Extensive Analysis of Lyrics and Music Features for Various Songs' Classifications

Arun Govindaiah
ag5305@nyu.edu

Raghul Somineni Raghupathy
rsr379@nyu.edu

Kushagra Agarwal
ka1745@nyu.edu

Abstract— **In this project, we try to classify songs into genres and mood based on the various properties associated with a song's lyrics and other music features. First, we perform experiments on music genre classification, exploring a variety of feature types, including semantic, sentimental and acoustic features. These experiments show that modeling semantic information contributes to outperforming strong bag-of-words baselines. And, this can be expanded for research to find additional properties associated with a song – year it was published, language etc.**

I. INTRODUCTION

The basic idea of our project is to get reliable Genre and Mood tagging of songs using the lyrics. We are going to use the millionsong dataset (MSD), which is an extensive dataset with around 1,000,000 songs along with their lyrics and other music attributes. We also use other datasets with exploratory features described in the later sections to get classification with more accuracy. We have used Naïve Bayes Classifier and Random Forest Classifier for mood classification. We have used Random forest, Ridge Classifier and Logistic regression for classifying Genres. In addition to lyrics and music features we have used Multimodal Album Reviews Dataset that consists of reviews from Amazon as exploratory features to test accuracy of the classification. Having split up the data set into training, testing and validation set, we train our model with the training set and set up a baseline for the tagging. From the results, we get from testing and validation phase, we calculate the accuracy we get from the different models we are using and conclude on the best model we feel gives more accuracy on tagging a song with its genre and mood. We aim to achieve at least 70% accuracy during the validation phase. After the successful completion of the project, we will also try to classify songs based on other experimental tags like Timeline, Language etc. Music streaming agencies such as Spotify and others have a particular need for machine learning since they have multiple sources of data that can be mined for insights. Analyzing data, for example, identifies ways to increase efficiency and save money. Machine learning can also help detect mistakes and minimize errors.

II. MOTIVATION

Classification of music into different genre has been there for more than 100 years now. Music genre is a conventional category that identifies some pieces of music in a different set which share same convention. It's used to classify different music styles into a group and people use that religiously. They are commonly used to structure the increasing amount of music available in digital form and are important for music information retrieval. The latest trends in machine learning come in handy in this regard. Also, just the thought that given the data you can extract something useful from it is already very motivating. All of these things mean it's possible to quickly and automatically produce models that can analyze bigger, more complex data and deliver faster, more accurate results – even on a very large scale. And by building precise models, a music organization has a better chance of identifying profitable opportunities – or avoiding unknown risks. Analyzing data to identify patterns and trends is key to the music industry, which relies on making classification more

efficient and predicting potential problems to increase profitability. The data analysis and modeling aspects of machine learning are important tools to music companies and organizations.

III. Related work

- **Automatic Lyrics-based Music Genre Classification in a Multilingual Setting** (https://kar.kent.ac.uk/33266/1/mgc-lyrics.pdf)
  This paper finds that there are significant challenges in preprocessing multilingual text, and that traditional techniques like stemming and stop words may actually do more harm than good in such circumstances. It also finds that classes with strong language bias may be more likely to perform better than those with multiple languages.

- **Classifying the Subjective: Determining Genre of Music from Lyrics** (http://cs229.stanford.edu/proj2012/BourabeeGoMohanClassifyingTheSubjectiveDeterminingGenreOfMusicFromLyrics.pdf)
  This paper too talks about determining genre based on lyrics and also looks into whether lyrics is sufficient for classification.

- **Song Genre and Artist Classification via Supervised Learning from Lyrics Dataset Description** (http://nlp.stanford.edu/courses/cs224n/2006/fp/sadovsky-x1n9-1-224n_final_report.pdf)
  This paper analyzes the effectiveness of our classifier at classifying lyrics by artist; EvilLyrics was again used to download lyrics of all albums for each artist in our dataset.

IV. Dataset Description

To begin with we are using the Million Song Database for our project. The Million Song Dataset is a freely available collection of audio features and metadata for a million contemporary popular music tracks. Million Song Database, along with some additional complimentary datasets provides us the user data and some song information. As our project is focused not only on lyrics, we will we using these complementary datasets.

