COMP9444 Project

Lyrics Mood Analysis

Term3 2022

Calvin Long(z5255352)
Alon Moss(z5160732)
Rongbo Zhao(z5286178)
Zhengye Ma(z5158505)
Hong Zhang(z5257097)

Introduction,Motivation and/or Problem Statement:

As time-poor students with a passion for music, we have identified an opportunity to combine this passion with our knowledge of Neural Network models.

Currently, if we wanted to curate a playlist of songs to reflect our mood, we would have to painstakingly listen to the music, analyse the lyrics and classify the song's emotion. Additionally, this arduous process limits the capacity for music to act as a comfort as our energy would be better spent on listening to the mood-based playlist instead of creating it. As such, our team has developed the problem statement of,

"As there currently does not exist a system that analyses and classifies the emotions of a song using its lyrics, we should create one to alleviate time pressure and maximise the utility of listening to music."

Data Sources or RL Tasks:

When approaching this task, the team understood that the input data source required would need to fulfil two requirements. Firstly, the text itself needed to be somewhat colloquial in order to reflect the fact that song lyrics do not follow all the rules of grammar. Additionally, we needed the input text data to be matched to a consistent model of human emotion.

We identified two different input files in order to train and test the multiple models that were developed. The first was sourced from Kaggle, under “The Emotions Dataset for NLP”, which contains 20,000 inputs (2,000 for testing, 16,000 for training and 2,000 for validation). This dataset was framed in the first person, contained common colloquialisms - for example, “I am not a people person but for some fu**because people feel that they can bore me with their petty garbage”. While we don’t condone the use of swear words, their prevalence within song lyrics cannot be denied, therefore increasing the utility for this dataset. The second input file that was used was the GoEmotions Dataset which was developed by the Google Research team and includes just under 50,000 inputs (5,000 for testing and 43,000 for training). In order to replicate the colloquialism, this dataset was sourced from reddit comments, giving it a more human feel, for example, “I WaS KIddInG i loVE U LOL ". In addition to both files having human-like text, both of them classify this text in line with Eckman’s Emotion Theory, which states that the six basic human emotions are; Happiness, sadness, fear, disgust, anger and surprise. As such, both datasets fulfil the two requirements listed previously.

Beyond a requirement for input data, two of the models (Text CNN and BiLSTM) required additional data in the form of word vectors. Global Vectors for Word Representation (GloVe) was utilised as well as FastTexts’s English Word Vectors.

FInally, once we had developed all the separate models, we needed a data source that would be able to provide lyrics for popular songs in a text format. For this we utilised MusixMatch, which contains a database of over 8 million song lyrics.

Exploratory Analysis of Data or RL Tasks:

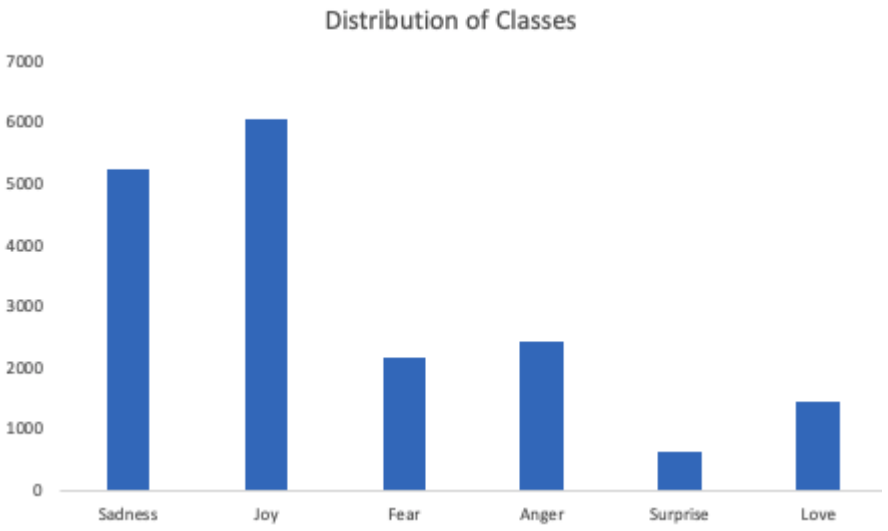
Properties

The Emotions Dataset for NLP

Source: https://www.kaggle.com/datasets/praveengovi/emotions-dataset-for-nlp

The Emotions Dataset for NLP consists of 20,000 classified text inputs, all coming from the first person perspective. The outputs are text based, with them being; Sadness, Anger, Joy, Fear, Surprise, Sadness and Love.

The distribution of these classifications is evident below:



This dataset also contains 16,184 unique words. Unfortunately, there is limited information on the collection methodology from the download page on Kaggle.

