SONG EMOTION PREDICTION BASED ON LYRICS USING MULTI-CLASS CLASSIFICATION

Siyal Sonarkar- sonarkar@usc.edu

Nivedita Suresh- nsuresh@usc.edu

Muthulakshmi Chandrasekaran - muthulac@usc.edu

PROJECT DESCRIPTION

The project aims to create a dataset that classifies English songs based on emotions. The songs will be classified into three moods – Happy, Sad or Angry. The data extracted contains the fields Artist, Song name, Lyrics and Label. The labels describe the mood of the song.

DESCRIPTION OF HOW THE DATA WAS COLLECTED:

The dataset contains Title, Artist and the URL for each song, along with the labels.

Data is collected from the website https://genius.com/ , that contains the lyrics of songs in multiple languages. This website has a free API for extraction of data. The website provides an extensive list of the top 5000 songs, with the song title and the artist name given. Using this API, song title and artist names were extracted. The URL for the lyrics of each song can be generated, as each URL has a definite pattern defined in terms of the song title and artist name.

The pattern is https://genius.com/ArtistName-SongTitle-Lyrics , where the ArtistName is the name of the artist, and SongTitle is the title of the song. Both ArtistName and SongTitle have to be given without any punctuation, and the spaces must be replaced by a hyphen. Hence, the initial pre – processing was done by simple regex commands, and the URL for each song are obtained.

Thus, the dataset contains the Song title, Artist Name and the URL for each song. The dataset initially had the entire lyrics, but due to memory constraints, the data field is changed to URL in the final dataset.

Since the actual lyrics cannot be directly extracted for each song, the Python Package BeautifulSoup was used to extract the lyrics. Using HTML Parser, the actual lyrics was extracted.

The dataset contains a total of 2022 data points. The actual aim was to collect a total of 2000 points (in the initial stage). The actual list extracted from the website contains close to 4000 songs, which is a mixture of many languages like Spanish, German, French etc. Songs in languages other than English are eliminated, and songs in form of conversations were eliminated. After preliminary filtering, a dataset of 2022 valid, properly labelled points are obtained. Figure 1 gives the number of songs that belong to each label, wherein there are 818 Happy songs, 666 Sad Songs and 538 Angry songs, summing to a total of 2022 songs.

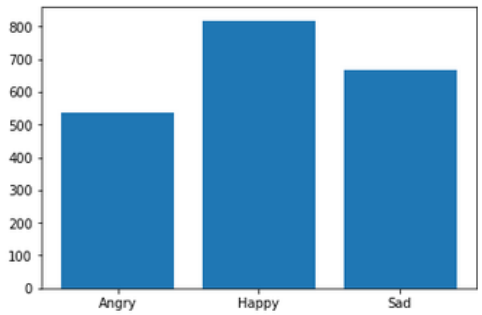


Figure 1 Number of Songs in each category

Figure 2 gives the percentage of each class in the total dataset.

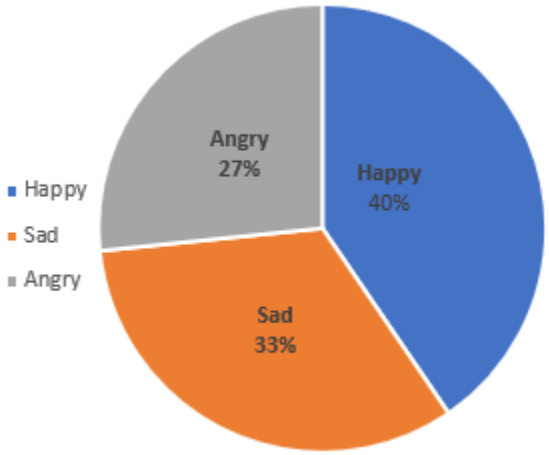


Figure 2 Percentage of each class in the dataset

DESCRIPTION OF HOW THE DATA WAS LABELED:

The aim is to classify the songs into moods based on its lyrics. Each song is to be classified into either of the three labels – Happy, Sad or Angry. Data labelling is done manually.
Overview of Labels:

- A song is categorized as Happy if it conveys positive feelings or positive energy; celebration, positive motivation, or excitement.
- A song is labelled Sad if it conveys a negative tone, feelings of broken trust or solitude.
- An Angry song carries feelings of frustration towards any person or thing, violence or hatred.