1) Million Song Dataset-

The core of the dataset is the feature analysis and metadata for one million songs, provided by The Echo Nest. The dataset does not include any audio, only the derived features.
http://labrosa.ee.columbia.edu/millionsong/

2) Last.fm Dataset

This dataset contains official song tags, or genre tag which we will be using to predict the genre for the songs. the largest research collection of song-level tags and precomputed song-level similarity. All the data is associated with MSD tracks, which makes it easy to link it to other MSD resources: audio features, artist data, lyrics, etc.
http://labrosa.ee.columbia.edu/millionsong/lastfm

3) musiXmatch Dataset

The MXM dataset provides lyrics for many MSD tracks. The lyrics come in bag-of-words format: each track is described as the word-counts for a dictionary of the top **5,000** words across the set. Although copyright issues prevent us from distributing the full, original lyrics, we hope and believe that this format is for many purposes just as useful, and may be easier to use.

The dataset comes in two text files, describing training and test sets. The split was done according to the split for tagging, see tagging test artists. There are 210,519 training bag-of-words, 27,143 testing ones. We also provide the full list of words with total counts across all tracks so you can measure the relative importance of the top 5,000.
http://labrosa.ee.columbia.edu/millionsong/musixmatch

4) MARD- Multimodal Album Reviews Dataset

MARD contains texts and accompanying metadata originally obtained from a much larger dataset of Amazon customer reviews, which have been

enriched with music metadata from MusicBrainz, and audio descriptors from AcousticBrainz. MARD amounts to a total of 65,566 albums and 263,525 customer reviews.
http://mtg.upf.edu/download/datasets/mard

V. Algorithms

Naïve Bayes Classifier

Naive Bayes is a simple technique for constructing classifiers: models that assign class labels to problem instances, represented as vectors of feature values, where the class labels are drawn from some finite set. It is not a single algorithm for training such classifiers, but a family of algorithms based on a common principle: all naive Bayes classifiers assume that the value of a feature is independent of the value of any other feature, given the class variable.

We have chosen two NB classifiers for our classification, **Bernoulli** NB and **Multinomial** NB. For our Naïve Bayes classifier, we experimented using different sets of words as our features, as well as different size training sets to find what would be most effective in determining. Finding the right sets of words and defining the size of the training sets is the challenge we faced. Once we finalize these parameters for classification, we believe that the accuracy would go up.

Random Forest

The Random forest algorithm is based on a majority-vote among many decision trees. For T decision trees, each one has N nodes, and we randomly choose S training samples, which we use to train that tree. For each node, we randomly choose a set of F features to use to construct the decision at that node. All trees are constructed this way. To test, we take the test data, evaluate the result of each decision tree, and in the case of binary classifier, take the majority vote among all trees (in the case of multi class, the output is the mode of all trees)

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks, that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

SVM

Support Vector Machines are a set of supervised learning methods used for classification, regression and outlier's detection. We chose to classify our data using SVM because it is effective in high dimensional spaces and very versatile.

We used the NuSVC class of svm from sklearn which can perform multi-class classification on a dataset. NuSVC implements the "one-against-one" approach for classification. If n_class is the number of classes, then n_class * (n_classs-1) /2 and each one trains data from two classes.

Ridge Classifier

The next classifier we chose to classify our data was Ridge classifier available in sklearn package. The reason as to why we chose this is because text classification tends to be quite high dimensional (many features), and are likely to be linearly separable. So, linear classifiers like, ridge regression or SVM with a linear kernel, are likely to do well. In Ridge regression, the ridge parameter controls the complexity of the classifier and helps avoid over-fitting by separating the patterns of each class by large margins (i.e. the decision surface passes down the middle gap between the two collection of points. Ridge regression also avoids over-fitting by regularizing weights to keep them small, and model selection is straight forward as you have only access to the value of a single regression parameter.

