You’ll submit a partial report (3-5 pages) and implementation of your project. This should include:

- Evidence that you’ve been able to obtain, load, and play around a bit with your data. (For example, some simple exploratory data analysis.)
- Results from a baseline model. This can be very simple, such as random predictor, most-common-class, or a bag-of-words model.

Your report should be the working/rough draft of your final project report (see below), although it’s expected that you won’t have fleshed-out results or conclusion sections. It’s also okay if your report changes substantially between here and the final, especially if you have exciting results in the interim!

For the milestone, your writeup should have sections similar to the following, in the vein of a proper research paper:

0. Abstract

In this paper, we analyze mood classification based on song lyrics. Mood classification of lyrics can help in the creation of automatic playlists, a music search engine, labeling for digital music libraries, and other recommendation systems.

We compare the accuracy of deep learning vs traditional machine learning approaches for classifying the mood of songs, using lyrics as features. We will use mood categories derived from Russell’s model of affect (from psychology, where mood is represented by vector in 2D valence-arousal space) and also calculate a valence-happiness rating. Mood categories will likely be happiness, anger, fear, sadness, and love (perhaps surprise, disgust).

We will test the extensibility of our deep learning model through genre classification, song quality prediction / album ratings, and additional text features such as part-of-speech tags, number of unique and repeated words and lines, lines ending with same words, etc.

The dataset we will use is the Million Song Dataset (MSD), a freely available million contemporary music track dataset. From MSD, we will use the Last.fm and musiXmatch

datasets for song tags and lyrics. We will also use language detection to focus on English lyrics.

Algorithms we are considering in addition to RNN can be naive bayes, KNN, binary SVM, n-gram models like topK.

• 1. Introduction (motivation for your work)

Music recommendation engines are becoming more and more expected norms for listeners and music lovers today. Since mood and emotional classification is crucial to the entertainment value of songs, music recommendation algorithms can take into account mood classification. Expert, and even layman, human assessment is still often superior to machine learning methodology, however as technology improves, appropriately classified mood labels can greatly ease this process, and can even be used as metadata in digital music libraries and repositories.

Much research on music classification is based on audio analysis versus lyric analysis. (Corona). However, classifiers that incorporate textual features can outperform audio-only classifiers. (Fell)

In this paper, we focus on the analysis of lyrics for classifying mood categories. We use mood and emotion as interchangeable concepts.

Song lyrics are different from ordinary text in that they often use more stylistic qualities like rhyming, and other forms, often contributing to the emotional value. (Yang) They are also shorter and often have smaller vocabularies than other text documents. This combined with their more poetic style metaphorism can cause more ambiguity when it comes to mood identification (Cano)

Classifying mood (or “sentiment”) using textual features has been studied less than musical features. One reason may be in obtaining a large dataset legally (as lyrics are copyrighted material).

We use the Million Song Dataset (MSD), a large freely-available dataset, and supplement it with lyrics scraping we conducted separately.

Using this created dataset, which can provide a basis for future studies on lyrics-based mood classification, we measure the effectiveness of different machine learning models on mood prediction.

This paper is organized as follows: Section 2 reviews related work on lyrics based mood classification. Section 3 outlines the methods (design and implementation) used in our analysis. Section 4 discusses results (include plots & figures), including a detailed analysis of these obtained results. Conclusion and recommendations are summarized in Section 5.

A clear statement of the problem you are trying to solve.

Novelty of your approach to the problem.

"How will I know when my project is successful?", usually in the form of an evaluation metric.

Tied back to overall problem statement - why is this the right objective?

Statement of a clear baseline (e.g. for classification, predict most common class for everything).

- ## 2. Background (literature review, or related work)

Previous work reached contradictory conclusion and employed smaller datasets and simpler methodology like td-idf weighted bag-of-words to derive unigrams, fuzzy clustering, and linear regression. Our proposed model approach (RNN) should have better accuracy.

Other research has focused on using joint music-lyric models (Mihalcea). Using SVM, classification accuracy between lyrics only, audio only, and joint lyric and audio classifiers found that joint is most accurate.

Similar results were found in other research using SVM implementation (Hu), though not all emotion categories saw combined feature sets outperforming lyric or audio only features.

Our research's focus on lyrics only is akin to that of music classification research also based on the MSD, which showed that lyrics alone can be used to classify music mood, achieving accuracy 70% of time. (Corona) SVM classification algorithms were used, and some moods (like anger) were much easier to detect. One downside is that the dataset still could use improvement as the small scale imposes limitations on the methodology.