Labelling was performed based on the following guidelines:

- This is a multiclass classification, where each song can have only one of the three labels.
- Each song is labeled based on the significant emotion it carries.
- Each song is to be labelled based on the lyrics only, and not according to the audio version of it.
- If a song is found to convey a mixture of emotions, the most prominent one is chosen based on the lyrics.
- The gender of the song writer or the song composer should not be considered while labeling.
- The title of the song doesn't necessarily convey emotions, the song should be read completely and thoroughly to determine its emotion.

- Data labeling was done manually. We did a preliminary data labeling of 100 data points per person, summing to a total of 300 data points. The rest of the datapoints were labeled by 10 different people (family and friends), strictly using the above guidelines.

Since each data point (or a song) is an extensive group of text, having an average of approximately 50 lines, the time taken for labeling each data point was quite high.

Resolving Ambiguity while labeling:

If any song is found to have ambiguous labels, it was marked separately (highlighted in the database). For each of these ambiguous songs, all annotators were asked to label them. The emotion of the song was chosen by the mood that received the maximum number of votes.

After labeling the data, representation of each of the labels could be visualized using the WordCloud Package in Python. This is shown in Figure 3; where bigger word size implies more frequent occurrences of the word.

Figure 3(a) Word Cloud Visualization for the label 'Happy'

Figure 3(b) Word Cloud Visualization for the label 'Sad'

Figure 3(c) Word Cloud Visualization for the label 'Angry'

Figure 3 Word Cloud Visualizations

[We had a fourth team member, who dropped the course 3 weeks before the project due date. Hence, her part of data collection and labeling could not be completed]

CLASSIFIER APPROACH:

The actual dataset contains the fields Song Title, Artist Name, URL and the labels.

Initial Filtering:

The dataset is checked for songs from languages other than English, and those entries are removed.
The actual lyrics are extracted using the HTML Parser feature of BeautifulSoup package.
These lyrics are crucial to building the classifier.
There are three classes, and hence a random approach would be giving an accuracy of around $33\%\left(=\frac{1}{3}\right)$.
There is also a slight class imbalance observed in the data, according to Figure 2.

Baseline Approach :

The baseline classifier is built based on the currently available Python Packages. The package WNaffect gives the emotion of the word using the resources of the WordNet database.

[Reference : https://github.com/clemtoy/WNAffect]

The already available code is modified to build the classifier.

After the initial filtering, the dataset is split into training and test data sets in the ratio 80:20% respectively. Using the training set, the features are extracted. Here, only one feature – Bag of Words is used.

BoW here contains only the emotions – each word in the song is coverted to the best emotion it expresses using WNaffect. Only the words that output an emotion when using WNaffect are used in BoW feature matrix. Hence, the features for each song will be a count of all emotions that words in the song lyrics represent (according to WNaffect).

Once the features are extracted, classification is done using Logistic Regression. The optimal parameters of the classifier are obtained by cross – validation. The final classifier with optimal parameters on the validation set is tested on the test set. The performance measures that are used to evaluate the classifier are accuracy and confusion matrix. Figure 4 gives the performance measures of the baseline classifier.

Modified Approaches:

To improve the accuracy of the classifier, the following approaches are followed.

Preprocessing and Feature Engineering:

- The classifier has only three labels – Happy, Angry and Sad. To avoid spurious labels, the format of each label is checked. Each entry in the labels column is checked for title case, and any other extra characters are removed.
- Lyrics are checked for continuous entries of other language characters and are replaced by their closest ASCII Character.

The most important feature that is used in classification is the lyrics. All processing is done using the lyrics obtained.

Initial Cleaning Techniques that are used are:

- Case insensitive: All the words are converted into one case.
- Punctuation removal: All punctuations are removed.
- Tokenizer: Each entry is tokenized – meaning split into tokens.
- Removal of stop words: All the frequently occurring words can be removed – like ‘a’, ‘the’ etc.
- Lemmatization: Lemmatization is used to group words based on the base form. This is better than stemming, because stemming doesn’t take care of the context of the word, while lemmatization considers the word contexts, and finds the base form of the word.

The following are the features that are obtained from the lyrics, that are more meaningful in representation than the actual lyrics.