VI. Result

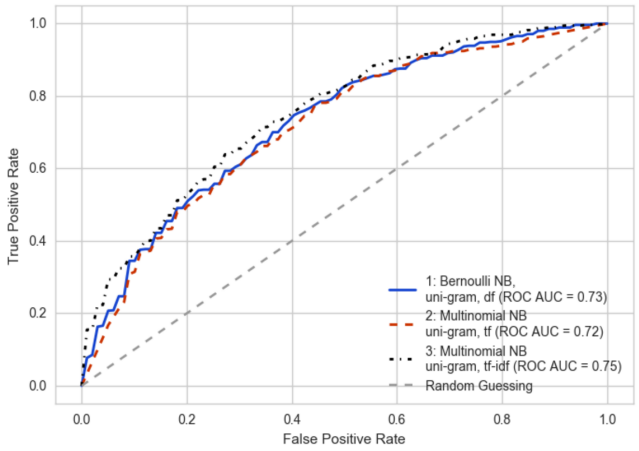
Mood Classification

Mood classification is about deciding whether the songs are happy or sad. These were done by analyzing lyrics data of the songs and comparing them against positive and negative bag of words.

Now, we used two classifiers to do the same, the results for both are given below:

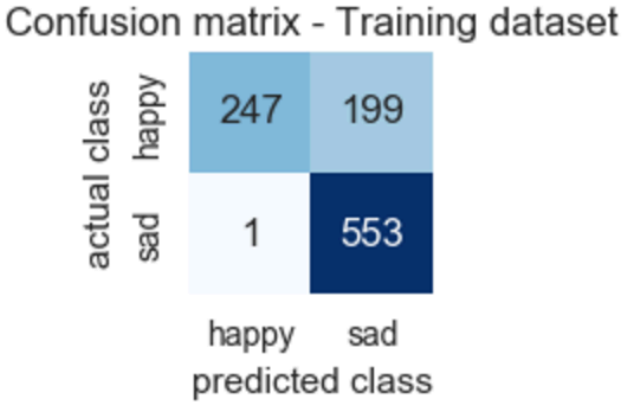
Naïve Bayes' Classifier – Bernoulli and Multinomial

The ROC is plotted for three different models of classifiers:
1. Bernoulli NB with CountVectorizer
2.Multinomial NB with Count Vectorizer
3. Mutinomial NB with TfidfVectorizer

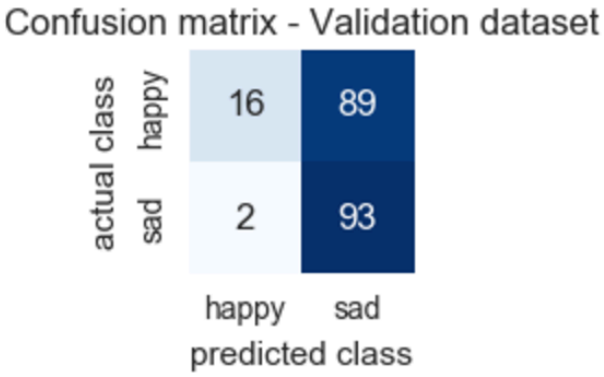


From the ROC AUC we chose the Mutinomial NB with TfidfVectorizer as the better among the three and continue with validation.

The confusion matrix of the chosen classifier when run on the Training dataset can be seen below:



When the same as been run on the Validation dataset, the confusion matrix is as follows:



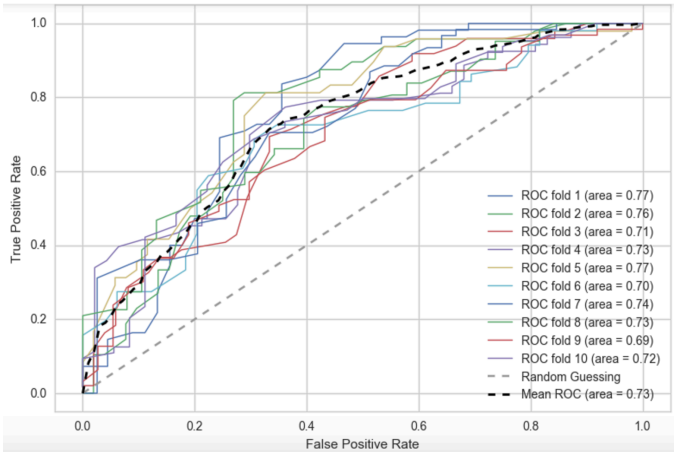
The performance of the NB classifier we used to classify the mood of the song can be seen below:

