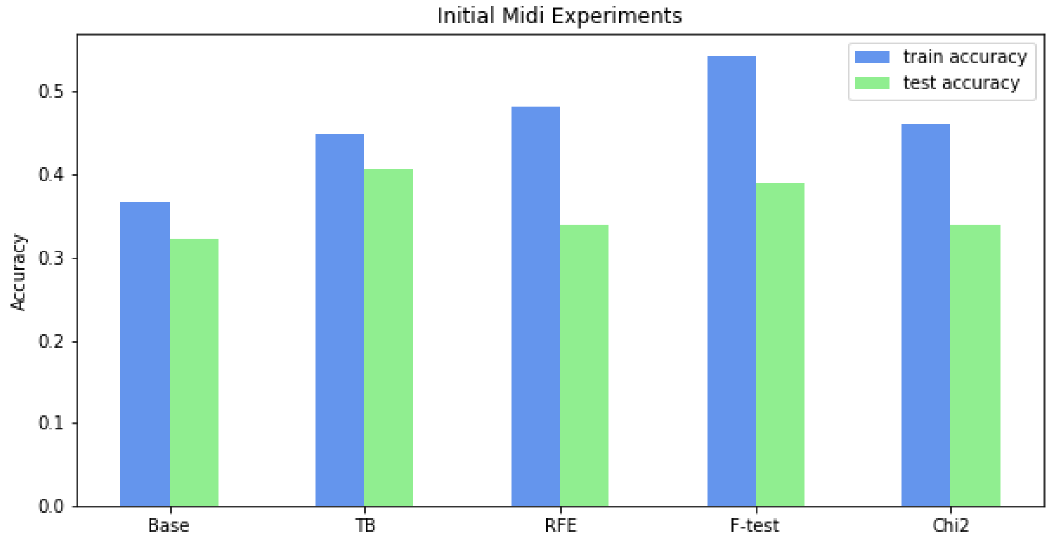

Figure 1: Initial Midi Experiments

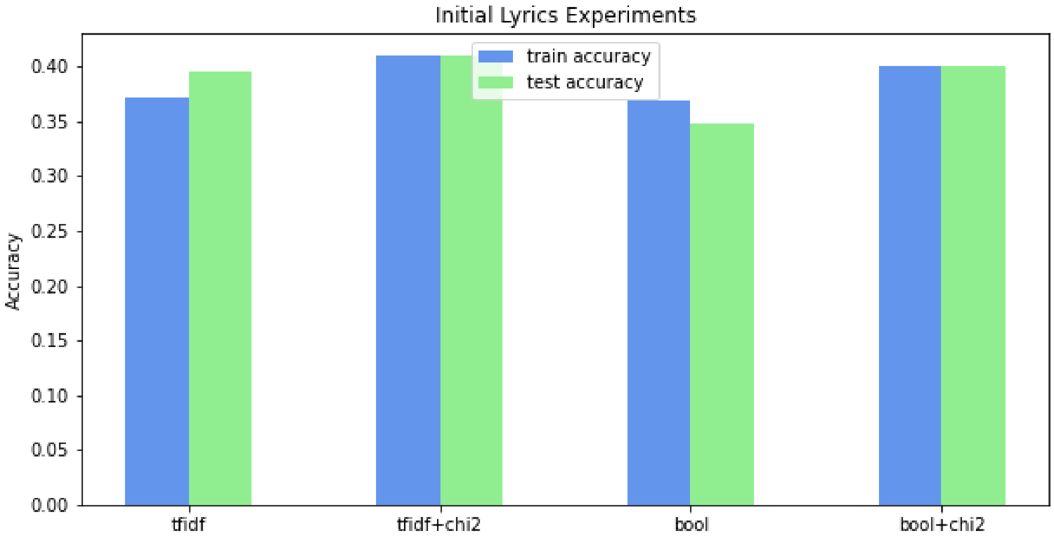

Figure 2: Initial Lyrics Experiments

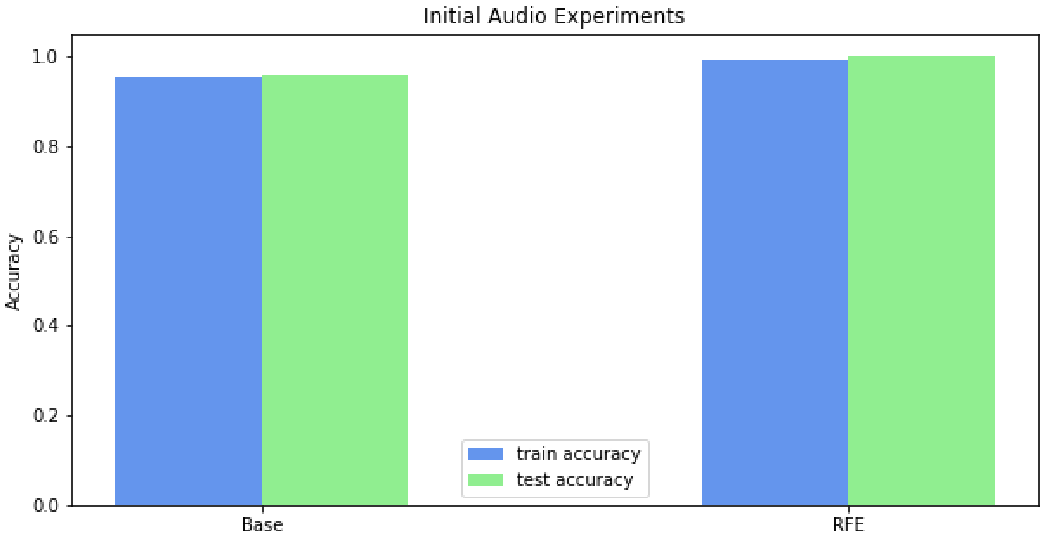