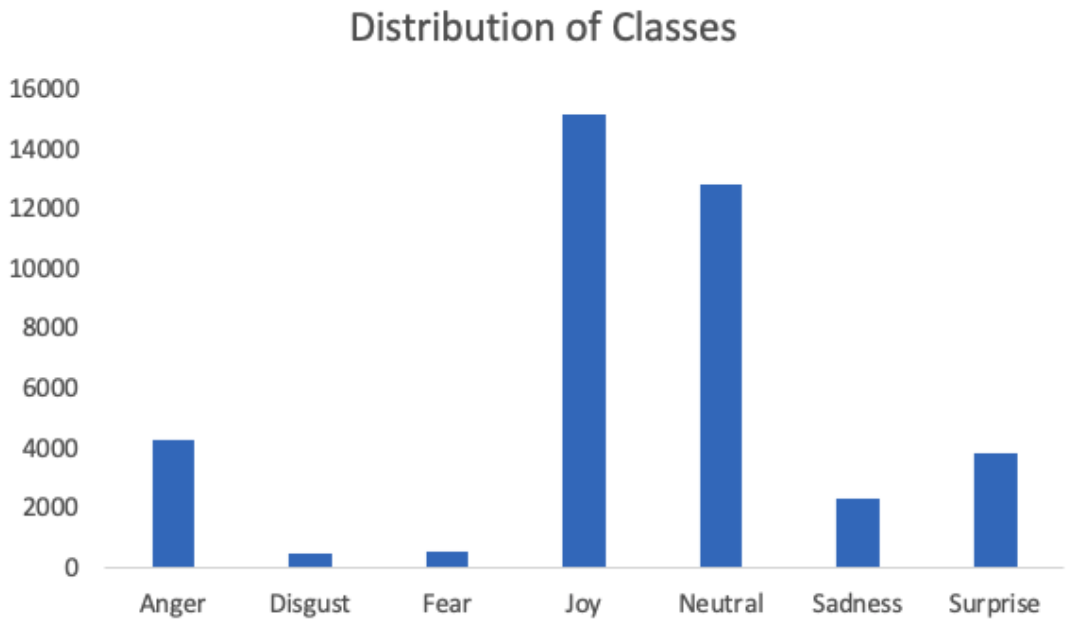
GoEmotions Dataset

Source: https://github.com/monologg/GoEmotions-pytorch/tree/master/data/ekman

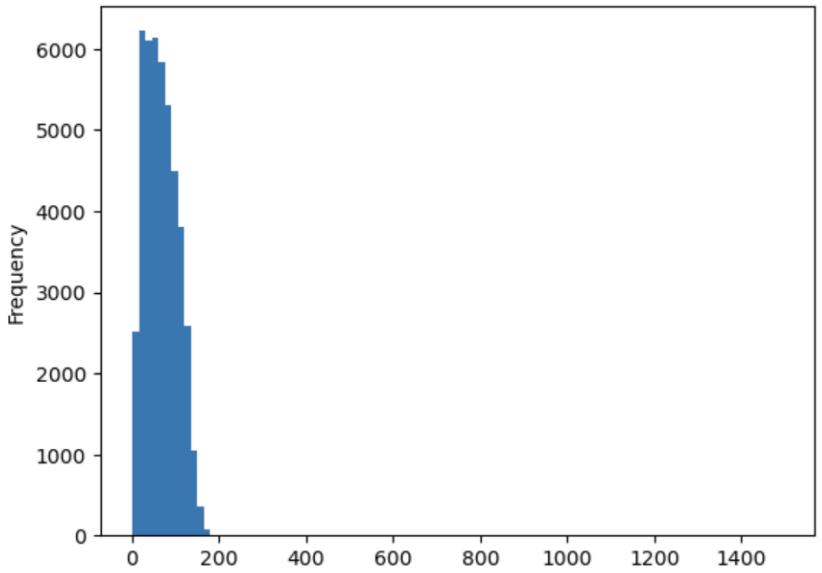
The GoEmotions dataset consists of just under 50,000 classified text inputs. The classifications are outputted as numbers between 0 and 6, each representing an emotion.

Output	Emotion
0	Anger
1	Disgust
2	Fear
3	Joy
4	Neutral
5	Sadness
6	Surprise

The distribution of each of these classes within the dataset is as follows:



Additionally, the dataset contains 35,385 unique words, as well as an input length distribution seen below:



On a more qualitative level, it is worth discussing that as the source of the data is Reddit, which skews towards a young male audience, there is the capacity that the data does not accurately portray other, more unique viewpoints. This could impact our overall model as the data we are inputting (song lyrics) are not all going to be coming from young male creators. However, the Google Research team has mitigated these effects by filtering and masking out any content which pertains to a specific identity (evident through the use of [NAME]) as well as taking the data source from a balanced selection of subreddits.

Pre-processing required

For any NLP model, three main steps of preprocessing are required. Each model completed the preprocessing slightly differently, but ultimately with the same goal.

Tokenization

As the input for all the models are sentences, the first step that needs to be completed is reducing these sentences into words. This is important so that

the algorithm can utilise individual understanding of the words to assign a larger meaning to the sentence.

Cleaning

As the data used as inputs for the model are coming from online sources (such as reddit) and also potentially contain incorrect spellings, or capitalisation, some form of cleaning has to be done. On a general level this will look like lowercasing the words, removing any contractions (i.e. transforming I'm into I am). This will allow consistency in the understanding of the inputs from the algorithms perspective.

Word Embeddings

Word embeddings are utilised to transform individual words into vectors, where the relationship between words is identified through relationships in the vectors. For one of the models, Global Vector for word representation (GloVe) was used. GloVe creates word embeddings on a global level, rather than local (like Word2Vec). For the other model, FastText which builds upon the Word2Vec skip-gram model and is a local based system.