- Bag of Words
  Bag of Words is used to represent if the word is present in the actual document or not. All the words from the document are collected, and unique words are put into the vocabulary. They are ranked based on the frequency of occurrence. Here, the CountVectorizer function is used to determine the Bag of Words, giving a maximum of 1000 feature. This is an important feature because the order of occurrences of a word are not taken into consideration.
  Both individual words and bigrams are used as features here.
- Song Structures
  The important song structures are Chorus, Verse, Bridge, Hook, Intro and Outro.
  Chorus refers to the part of the song which are to be sung by two or more people, mostly stressing the emotion of the song. More number of choruses imply more stronger emotions, and hence is a better feature.
  Verse usually contains the details of the song that convey the primary meaning of the song. This is a useful feature, because the mood of the song is best represented by verse.
  Bridge is a section that contradicts verse, and sometimes helps to accurately get the meaning of the song. Hence, useful feature.
  Hook is the most attention – seeking part of the song, that expresses the strongest emotions. This is an useful feature.
  Intro and Outro are introduction and conclusion to the songs respectively, that signify the mood of the song.
- Song Length
  This feature represents the length of the song, in terms of the number of lines in the song. This is a useful feature since the number of lines determine the extent of emotions of a song.
- Artist Name
  The name of the artist is used as a feature, since there are patterns that each artist makes songs of a particular emotion. This need not be always applicable, but this could be a useful feature since this aids in finding some patterns.
- Song Title
  Sometimes the mood of the song could be reflected in the song title, and hence this could be a useful feature.

BUILDING THE CLASSIFIERS:

The labels are label encoded, before using in the classifier.

The data is then split into train and test sets, with the ratio being 80:20 for training and test sets respectively.

Then the following supervised – learning classifiers are tried and implemented. For each classifier, the best parameters are selected by cross – validation of the training set. The parameters that give the highest accuracy on the validation set are taken as the optimal parameters.

- **Random Forest Classifier**
  Random Forests are tree-based classifiers, that predict on randomly sampled subsets of data from the actual dataset. The final label is chosen based on the number of votes of all the decision trees.
- **Support Vector Machines**
  Support Vector Machine algorithms classify data by creating optimal hyperplanes, that differentiates different classes.
- **Logistic Regression**
  The logistical regression classifier is used for classification rather than regression. It uses sigmoid function to determine the probability of a data belonging to a class.
- **Naïve Bayes Classifier**
  Naïve Bayes Classifiers assume that features are strongly independent of each other. Gaussian Naïve Bayes, Multinomial Naïve Bayes and Bernoulli Naïve Bayes Classifiers are implemented. It is also required that only a small amount of training data is necessary for good classification. They are also fast because each feature is assumed to be independent.
  In Gaussian Naïve Bayes, the likelihood of the features is assumed to be Gaussian.
  Bernoulli Naïve Bayes is useful for classification of data where each feature is Bernoulli – distributed.
  Multinomial Naïve Bayes is useful for text classification of data where the data is represented as word – vector counts.
- **Convolution Neural Network**
  One – dimensional Convolutional Neural Network is constructed to build the classifier. CNN is built to learn from the scratch, without using any word embeddings. This used many multi- layer perceptron for faster processing. The following layers are constructed:
  - Convolution 1D Layer – 64 filters, and kernel size 10
  - Max Pooling Layer, with pool size 5
  - Convolution 1D layer, with 64 filters, and kernel size 8
  - Max Pooling Layer, with pool size 2
  - Dropout of 0.5
  - Learning Rate of 0.05
  - Stochastic Gradient Descent Optimizer
  - Number of epochs = 30

RESULTS:

The following are the metrics that are used in the evaluation of classifiers:

- Accuracy Score – This is a measure of how close the true labels and the predicted labels are for a given dataset.
- F1 Score – This is a weighted average of Precision and Recall.
- Confusion Matrix – This is a matrix representation of how close true values are to the predicted values. It has four elements – True Positive, True Negative, False Positive and False Negative. True Positive and True Negative. Precision and Recall are calculated based on these values, which eventually gives F1 Score.

Baseline Approach Results:

In the Baseline approach we use emotion features generated using WNAffect Python package. This method gives lesser number of features from the lyrics data, but still manages to give accuracies that are higher than random label classifier. The highest test accuracy obtained with this approach is 44.9%. We see that the accuracy is not high enough. We need to improve the classifier by using a more effective feature selection method. The train accuracy is quite poor, and the classifier has really not learnt any feature of the input data.