Figure 3: Initial Audio Experiments

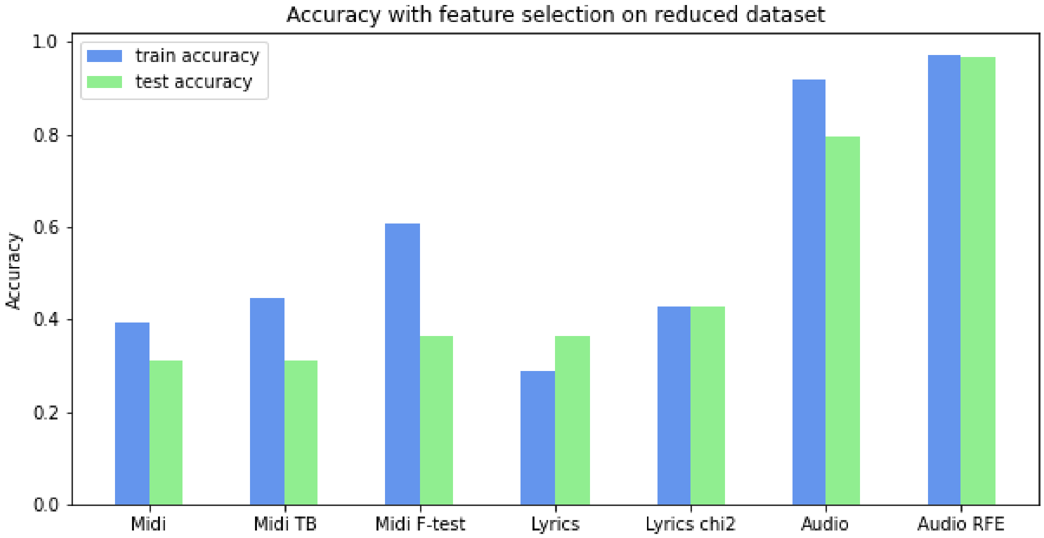

Figure 4: Accuracy with feature selection on reduced dataset

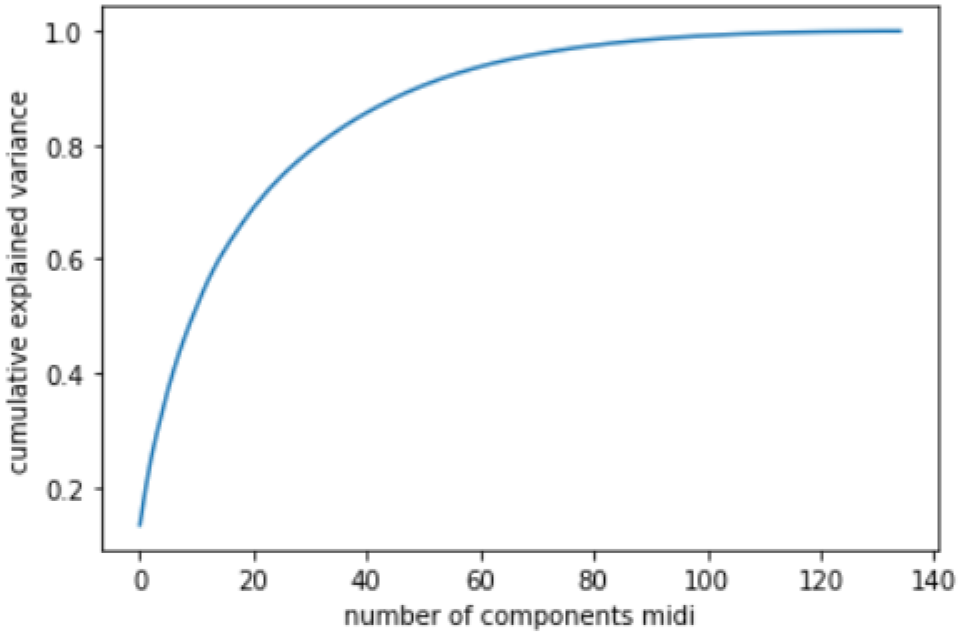