GloVe: https://www.kaggle.com/datasets/takuok/glove840b300dtxt

FastText: https://fasttext.cc/docs/en/english-vectors.html

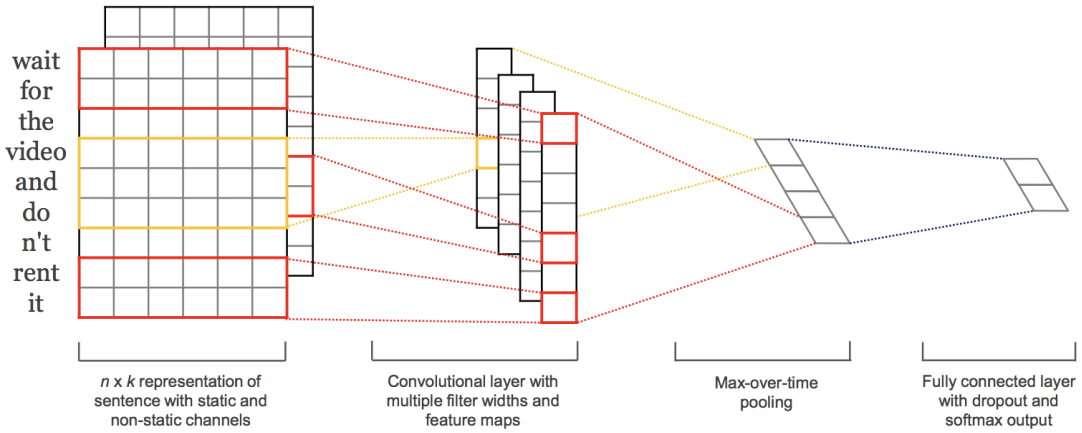
Models and/or Method(4 marks):

To tackle our problem, we decided to use Natural Language Processing (NLP). There were 3 different types of models that were trained and used to determine the type of emotions present in the lyrics of songs. We decided to utilise the TextCNN, Bert-based and BiLSTM models.

TextCNN

Definition

The first model was the TextCNN model which is a deep learning algorithm used for sentence/text classification tasks such as sentiment analysis. To perform text classification, labels are provided to a set of texts or words which will reveal the sentiment of the set of words. The concept of this model is to treat sentences as images. By having a sentence, a matrix of numbers can be created to represent the sentence where each row of the matrix corresponds to a one-word vector. The advantages of the CNN is that it is the most common model for every image related problem and with textCNN treating the text as an image, it is well-suited for our task and it is also computationally efficient.



Our model - How it works

The code was modified from notebook “Multiclass Text Classification - Pytorch” [1] so it could adhere to our specifications. This code can be found in “Alon_nn/prediction.py”. First libraries are import for the TensorFlow backend before basic parameters are set such as:

- Size of each word vector
- Number of unique words
- Number of samples to be process at once
- How many iteration over all samples

The data is imported and separated as a table. There are checks for null data which are promptly removed. The max length is then found before the y column is preprocessed. The data is then filtered by cleaning of numbers and contradictions. The data is then broken down further to enable the tokenization and padding of sentences. The class CNN_Text is made before it is trained. This is done in CUDA Memory where Torch datasets and Data loaders are created so the model can be trained in a certain configuration. The prediction, loss, and accuracy are stored and printed so the user can check those parameters. The last function predict_single is used to clean the dataset which is then inputted to the pre-trained model so it can output the emotion classification of the song.

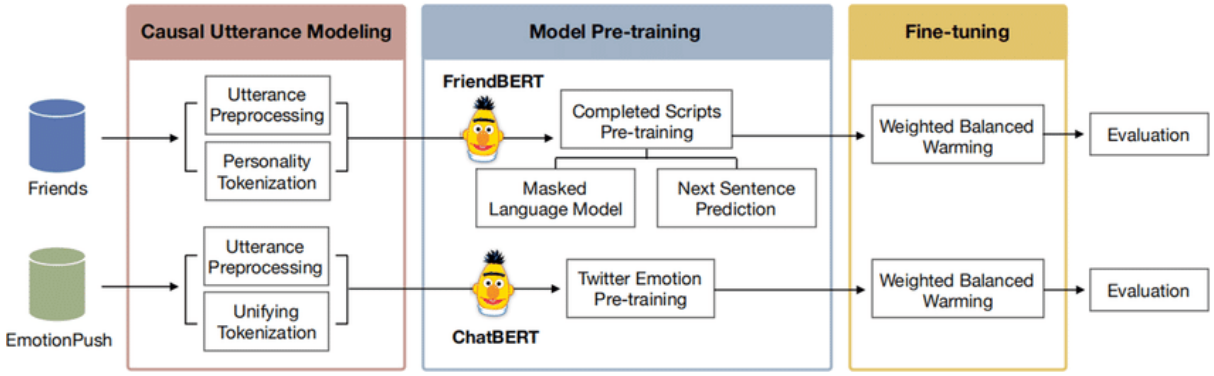
BERT-based

Definition

The next model is a pre-trained BERT-based model used in Go-Emotion which is fine-tuned. BERT which stands for Bidirectional Encoder Representations from Transforms is a transformer-based machine learning technique developed for NLP. BERT is first pre-trained through a large repository of specialised, labelled training data. Due to transformers providing an increased capacity for understanding context and ambiguity in language, any given word can be processed in relation to other words in a sentence, rather than one at a time which allows the BERT model to understand the full context of the word. This model is in contrast with the traditional method of language processing known as word embedding which maps every single word to a vector and is utilised in the previous model TextCNN. Word embedding models are adept at many general NLP task but fail at context-heavy tasks due to fixing words to a meaning independent of context. Therefore, since BERT identifies the words by their

[1] https://www.kaggle.com/code/mlwhiz/multiclass-text-classification-pytorch/notebook#Pytorch-Model---TextCNN

surrounding rather than a pre-fixed identity, this makes it the best suited model for identifying the emotions present in the lyrics of songs as they require context to comprehend.