	ACC (%)	PRE (%)	REC (%)	F1 (%)	ROC AUC (%)
Training	80.0	99.60	55.38	71.18	77.60
Validation	54.5	88.89	15.24	26.02	56.57

As you can seem we have achieved only 54.5% accuracy on the validation data set.

Random Forest Classifier

For the Random Forest classifier, we had about 11 models to choose from varying by the usage of CountVectorizer or TfidfVectorizer. Also, for each Vectorizer chosen we varied the tokenizer used. The tokenizers being raw words, Porter stemmed, snowball stemmed.



For all the models chosen, metrics during the training phase were:

	ACC (%)	PRE (%)	REC (%)	F1 (%)	ROC AUC (%)
Train CountVec	99.8	99.78	99.78	99.78	99.80
Train CountVec porter	99.8	99.78	99.78	99.78	99.80
Train CountVec snowball	99.8	99.78	99.78	99.78	99.80
Train CountVec wl	99.5	99.77	99.10	99.44	99.46
Train CountVec porter+wl	99.5	100.00	98.88	99.44	99.44
Train CountVec snowball+wl	99.5	100.00	98.88	99.44	99.44
Train TfidfVec	99.8	99.78	99.78	99.78	99.80
Train TfidfVec porter	99.8	100.00	99.55	99.78	99.78
Train TfidfVec snowball	99.8	99.55	100.00	99.78	99.82
Train TfidfVec wl	99.4	99.77	98.88	99.32	99.35
Train TfidfVec porter+wl	99.4	99.77	98.88	99.32	99.35
Train TfidfVec snowball+wl	99.4	99.77	98.88	99.32	99.35

Though the accuracy seems quite promising, during the validation testing, the accuracy gained was about 72%.

Genre Classification

Genre classification is to classify the data among 11 genres that were present on our dataset. Now the classifiers used had to analyze the lyrics, acoustic data and semantic data to come to conclusion. For this we used about three classification techniques, the results of the same are given below.

Ridge Classifier

As discussed in the Algorithms section, Ridge Regression classifier was very suitable in analyzing high dimensional data. We used the RidgeClassifier present in the sklearn module of python to get on with our training and classification. The results for the same are given below:

```
================================================================================
Ridge Classifier
________________________________________________________________________________
Training:
train time: 1.376s
test time:  0.010s
accuracy:   0.604
dimensionality: 133478
density: 1.000000
```

```
                    precision    recall  f1-score   support

  Alternative Rock       0.25      0.25      0.25        20
         Classical       0.63      0.95      0.76        20
           Country       0.50      0.70      0.58        20
Dance & Electronic       0.44      0.40      0.42        20
              Folk       0.83      0.25      0.38        20
              Jazz       0.78      0.90      0.84        20
       Latin Music       0.87      0.65      0.74        20
             Metal       0.59      0.80      0.68        20
          New Age       0.86      0.60      0.71        20
               Pop       0.56      0.75      0.64        20
               R&B       0.50      0.55      0.52        20
    Rap & Hip-Hop       0.71      0.75      0.73        20
             Rock       0.67      0.30      0.41        20

      avg / total       0.63      0.60      0.59       260

confusion matrix:
[[ 5  1  0  3  0  0  0  3  0  1  4  1  2]
 [ 0 19  0  0  0  0  1  0  0  0  0  0  0]
 [ 0  0 14  0  0  1  0  0  0  4  1  0  0]
 [ 5  0  0  8  0  1  0  1  0  1  2  2  0]
 [ 2  3  6  0  5  1  0  1  1  1  0  0  0]
 [ 0  0  2  0  0 18  0  0  0  0  0  0  0]
 [ 1  4  0  1  0  0 13  0  1  0  0  0  0]
 [ 2  0  1  0  0  0  0 16  0  0  0  0  1]
 [ 0  2  0  0  1  2  1  1 12  1  0  0  0]
 [ 0  0  2  1  0  0  0  0  0 15  1  1  0]
 [ 1  0  0  2  0  0  0  0  0  4 11  2  0]
 [ 0  0  0  2  0  0  0  1  0  0  2 15  0]
 [ 4  1  3  1  0  0  0  4  0  0  1  0  6]]
```