Our choice of RNN is based on research demonstrating that NN-based systems can often be transferrable to other languages more easily (Becker)

Genre classification based on lyrics has been studied using shallower textual features like bags-of-words. A further study used n-gram modeling to predict one of 8 genres, by honing in on the topic of the text, examining features like length, use of pronouns, past tense, repetitive structures, rhyme features, etc. (Fell) Certain genres, like rap, have more unique properties (e.g., complex rhyme, long lyrics, distinctive vocab), that make it easier to classify than more difficult and frequently confused genres like Folk, Blues, and Country. These often confused genres share similar topics and are stylistically and structurally similar. In these instances, audio and musical properties may be better than lyrics at accurate prediction.

Though our focus is on English language lyrics, there have been studies conducted on lyrics in other languages, such as Chinese (Chen) and Hindi (Das), that also used mood affects predicted based on combinations of text stylistic features and features based on n-grams (e.g., term frequency-inverse document frequency scores of trigrams.

Not all emotional attributes are the same.

A recent study on sentiment analysis of lyrics showed that music corresponding to happy moods can be predicted with greater accuracies. (Raschka) This study used a naive bayes classifier trained on lyrics alone, also pulled from the Million Song Dataset. Specifically, their best performing model was a multinomial naive Bayes classifier with unigram tf-idf feature representation, better than the Bernouilli naive bayes model. Increasing the n-gram range and other tuning such as parameter smoothing had little effect on classification performance.

Another angle is looking at happiness (emotional states), which found that the happiness of song lyrics has been trending downward since the 60s though stable within genres (Dodds)

Who else has worked on this problem? What did they do? What are you doing differently?

Is it clear you read and understood the papers you cited? Are they the right ones?

Is this the objective others working on this or similar problems use? If not, why?

- 3. Methods (design and implementation)

Are you using appropriate techniques for the problem you're trying to solve?

What did you learn from the first approaches you made? How did you take what you learned and use that to improve subsequent iterations?

What weird patterns in the data have you found? What about how the model interacts with your data? (Did you find any interesting loss patterns?)

Do the patterns you observe align well with patterns other researchers have found in the papers you read? What did they do about them?

Data Acquisition

XXX songs were downloaded from the Million Song Dataset (MSD). This dataset provides song - mood mapping at a song but not lyric level. A component of the MSD includes Last.fm, which maps songs to user “tags” based on Last.fm user input. These tags can be anything from a human emotion to genre to animals. Another is the MusiXmatch dataset, which is a collection of song lyrics mapped to song in a bag-of-words format.

The LastFM dataset contains song-level tags for more than 500,000 songs. The mood categories are derived using the social tags found in this dataset. The MusixMatch dataset contains lyrics for 237,662 songs. Each song is described by word-counts of the top 5,000 stemmed terms across the set.

Pre-Processing: Scraping, Indexing, and Labeling Lyrics

One initial challenge was in acquiring corresponding lyrics to song data. To obtain the 70,000+ lyrics in our dataset, we used the python package lyricsgenius for retrieving lyrics. The package interfaces with the www.genius.com API for lyric access.

We attempt to match songs on all combinations of the MSD song title, MSD artist name, MXM song title, and MXM artist name.

After scraping and downloading lyrics into txt files, we next index the files and perform basic checks on the validity of each. The checks include:

Does a downloaded lyric text file exist? We remove repeated songs, e.g., songs with the same title and lyrics, but different dataset ids.

Are the lyrics in English?

We also applied language filtering rules to remove non-English lyrics. Some of the files have section tags (Verse, Chorus, etc) and some have guitar chords; we removed guitar chords, and kept "chorus" for our model to use as an indicator of genre