Our model - How it works

The source of this code is from GoEmotions-pytorch github [2]. The code, which can be found in Jupyter “notebooks/go-emotion.ipynb " was modified to meet our project’s requirements such as reducing 28 emotions to 6 types of emotions by grouping them. Firstly, the Go-Emotion relevant libraries should be imported so then the bert based neural network model can be built. The specific model is bert-based-cased. Next all configurations such as the dataset and parameters should be set. We’ll be using the Ekman Taxonomy. To run the machine learning model, we initialise the model using the already existing pre-trained model so we can load the data set to train the model. An evaluation of the results of the training will be outputted.

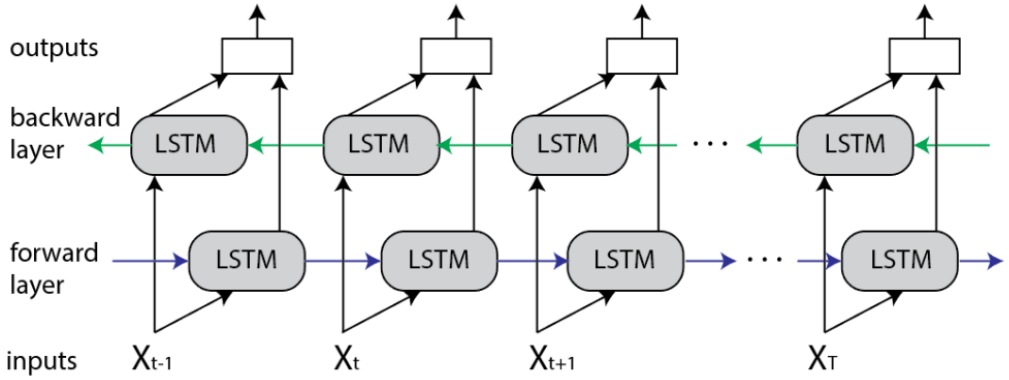
BiLSTM

Definition

The final model is a Bidirectional Long short-term memory which is a sequence processing model that consists of two LSTMs with input flowing in both directions. This increases the amount of information available to the network by improving the context available to the algorithm, for example knowing what words follow and precede a word in a sentence. The words are detected first from their dictionary meanings before other immediate words are analysed to check the impact of the meaning of the next word therefore, the final meaning is derived

[2] https://github.com/monologg/GoEmotions-pytorch

from textual data and understanding the context. This means the model can detect emotions in sentences with relation to its context thereby well-suited to tackle the problem.



Our model - How it works

The main files are “LSTM_Keras/emotions.py” where the codes are inspired by the article “Emotion Recognition using Text” [3] while “LSTM_Keras/emotion_detector.py” is an entire new file created to separate the training and input section so the music inputs can be fed to a function that calls the pre-trained model. This text model will be using BiLSTM (Birdirection Long Short Term Memory). FIrst, the necessary packages are imported so then the datasets can be extracted and inserted into training. testing , and validation data frames. After the sentences are inserted, they are cleaned by eliminating all the prepositions, articles, punctuation marks, stop words to leave only the important words since those extra words act as noise. The sentences are then tokenized thereby extracting each unique word and creating a dictionary where each unique word is assigned an index. The parameters are then set so we can pad the training and testing sequences while the emotions are mapped according to their categorical value. To create this model, a 1 million trained word vector is used to help train the model in a more efficient and thorough manner which will lead to higher training accuracy. Next, the architecture to train the model needs to be created. An embedding layer is first created using the weights obtained from the word vectors file where we then add a bidirectional and dense layer. The model is now finally trained using a batch size of 128 and epoch size of 8. Now, the model is trained and can be fed the music inputs which will output the emotions in return.

---

3 https://medium.com/analytics-vidhya/song-recommendation-based-on-textual-and-facial-emotion-recognition-a95e6259c5d8

Results:

Quantitative Analysis:

When evaluating NN models qualitatively we have to look at a couple of factors and values which can determine a models effectiveness and accuracy for a particular data set. Looking into accuracy we see that we generally can use 2 values to determine this; Loss and Average. Looking at a model's average will tell us how much of the evaluation data a model has gotten correct or within an acceptable range, calculated as total correct guesses / total guesses, and loss will tell us how far away from the correct values our NN model has guessed results to. To evaluate a model's effectiveness we use F1-scores. An F1-score is the harmonic mean (weighted average) of the models predict and recall components which it uses to determine how well the model works for a particular data set. With F1-scores the higher the F1-score the more effective the model is at guessing the correct values and thus the more effective it is.

After training and testing, the accuracy, loss and F1-scores were determined through an evaluation data set, respective to each of the models:

CNN Model		Go-Emotion		LSTM-Keras	
Average:	0.6547	Average:	0.5678	Average:	0.5262
Loss:	0.8219	Loss:	0.7012	Loss:	0.7259
Macro F1-Score:	0.5262	Macro F1-Score:	0.5849	Macro F1-Score:	0.5262
Micro F1-Score:	0.6547	Micro F1-Score:	0.6713	Micro F1-Score:	0.6547
Weighted F1-Score:	0.6430	Weighted F1-Score:	0.6666	Weighted F1-Score:	0.6430

Looking at average and loss for each of the models, we see although CNN has the highest average it also has the highest loss which would introduce additional errors in our measurements (evidenced in the confusion matrix in appendix). Comparatively the LSTM-Keras model has the lowest accuracy and a moderate loss which is in between that of Go-Emotion and the CNN model. Due to the LSTM-Keras models performance in this aspect we chose not to use this model in the qualitative analysis. Looking at the Go-Emotion model, we see that we have a moderately high average in comparison to the CNN and LSTM model, and the lowest loss calculated.