Figure 5: Variance vs No. of components Midi

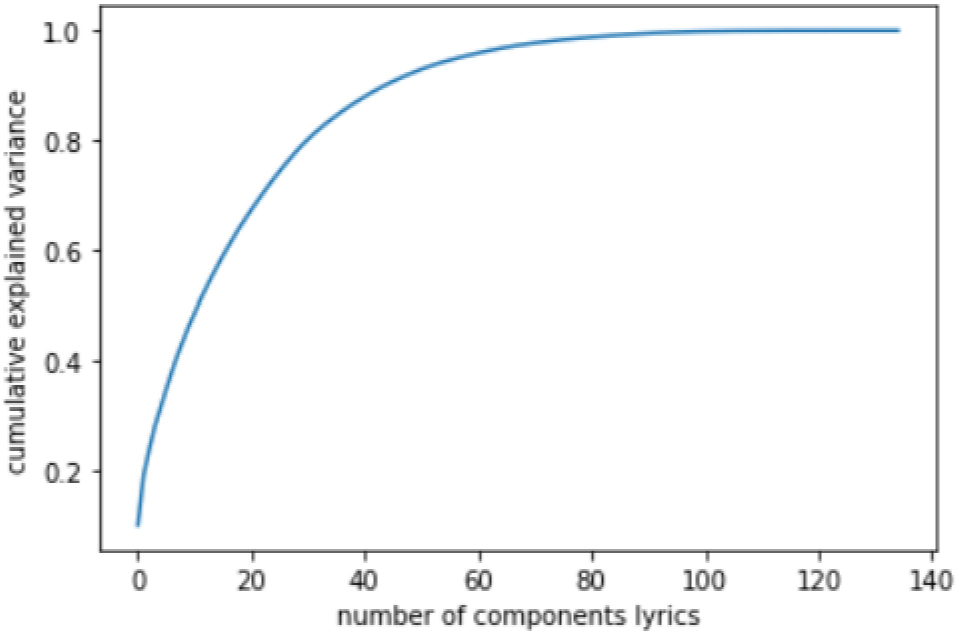

Figure 6: Variance vs No. of components Lyrics

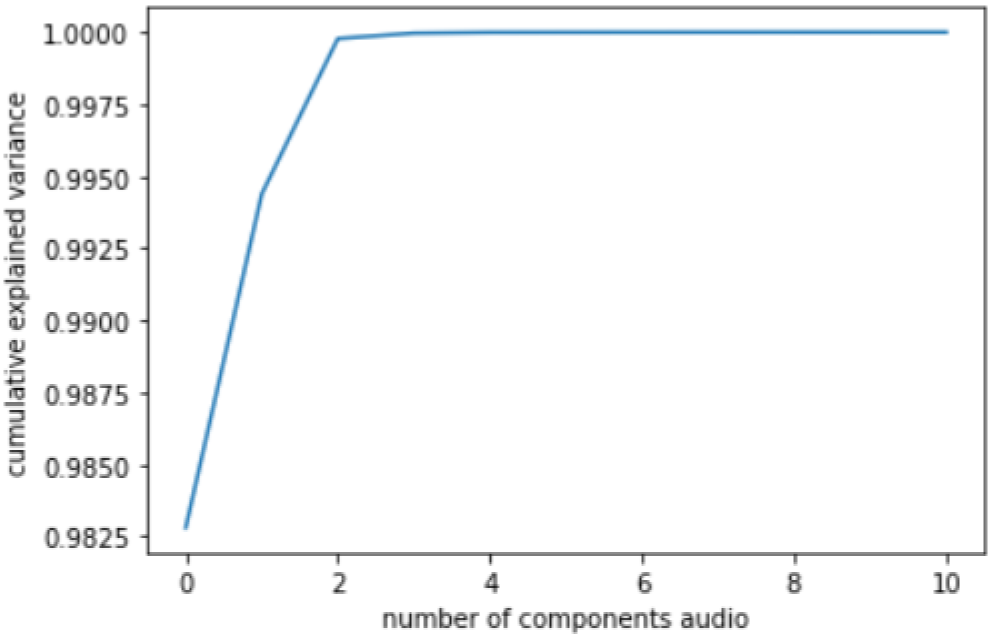

Figure 7: Variance vs No. of components Audio