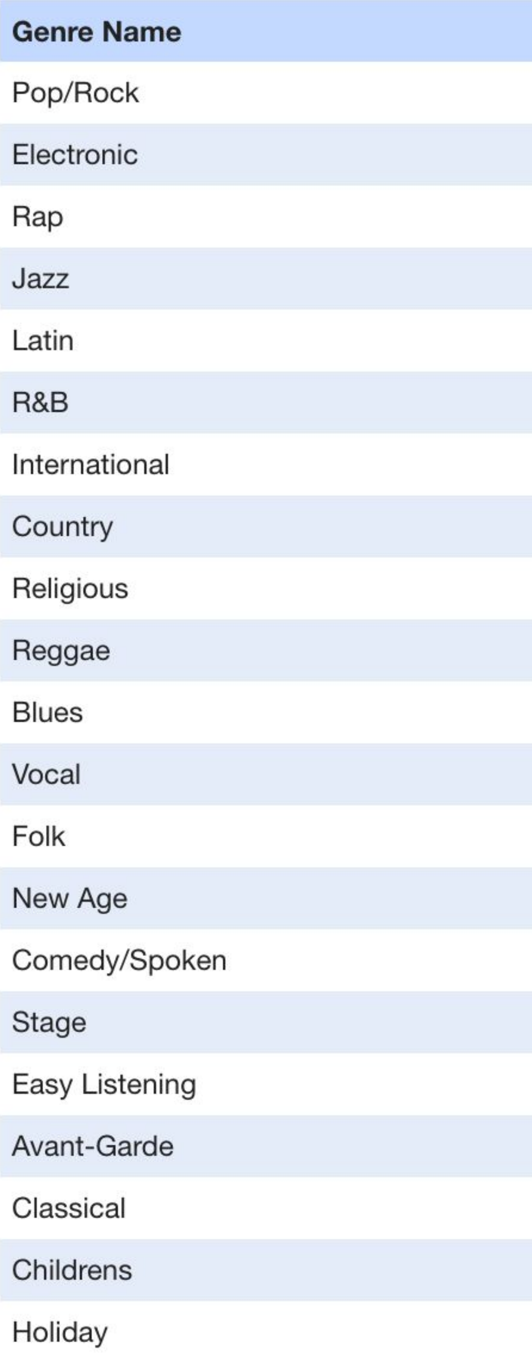
What is the total word count?

Now that we have a nice index built, we can easily match the lyrics to the mood tags from the last.fm dataset. To do this, we iterate over each row of the index, query the sqlite Last.fm database for all associated tags, then attempt to match tags against our Mood Categories.

Basically, we create an index csv with metadata for each song like english yes/no, lyrics yes/no, wordcount, and also mood. The end result is a *labeled_lyrics.csv* that should make it very easy to work with the lyrics dataset

Genre Classification

We use the MSD Allmusic Genre Dataset (MAGD), which contains all genre labels collected from Allmusic.com. This includes generic genres like Religious or Christmas, non-musical content like Comedy/Spoken and significantly small genres. (Schreiber):



Genre Name
Pop/Rock
Electronic
Rap
Jazz
Latin
R&B
International
Country
Religious
Reggae
Blues
Vocal
Folk
New Age
Comedy/Spoken
Stage
Easy Listening
Avant-Garde
Classical
Childrens
Holiday

The remaining XXX songs are then randomly divided into a training set (XX songs), dev set, and test set

Mood Annotation / Taxonomy

We use the popular Russell’s model of affect and focus on 18 mood classes similar to previous studies. (Hu)

In Russell’s model, mood is represented by a vector in the 2-D valence-arousal space where mood is an element of (valence, arousal): $m \varepsilon M = (v, a)$

Valence measures the good vs bad dimension (pleasant / unpleasant)

Arousal measures the active vs passive dimension of sentiment (aroused / sleepy).

Mood in our model refers to categories to be learned in the classification problem. We group mood into 18 categories:

'calm': ['calm', 'comfort', 'quiet', 'serene', 'mellow', 'chill out']
'sad': ['sadness', 'unhappy', 'melancholic', 'melancholy']
'happy': ['happy', 'happiness', 'happy songs', 'happy music']
'romantic': ['romantic', 'romantic music']
'upbeat': ['upbeat', 'gleeful', 'high spirits', 'zest', 'enthusiastic']
'depressed': ['depressed', 'blue', 'dark', 'depressive', 'dreary']
'anger': ['anger', 'angry', 'choleric', 'fury', 'outraged', 'rage']
'grief': ['grief', 'heartbreak', 'mournful', 'sorrow', 'sorry']
'dreamy': ['dreamy']
'cheerful': ['cheerful', 'cheer up', 'festive', 'jolly', 'jovial', 'merry']
'brooding': ['brooding', 'contemplative', 'meditative', 'reflective']
'aggression': ['aggression', 'aggressive']
'confident': ['confident', 'encouraging', 'encouragement', 'optimism']
'angst': ['angst', 'anxiety', 'anxious', 'jumpy', 'nervous', 'angsty']
'earnest': ['earnest', 'heartfelt']
'desire': ['desire', 'hope', 'hopeful', 'mood: hopeful']
'pessimism': ['pessimism', 'cynical', 'pessimistic', 'weltschmerz']
'excitement': ['excitement', 'exciting', 'exhilarating', 'thrill', 'ardor']