Additionally looking at the F1 scores we see that the Go-Emotion model has the highest macro, micro and weighted F1-scores with the CNN model coming in second and LSTM-Keras last in this aspect.

We can see that the CNN model works better based on the average score with high loss and overall lower F1-score than go-emotion, we choose the go-emotion model for lyric analysis in the end.

Qualitative Analysis:

When Evaluating whether the NLP analysis was successful with allocating emotional songs, we have to judge whether the emotion of the lyrics of songs for ourselves as humans. With Music being able to express emotion and the ability conveying understanding of that Emotion being a very human expression, evaluating the emotion within songs and by extension the NLP allocation of emotion to song lyrics should be done by humans. Since “Music produces a kind of pleasure which human nature cannot do without ” (Confucius, The book of Rites) we see that if we only compare the values output form this ML model to that of other models, we may only understand what a machine thinks of a song, not the actual human emotion behind it. With this being said, our ML model was successful in categorising 25 different songs into 6 differing emotional categories these being:

1. Anger
2. Fear
3. Sadness
4. Surprise
5. Disgust
6. Joy

Within these 6 emotions our AI categories our 25 different songs into these categories:

1. Anger
    a. 1-800-273-8255 By Logic
    b. 34+35 By Ariana Grande
    c. Come Down By Anderson Paak
    d. Feelings By Lauv
    e. Monster By Kanye West
    f. Rap God By Eminem
    g. Rasputin By Boney M.
    h. The Way I Am By Eminem
2. Fear
    a. 2 Soon By Keshi
    b. Run Boy Run By Woodkid

    c. Without Me By Halsey
3. Sadness
    a. 22 (Taylor's Version) By Taylor Swift
    b. Bury A Friend By Billie Eilish
    c. Drivers Licence By Olivia Rodrigo
    d. Drunk By Keshi
    e. Glimpse of Us By Joji
    f. Never Gonna Give you Up By Rick Astley
    g. Thank u, next By Ariana Grande
4. Surprise
    a. Diggy Diggy Hole
    b. I Ain't Worried By OneRepublic
5. Disgust
    a. My Favourite Part By Mac Miller, Ariana Grande
    b. Unholy By Sam Smith
6. Joy
    a. Golden Hour By JVKE
    b. Happy By Pharrell Williams
    c. Hotel California By Eagles

In order to evaluate these song categorisations against people, we created a survey in order to display what emotion people thought for a particular song with our standard emotion list above. The results for the survey can be seen in the appendix.

The accuracy of the model in comparison to emotion is:

1. Anger: 2/8
2. Fear: 1/2
3. Sadness: 3/6
4. Surprise: 0/2
5. Disgust 1/2
6. Joy: 3/3

Leaving us with a combined accuracy of 40% for evaluating emotions for songs when comparing our NN model with the results from the emotion song survey.

Discussion:

There are multiple factors that would explain the accuracy of our results. Firstly, it is worth mentioning the subjectivity of music taste. While our score of 40% is based on which response from the survey had the most selections, 24 out of 25 songs had multiple emotions selected, highlighting this subjectivity. Additionally, another factor to consider is that our models only look at the lyrics of a song, and not the entire composition. It is likely that if we added in other inputs, including beats per minutes, the scale the song was written in or song reviews that we may have been able to get a higher level of accuracy.

Generally speaking though we are quite happy with the results of our models. Firstly, all the models were able to run in a timely manner while still providing better than naive (⅙ correct) outcomes, showing that our model was likely able to make some insights on the mood of song lyrics. Additionally, there was a correlation between epochs and accuracy / loss, in which the longer the algorithms trained for, the more accurate they became.

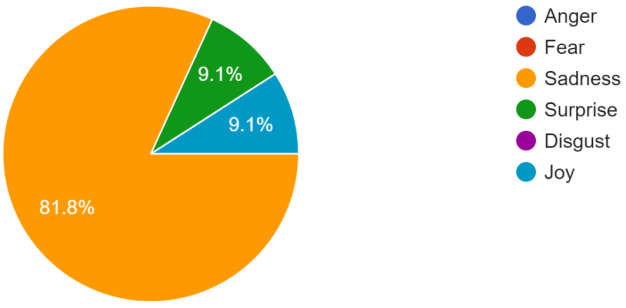
However, there were some potential weaknesses. Firstly, the large loss in a practical sense might limit the value of our algorithm, as even though it would have the majority of songs in the playlist being categorised correctly, it might be jarring to have an angry song in your joyful playlist. Secondly, there is the potential that following Eckman's theory of emotions limited our algorithm too, as we were focused only on 6 emotions (minusing neutral), thereby increasing our loss. Finally, as visible through the distribution of the inputs and classes from the GoEmotion dataset, finding a more balanced set might have been beneficial to producing a more accurate model.