As you can see, an accuracy of 60% has been achieved using this classifier for training and validation.

Random Forest Classifier

The next classifier used in genre classification is Random forest. We used the RandomForestClassifier in the sklearn module of python. The results for the same are below:

```
================================================================================
Random forest
________________________________________________________________________________
Training:
train time: 11.443s
test time:  0.196s
accuracy:   0.554
```

```
classification report:
                    precision    recall  f1-score   support

  Alternative Rock       0.35      0.30      0.32        20
         Classical       0.72      0.90      0.80        20
           Country       0.59      0.65      0.62        20
Dance & Electronic       0.55      0.60      0.57        20
              Folk       0.82      0.45      0.58        20
              Jazz       0.63      0.85      0.72        20
       Latin Music       0.75      0.75      0.75        20
             Metal       0.60      0.90      0.72        20
          New Age       0.92      0.55      0.69        20
               Pop       0.50      0.70      0.58        20
               R&B       0.63      0.60      0.62        20
    Rap & Hip-Hop       0.90      0.95      0.93        20
             Rock       0.50      0.15      0.23        20

      avg / total       0.65      0.64      0.63       260
```

```
confusion matrix:
[[ 6  0  0  2  1  0  1  5  0  2  1  0  2]
 [ 0 18  0  0  0  1  1  0  0  0  0  0  0]
 [ 2  0 13  0  0  1  0  1  0  1  2  0  0]
 [ 2  1  0 12  0  2  0  0  1  1  1  0  0]
 [ 2  1  1  1  9  3  0  1  0  2  0  0  0]
 [ 0  1  0  0  0 17  0  1  0  1  0  0  0]
 [ 0  0  0  1  0  0 15  0  0  2  2  0  0]
 [ 1  0  0  1  0  0  0 18  0  0  0  0  0]
 [ 0  3  0  1  1  0  0  0 11  3  0  1  0]
 [ 0  1  2  0  0  0  1  0  0 14  1  0  1]
 [ 2  0  2  0  0  1  1  0  0  2 12  0  0]
 [ 0  0  0  1  0  0  0  0  0  0  0 19  0]
 [ 2  0  4  3  0  2  1  4  0  0  0  1  3]]
```

As you can see, we have achieved an accuracy of 55.4 % using the Random Forest classifier.

SVM:

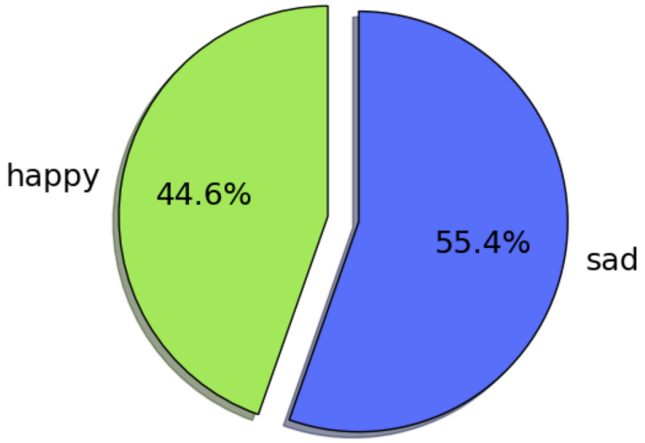
The next classifier being SVM, we used the svm module in sklearn package of python. The results for the same are below:

```
0.2 80.0
Pop accuracy: 76.66666666666666%
Jazz accuracy: 77.86666666666666%
Metal accuracy: 80.0%
Classical accuracy: 86.66666666666667%
Hiphop accuracy: 93.33333333333333%
```

As you can see, svm has classified songs genre-specifically with a very good accuracy.