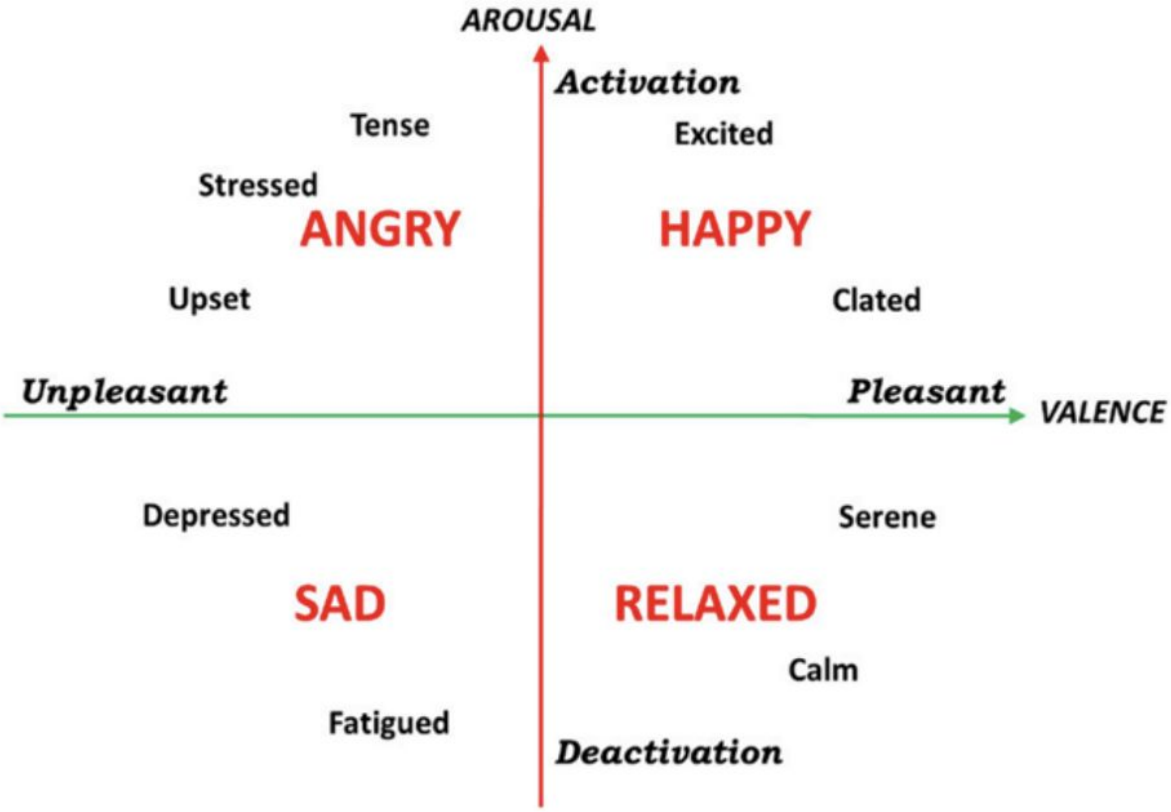
Each of the mood categories represent one of the 4 quadrants in a 2-D Euclidean plane.

The most frequent of sentiments is love, which is expected, as most songs are about love in some fashion

Table below shows the mood tags, groups, and quadrants used. For example, the song XX by XX is tagged as “xx” in the LastFM dataset and is included in group XX and quadrant XX

A song has to be tagged at least twice with one term in a tag group, or with at least two terms in a tag group, each at least once.

IMAGES PLACEHOLDERS



Stylistic features: such as the number of unique words, number of repeated words, number of lines, number of unique lines and number of lines ended with same words were considered in our experiments.