In terms of possible future work, the first step would be to integrate larger amounts of data as inputs into the models in order to get a more comprehensive understanding of the music. This would ensure that every aspect of a song is being taken into account before making a prediction of the mood. Additionally, it would be best if we segmented the emotions beyond Eckman's in order to minimise loss and maximise utility of the output. Finally, sourcing another dataset that maintains the colloquialism and is labelled based on mood with more balanced counts of the labels would also be useful in making our models more accurate.

Appendix:

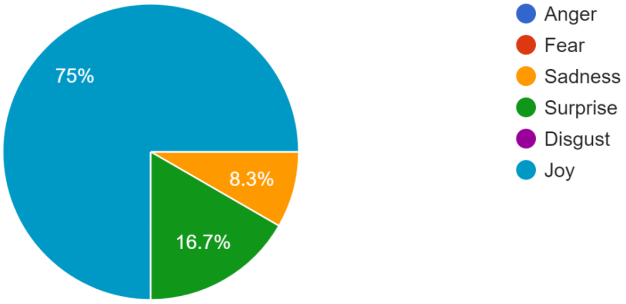
1-800-273-8255 By Logic, Alessia Cara, Khalid

11 responses



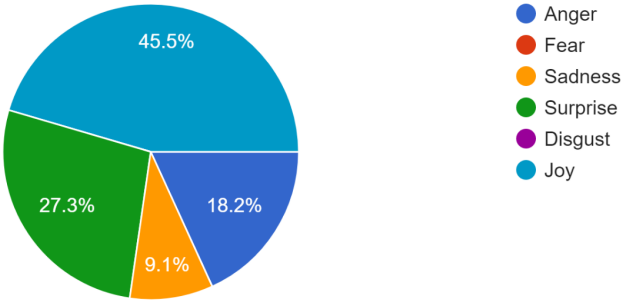
34+35 By Ariana Grande

12 responses



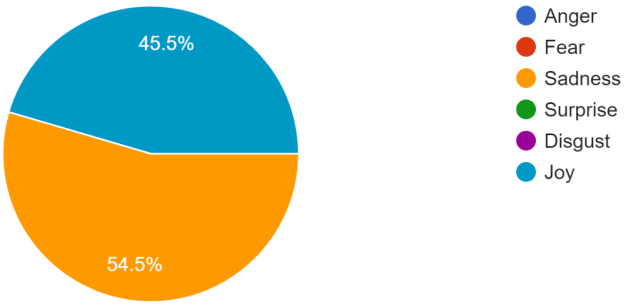
Come Down by Anderson Paak

11 responses



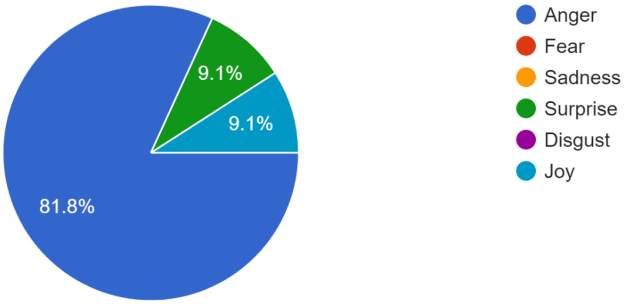
Feelings by Lauv

11 responses



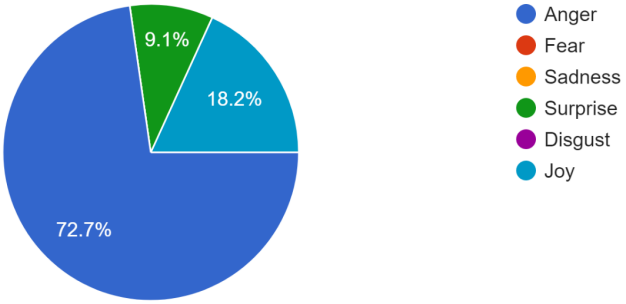
Monster By Kanye West, Jay-z, Rick Ross, Nicki Manaj, Bon iver

11 responses



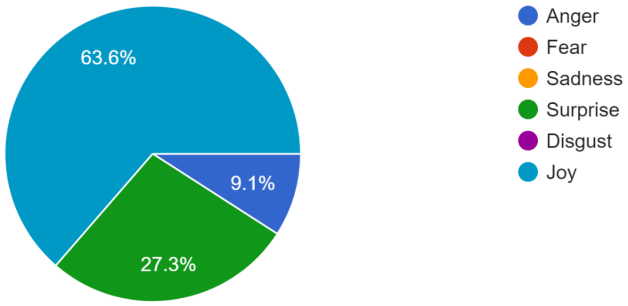
Rap God by Eminem

11 responses



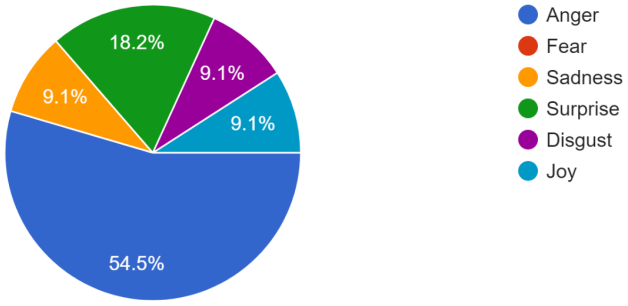
Rasputin By Boney M.