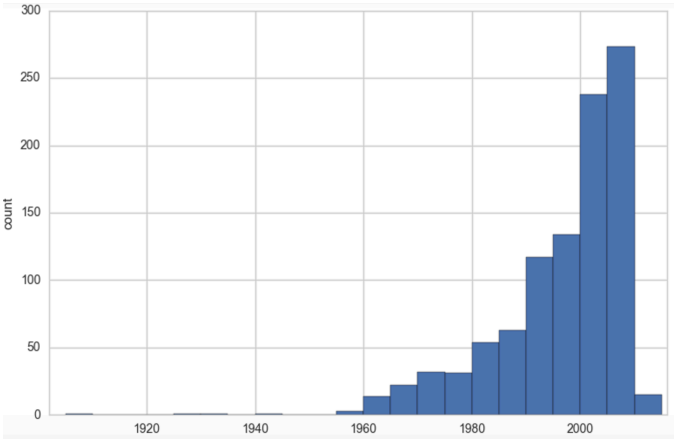
Exploratory Analysis of Lyrics Data

Result 1



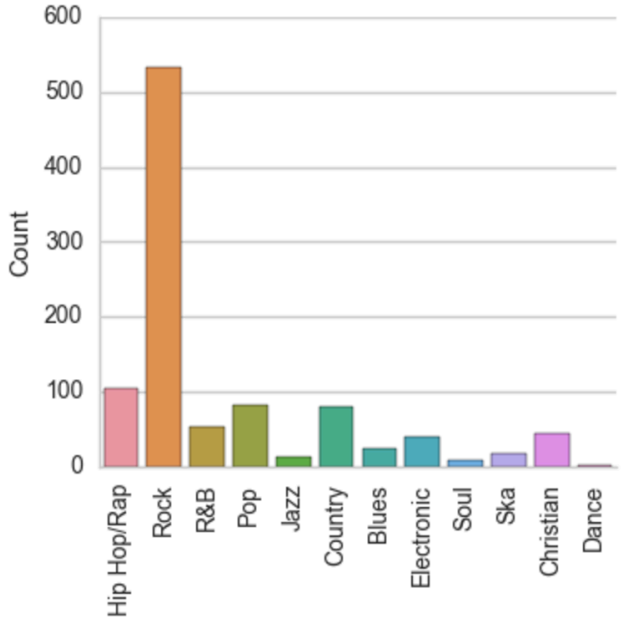
Out of all the Lyrics data we had, a quick exploratory analysis had an interesting result that shows slightly higher proportion of Sad songs.

Result 2



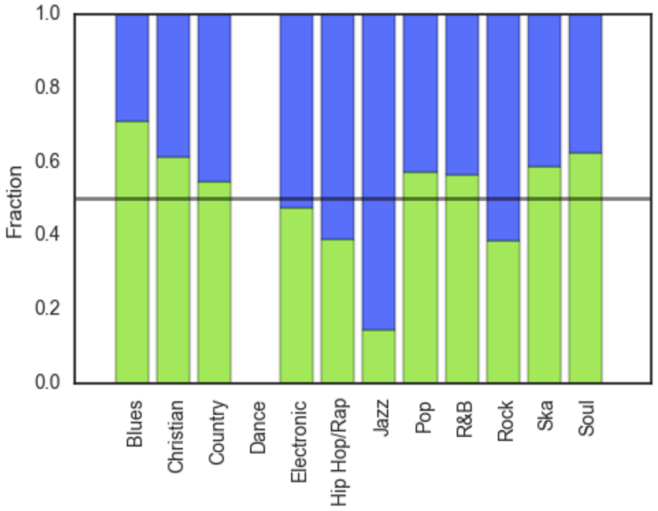
This graph shows the trend of songs that we have considered for the given period. We have used a reduced data set that matches the overall dataset properties.