The baseline approach results are as follows:

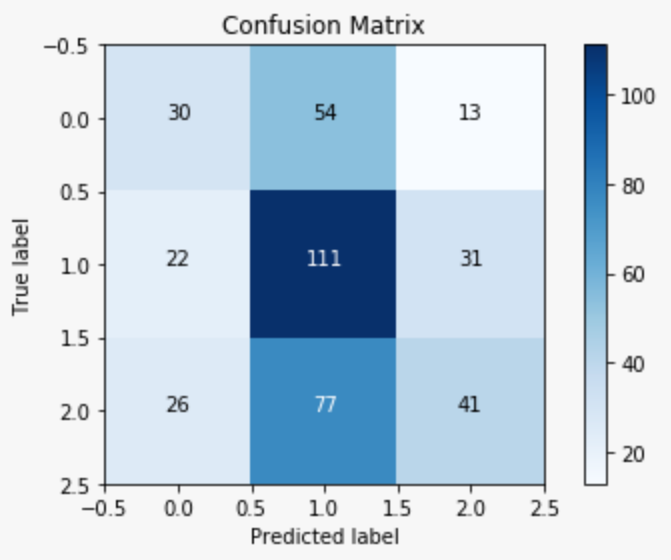
```
              precision    recall  f1-score   support

    class 0       0.38      0.31      0.34        97
    class 1       0.46      0.68      0.55       164
    class 2       0.48      0.28      0.36       144

avg / total       0.45      0.45      0.43       405

Train Accuracy:  0.551638837353123  Test Accuracy:  0.44938271604938274
Train Precision:  0.5840152335659395 Test Precision:  0.44188200391408955
Train Recall:  0.5221230113895183    Test Recall:  0.42360994701012306
Train F1 Score:  0.5252007126649     Test F1 Score:  0.4159112583446269
```

The confusion matrix obtained is shown below.



Confusion matrix gives us a measure of Precision and Recall of the labels of each class. A higher percentage of data is correctly classified into the second class. This is the class having highest number of labels (Label: Happy).

Baseline+Modified Approach:

Due to the low accuracy of the Baseline approach, we develop another classifier which extracts additional features based on the song lyrics, artist names, song title and the number of choruses, verses, hooks, intro etc. of each song. From intuition we know that these features will provide a lot of information about the mood of the song. For instance, artist name and song mood are directly correlated as many artists tend to write songs of similar mood/emotion. The lyrics of the song gives valuable information about the emotion the song is trying to convey.

To evaluate this approach, we use four types of classifier and compare them to find out which one gives the highest accuracy. The results are consistent with our intuition and we see that this approach is way better than our Baseline approach. The train accuracies are very good, and we get a maximum test accuracy of around 55.55% when using a Naïve Bayes classifier.

The following table gives the Accuracies and F1 Scores for different classifiers in the Modified approach.

	Training Accuracy	Test Accuracy	Training F1 Score	Test F1 Score
Naïve Bayes	0.9672	0.5555	0.9667	0.5502
SVM	0.7953	0.5259	0.7883	0.5131
Random Forests	0.9963	0.5136	0.9963	0.4753
Logistic Regression	0.8287	0.5407	0.8227	0.5344

Despite the higher train accuracies when compared to the Baseline approach, we see that the test accuracy is still not high enough. This is because, in combining the Baseline features and the new ones generated, we create a model that tends to overfit on the training data by using lot of features.

Modified Approach:

In this approach we do not combine the features obtained from the Baseline approach. This reduces overfitting as we use only those features that are directly related to the structure and words of the song lyrics. Intuitively, this should give us the best results in terms of accuracy, and surely, we see that our test accuracies are better than the other approaches.