11 responses



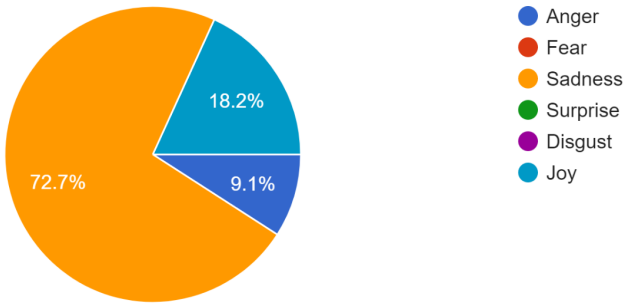
The Way I am by Eminem

11 responses



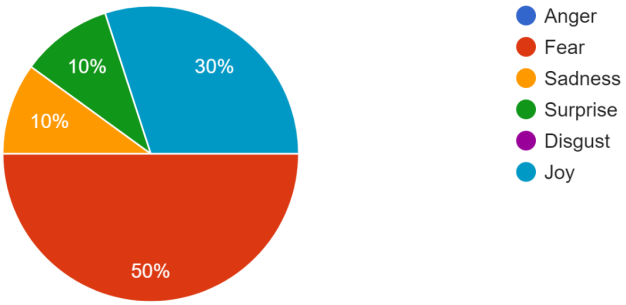
2 Soon by Keshi

11 responses



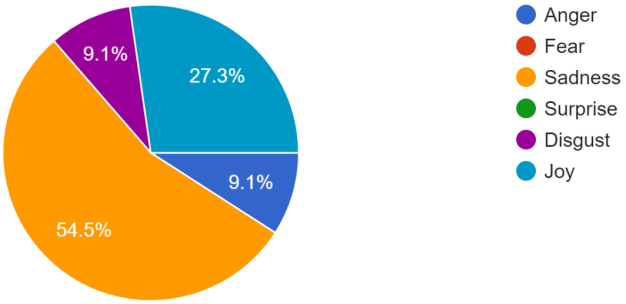
Run Boy Run By Woodkid

10 responses



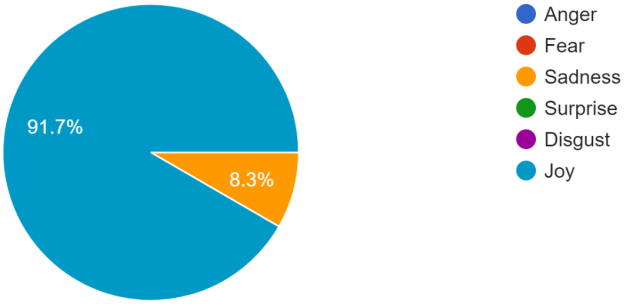
Without Me By Halsey

11 responses



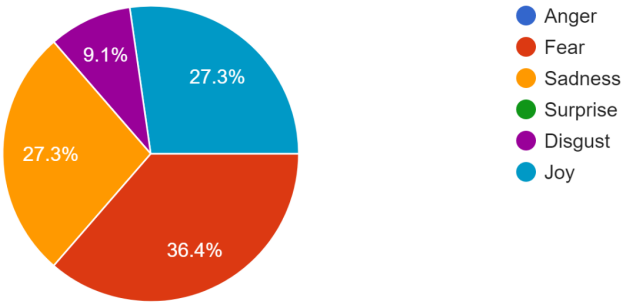
22 (Taylor's Version) By Taylor Swift

12 responses



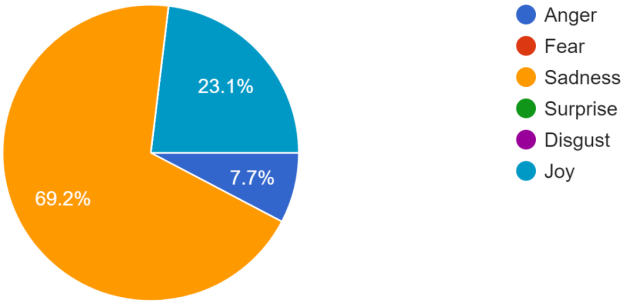
bury a friend By Billie Eilish

11 responses



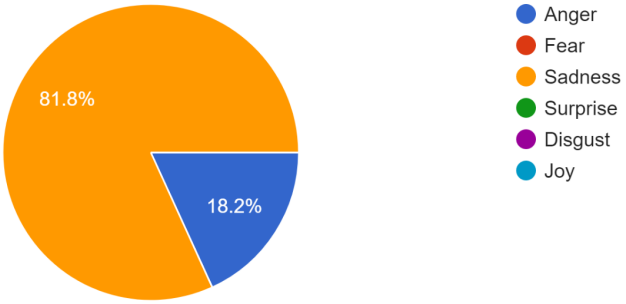
Drivers License By Olivia Rodrigo

13 responses



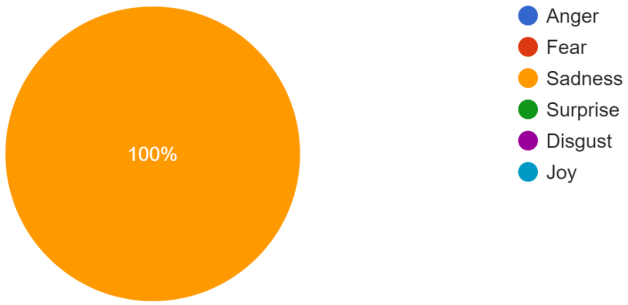
Drunk by Keshi

11 responses



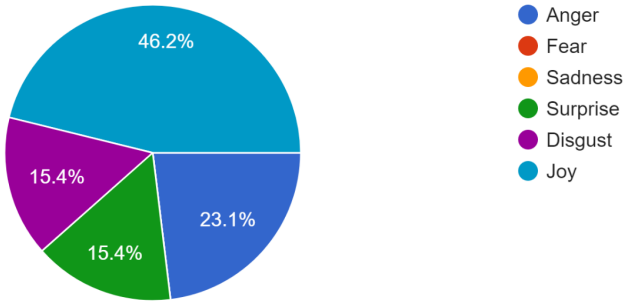
Glimpse of Us By Joji