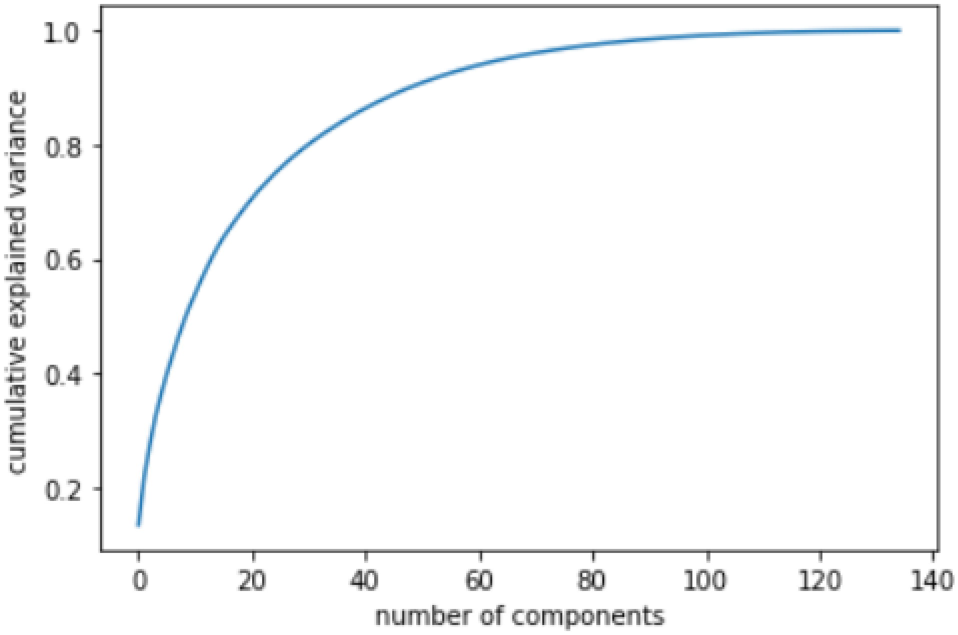

Figure 8: Variance vs No. of components After fusion

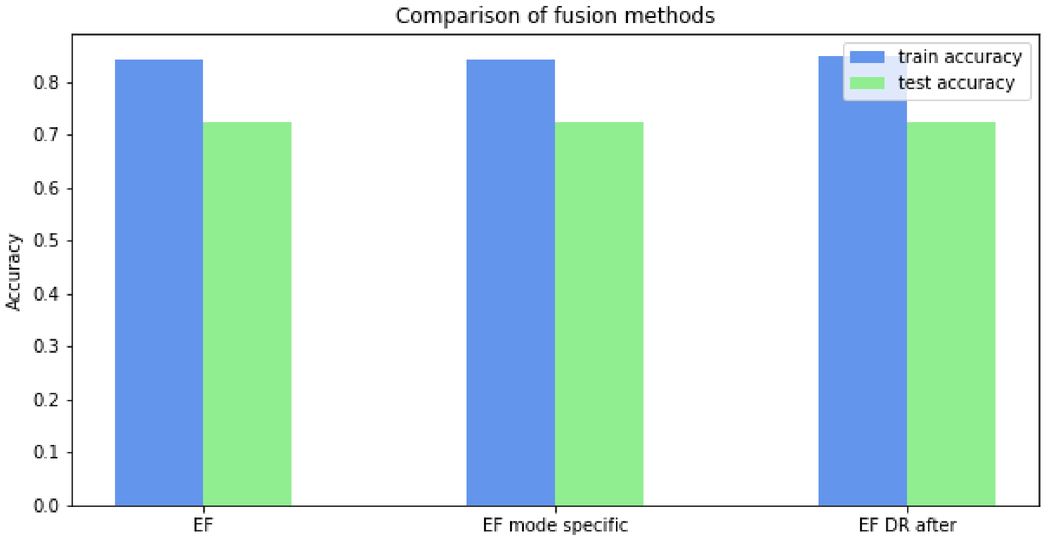

Figure 9: Comparison of fusion methods