Result 3



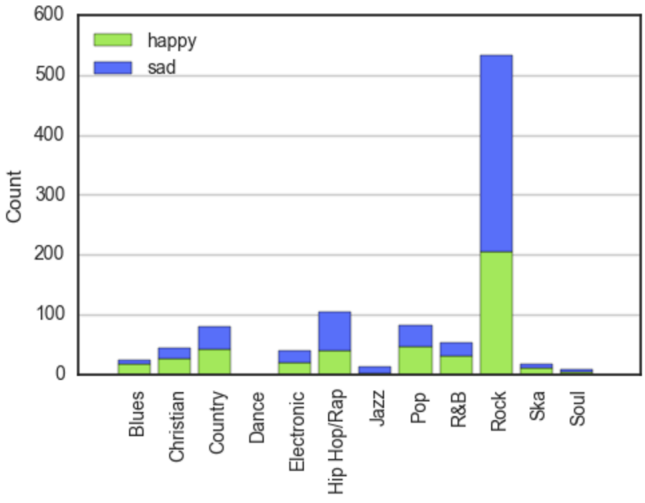
This graph shows the distribution of genres. An interesting observation is that Rock songs seem to dominate in number. This graph is reflective of the larger dataset although we have used a reduced one for this graph.

Result 5



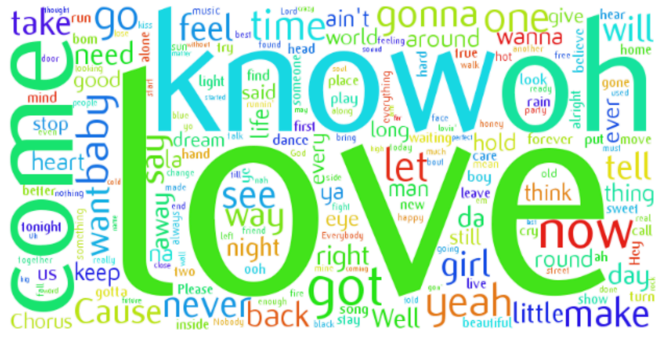
This graph shows the ratio in terms of the percentages of happy and sad songs. While Jazz seems to be largely made of sad songs, Blues and Souls fans seem to enjoy happier songs.

Result 4



This is an extension of the previous visualization. Here we see the split between happy and sad songs. In Rock, we see a higher proportion of sad songs.

Result 6



The final graph depicts the word frequency in the lyrics we had. As expected “love” seems to dominate other words by far. Also, interestingly the expression “oh” also frequently appears in more songs.

VII. Code

https://github.com/arung5305/MusicClassify

VIII. Video link

Channel Link (in case video link doesn't work)

https://www.youtube.com/channel/UCnaKmb_njbue5xvIPPlBnJg

Please copy the whole link.

Video Link
https://www.youtube.com/watch?v=nbePjRqFO5I&feature=youtu.be

IX. Evaluation

We set out to analyze lyrics based genre classification. However, we realized through many trials that just lyrics will not give us reliable genre information. We could however use the lyrics for other analysis like mood classification.

We also did exploratory analysis with our data to find interesting results. We added music features to our dataset to get reliable genre classification. We further added Amazon reviews dataset which was merged later to the existing dataset. We have finally achieved an accuracy of 0.62 for Ridge Classifier and an accuracy averaging 80% with SVM.

Therefore, we see the difficulty in classifying genre based on all these features. We believe, given more time we could have arrived at a higher accuracy with adding other features from other datasets. There is a lot of potential for Songs Classification since many datasets are available publicly today.

Many more attributes like artist classification/prediction, timeline prediction is achievable with this dataset. Although we didn't have the opportunity to pursue it during this project due to lack of time, we plan to continue the research outside of this class project in these regards.

X. Conclusion

We have achieved the required accuracy in the classifiers used and we have concluded that just lyrics is not sufficient for genre classification. However, we can predict the mood of the songs with high accuracy with just Lyrics. We have also found interesting trends during our exploratory analysis of our data. We believe that these results can be used for analysis of larger data to extract more useful data. There is lot of potential for more exploratory analysis given more features.