13 responses



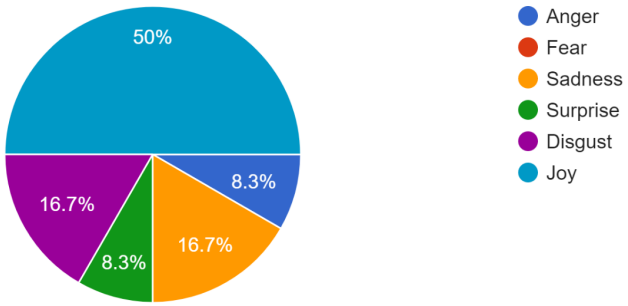
Never Gonna Give You Up By Rick Astley

13 responses



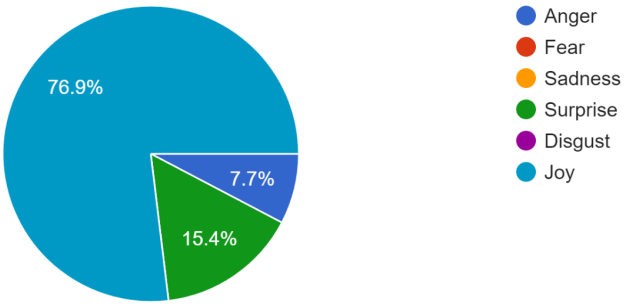
Thank u, next By Ariana Grande

12 responses



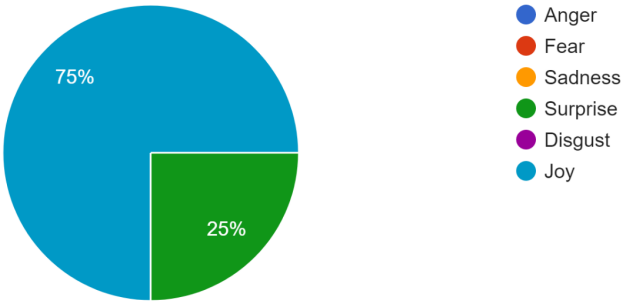
Diggy Diggy Hole By The Yogscast

13 responses



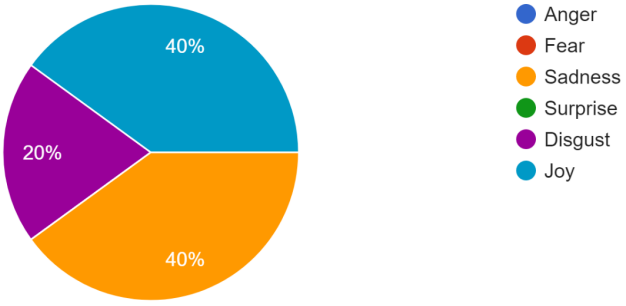
I Ain't Worried By OneRepublic

12 responses



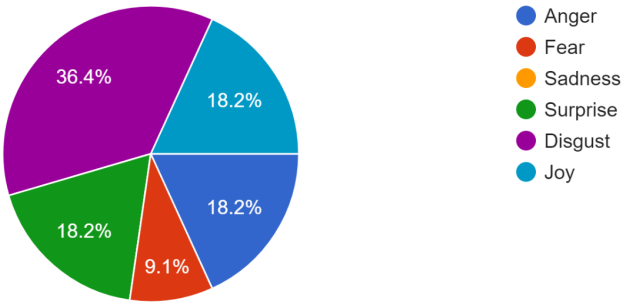
My Favorite Part By Mac Miller, Ariana Grande

10 responses



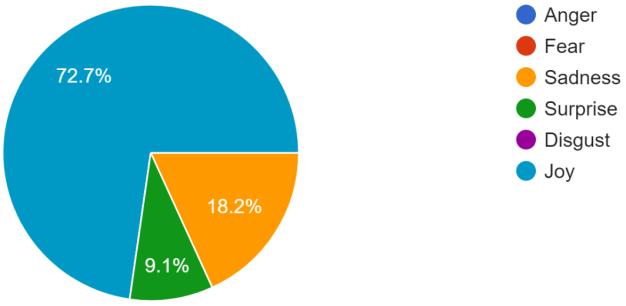
Unholy (feat. Kim Petras) By Sam Smith, Kim Petras

11 responses



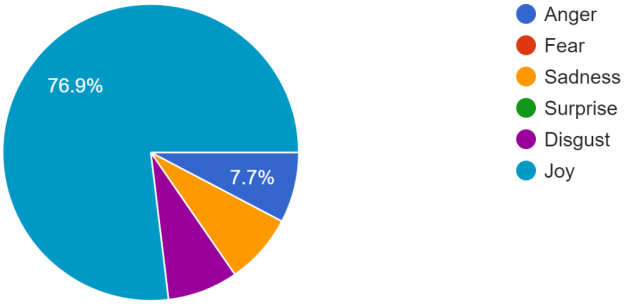
golden hour By JVKE

11 responses



Happy By Pharrel Williams

13 responses



Hotel California By Eagles

11 responses