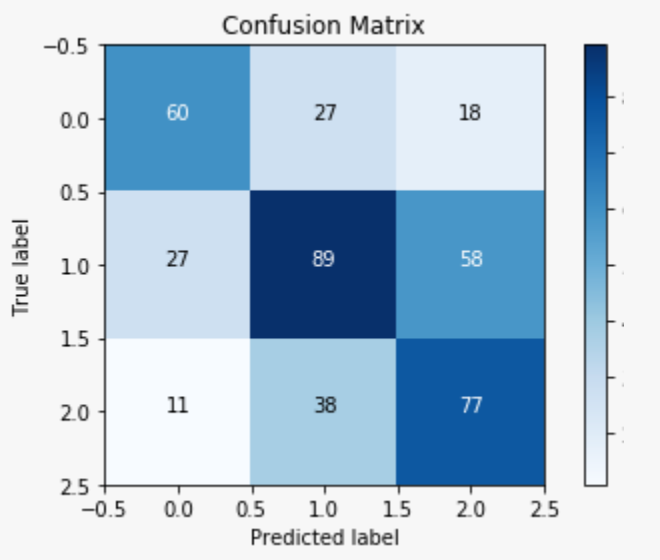
In the tables below, we compare different classifiers that were used. The non-deep learning classifiers did quite well, with Logistical Regression giving the maximum test accuracy of 57.04%.

	Training Accuracy	Test Accuracy	Training F1 Score	Test F1 Score
Naïve Bayes	0.9715	0.5580	0.9715	0.5619
SVM	0.7786	0.5259	0.7710	0.5219
Random Forests	0.9987	0.5259	0.9987	0.4978
Logistic Regression	0.8312	0.5704	0.8262	0.5663

The following graphs compare them.

The confusion matrix for different classifiers are shown separately below.

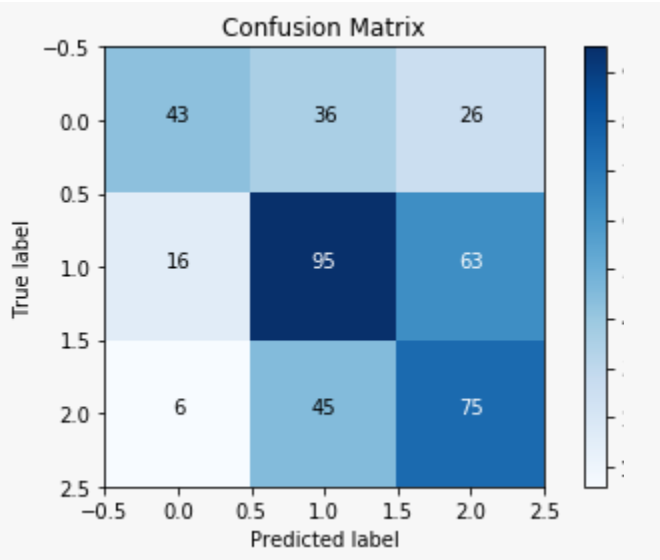
Naive Bayes



	precision	recall	f1-score	support
class 0	0.61	0.57	0.59	105
class 1	0.58	0.51	0.54	174
class 2	0.50	0.61	0.55	126
avg / total	0.56	0.56	0.56	405

Train Accuracy: 0.9715522572665429
Test Accuracy: 0.5580246913580247
Train Precision: 0.9719444118321414
Test Precision: 0.5644783165791569
Train Recall: 0.9721080504141053
Test Recall: 0.5646779784710819
Train F1 Score: 0.9715601802739428
Test F1 Score: 0.5619290859734171

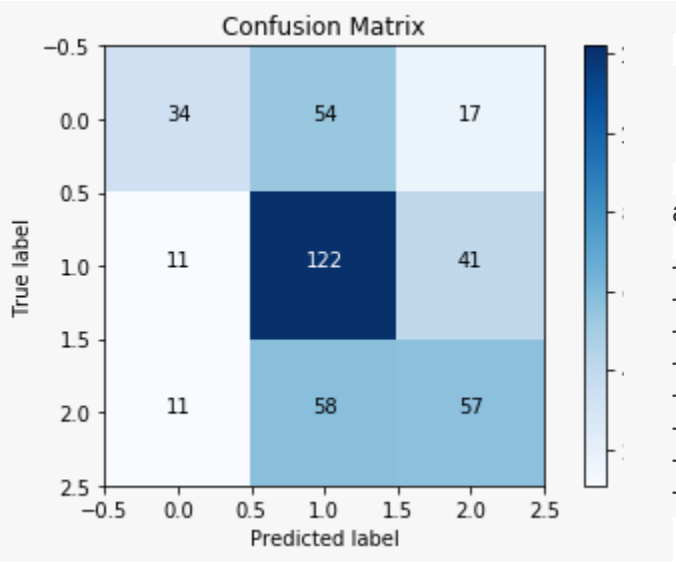
SVM



	precision	recall	f1-score	support
class 0	0.66	0.41	0.51	105
class 1	0.54	0.55	0.54	174
class 2	0.46	0.60	0.52	126
avg / total	0.55	0.53	0.53	405