Model Training and Tuning

Feature Extraction Via Bag of Words BOW

Prior to the tokenization of the lyrics, a bag of words model was used to transform the lyrics into feature vectors. Further processing of the feature vectors include the choice of different n-gram sequences $n \varepsilon \{1, 2, 3\}$

BOW is an unordered model where each word is assigned a value representing the frequency of the word (tf-idf weight). We did not employ stemming (process of merging words with same morphological roots, which has shown mixed effects in text classification) (Hu)

Term-Weighting

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

$$tf - idf(t,\ d)\ =\ tf(t,\ d)\ x\ idf(t)$$

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) is the normalized term frequency and idf(t) be the inverse document frequency, where nd is the total number of lyrics and df(d, t) the number of lyrics that contain the term t

$$idf(t)\ =\ log\ ((1\ +\ nd)\ /\ (1\ +\ df(d,\ t)))\ +\ 1$$

- Build word embeddings from song lyrics themselves. Do not use Brown or something like that. Check out BERT from Google as a model to build the word embeddings. Dan say's they are currently offering a publicly available Colab notebook that runs on a TPU accelerator for _free_.

- LSTM might suffer from performance on whole songs. Maybe consider a CNN instead. Maybe split the songs into lines and just apply the same label for all lines. Also look into document classification.

- Train an LSTM network with the train dataset and evaluate its performance on the dev dataset.

- Adjust parameters as needed. Evaluate performance on the test dataset.

- Sample the trained model to produce lyrics of a specified mood.

We also observe sections of songs, as many songs express a wide range of moods over the course of the song

We compare XX classifiers: KNN, SVM, Naive Bayes

The XX classifier performed best

XX achieved classification accuracy of XX%

- Next Steps section for work you plan to do before submitting the final version (you'll remove this section and replace it with your conclusions, final results and analysis in your final report)
- ## Results and discussion (include plots & figures, and detailed analysis in comparison to baseline and the literature, if applicable)
- ## Conclusion

Lyrics based mood classification is just beginning. But as more music moves online, intelligent and accurate music recommendation systems will be greatly enhanced by the use of lyrics to aid prediction. As larger datasets become more readily available, the real-life applicability of these algorithms will become easier to vet. Also in the future, we may want to do some analysis on a few songs that looks at the verse by verse mood and then relate that to the overall mood (e.g., are sad songs mostly sad all the time or do they tend to have on average X% happy verses).

Paper References:

1. DONE Bandyopadhyay, Sivaji, Das, Dipankar, and Patra, Braja Gopal. (2015). Mood Classification of Hindi Songs based on Lyrics.
2. OPTIONAL Becker, Maria, Frank, Anette, Nastase, Vivi, Palmer, Alexis, and Staniek, Michael. (2017). Classifying Semantic Clause Types: Modeling Context and Genre Characteristics with Recurrent Neural Networks and Attention.
3. SEMI: Cano, Erion and Morisio, Maurizio. (2017). “MoodyLyrics: A Sentiment Annotated Lyrics Dataset.”
4. DONE: Chen, Xiaoou, Hu, Yajie and Yang, Deshun. (2009). “Lyric-based Song Emotion Detection with Affective Lexicon and Fuzzy Clustering Method.”
5. DONE Corona, Humberto and O’Mahony, Michael. (2015). An Exploration of Mood Classification in the Million Songs Dataset.
6. DONE Danforth, Christopher M. and Dodds, Peter Sheridan. (2009). Measuring the Happiness of Large-Scale Written Expression: Songs, Blogs, and Presidents
7. DONE Downie, J. Stephen, Ehmann, Andreas F., and Hu, Xiao. (2009). Lyric Text Mining in Music Mood Classification.
8. Good DONE Fell, Michael and Sporleder, Caroline. (2014). Lyrics-based Analysis and Classification of Music.
9. DONE: Lee, Won-Sook and Yang, Dan. (2010). Music Emotion Identification from Lyrics.
10. DONE Mihalcea, Rada and Strapparava, Carlo. (2012). Lyrics, Music, and Emotions.
11. OPTIONAL Manning, Christopher D. and Wang, Sida. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification.
12. Good DONE: Raschka, Sebastian. (2014). MusicMood: Predicting the Mood of Music from Song Lyrics Using Machine Learning.
13. Russell, J. A. (1980). “A Circumplex Model of Affect,” Journal of Personality and Social Psychology, vol. 39, no. 6.
14. Schreiber, Hendrik. (2015). Improving Genre Annotations for the Million Song Dataset.
15. Bertin-Mahieux, T, Ellis, D, Lamere, P, and Whitman, Brian. (2011). “The Million Song Dataset.”