Train Accuracy: 0.7786023500309215
Test Accuracy: 0.5259259259259259
Train Precision: 0.7973458810692854
Test Precision: 0.5528760873273069
Train Recall: 0.7677525650581848
Test Recall: 0.5169129720853859
Train F1 Score: 0.7710541714394932
Test F1 Score: 0.5219936250362213

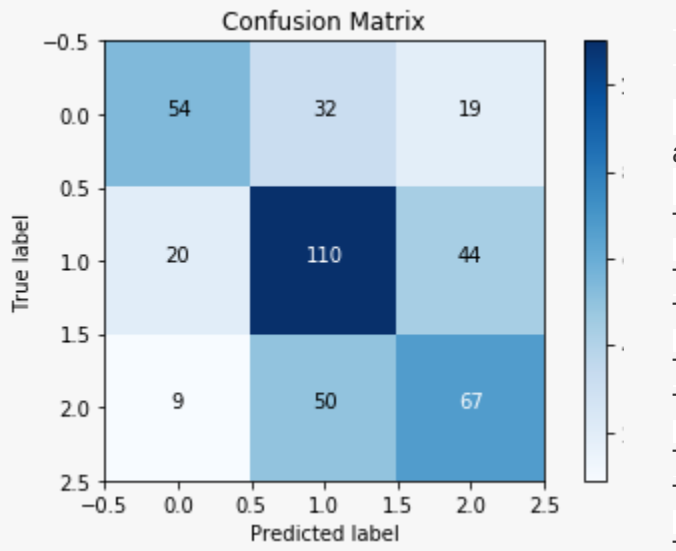
Random Forest



	precision	recall	f1-score	support
class 0	0.61	0.32	0.42	105
class 1	0.52	0.70	0.60	174
class 2	0.50	0.45	0.47	126
avg / total	0.54	0.53	0.51	405

Train Accuracy: 0.9987631416202845
Test Accuracy: 0.5259259259259259
Train Precision: 0.9987699876998771
Test Precision: 0.541387517474474
Train Recall: 0.9987125787155912
Test Recall: 0.49244663382594417
Train F1 Score: 0.9987395003411158
Test F1 Score: 0.4978095032588777

Logistical Regression



	precision	recall	f1-score	support
class 0	0.65	0.51	0.57	105
class 1	0.57	0.63	0.60	174
class 2	0.52	0.53	0.52	126
avg / total	0.58	0.57	0.57	405

Train Accuracy: 0.8311688311688312
Test Accuracy: 0.5703703703703704
Train Precision: 0.8366085680766183
Test Precision: 0.579634563896612
Train Recall: 0.8214507080440742
Test Recall: 0.5594052180259076
Train F1 Score: 0.8261707829880943
Test F1 Score: 0.5663328270937488

Convolutional Neural Network:

CNN for the Modified Approach gives the result as shown below.
We also use a Deep learning model for evaluating the performance of this approach. A Convolutional Neural Network with the architecture described above was used. The train accuracy is close to 100% but the test accuracy is lower than non-deep learning classifiers. This is due to smaller number of data points and over-fitting of the CNN model to the training data.

```
Test Loss: 3.7098976488466615
Test Accuracy:  0.4962962977680159
Train Loss: 0.00976430055958633
Train Accuracy:  0.9987631416202845
```

SUMMARY:

The following are the observations made:

- The Modified approach of feature extraction gave the best performance on test data.
- Using multiple classifiers like Naïve Bayes, SVM, Random Forests and Logistic Regression help in improving the Test Accuracy and F1 Score.
- Logistical Regression model performed better than other classifiers on Test data.
- The test accuracies and the train accuracies obtained for CNN were lower than other classifiers used.

Final Result: The song emotion predictor based on lyrics gives a **Train accuracy of 83.12% and Test accuracy of 57.04%.**