MUSIC EMOTION RECOGNITION WITH DEEP LEARNING ON AWS SAGEMAKER

DOMAIN BACKGROUND

Music Information Retrieval (MIR) is an interdisciplinary science of retrieving information from music, whose aim is to develop computational tools for processing, searching, organizing, and accessing music-related data.

The field of Music Emotion Recognition (MER) has been evaluated consistently since 2007 in the Music Information Retrieval Evaluation Exchange (MIREX) Audio Mood Classification task. This task consists of the classification of audio into five mood categories or clusters, containing different emotional tags. Emotions in this context refer to what listeners perceive since; 1. It's relatively less influenced by situational factors of listening e.g environment, and 2. Listeners are generally consistent in their ratings of the emotional expression of music (1909.05882).

In terms of the definition of emotions, the following attributes and characteristics of emotions were considered; 1. Emotions appear to vary their intensity e.g irritation – rage, 2. Emotions appear to involve distinct qualitative feelings e.g feeling of being afraid – nostalgic, 3. Some pairs of emotions appear more similar than others e.g joy and contentment versus joy and disgust, and 4. Some emotions appear to have opposites e.g happiness – sadness, love - hate, calm – worry. These attributes altogether led to the definition of the following two approaches for representing emotions; 1. A categorical approach where emotions are represented as categories distinct from each other, i.e happiness, sadness, anger, surprise, and fear, and 2. The Dimensional approach on the other hand is where we have emotions conceptualized based on their positions on a small number of dimensions i.e valence and arousal, which is what I'll be using in this project as it helps reduce the dimension of training a MER model.

Valence refers to the space from negative to positive mood, and arousal is the space from calm to energetic. The paper (1809.07276) further explains that arousal is highly correlated to the audio while valence is highly correlated to both audio and lyrics. The project decision, however, will be based on mood or emotion detection based on lyrics only as using audio will require higher data transfers and there may be copyright issues.

For this project, I'll be applying a concept in the paper (1909.05882) called 4Q, where emotional categories are further categorized into 4 Quadrant Dimensions of Arousal and Valence as shown in the table below;

Quadrant	**Emotions**	**Synonyms**
Q1 (A+V+)	Joyful activation	joy

	Power	-
	Surprise	-
Q2 (A+V-)	Anger	angry
	Fear	anguished
	Tension	tense
Q3 (A-V-)	Bitterness	bitter
	Sadness	sad
Q4 (A-V+)	Tenderness	gentle
	Peace	-
	Transcendence	spiritual

PROBLEM STATEMENT

Music recommendation has come a long way such that a single search of a song almost guarantees you a continuous stream of other songs that you might like, and almost always do. These recommender systems usually use techniques such as collaborative filtering, and audio models to recommend music. This means that the music that gets recommended always has similarities with music you already listen to or what group of users (in collaborative filtering) you belong to.

Emotions in humans usually have a tendency to be erratic, and recommender systems sometimes won't always be able to pick up on this since a user's current emotional state may force them to deviate from the usual user space or profile the system has built on them, meaning that a song being recommended may not be able to fit in well with a user's current emotion, thus using music emotion recognition seems like something that will always be able to tailor music based on emotions, and be used by existing recommender systems to determine which songs based on the emotion they should recommend.

For most music streaming companies, the above-mentioned methods of music recommendation are mainly for financial gain, so they tend to be very effective in terms of getting music in front of the users of their platform, thus there is some margin of error in terms of the experience they are willing to forgo regardless of how it affects a user's experience.

From my experience, their systems cannot usually create recommendations that necessarily have the emotional impact which as a listener, I may be in search of, especially when I need to slip into and out of certain moods.

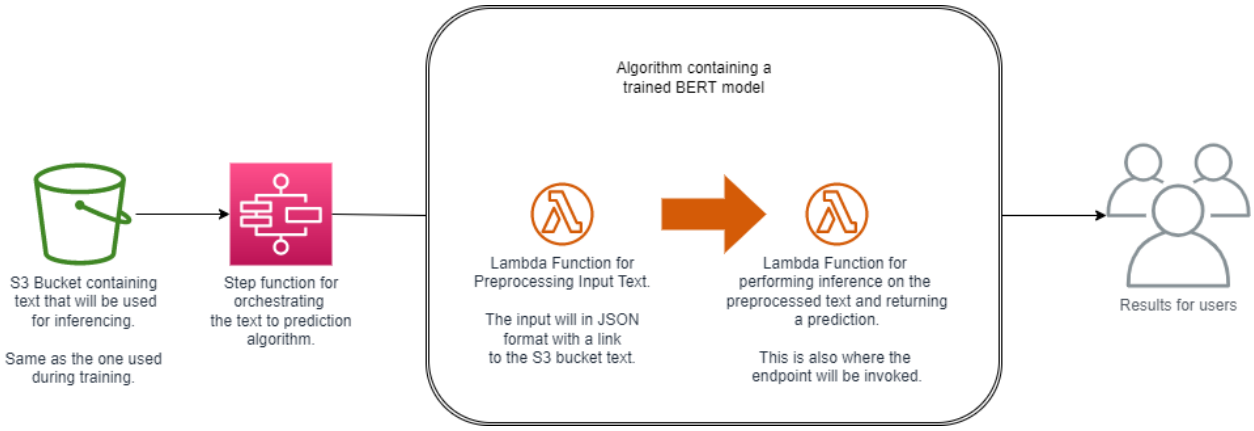
SOLUTION STATEMENT

In this project, I focused on the use of Deep Learning for the classification of text as part of Natural Language Processing (NLP), more specifically Natural Language Understanding.
The model to be used in the project for lyrics classification is BERT or Bidirectional Encoder Representation Transformer, more specifically a pretrained BertForSequenceClassification from the HuggingFace platform. It is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering, text classification and language inference, without substantial task-specific architecture modifications. (1810.04805).

A variation of BERT called *bert-base-uncased* pre-trained model from the transformers library was used in the project, where I assumed that for example, the word BAD and bad, both shared the same sentiment and since the *base* model only had 110 million parameters, I could first train it locally on my computer to save on AWS resources.

The platform that will be used is AWS, more specifically the following services;

- S3. This is where all data related to the project will be stored. This may include the data sets. And even the model data.
- SageMaker Studio. This is where all data preprocessing such as removal of punctuations and contractions, data tokenization, hyperparameter tuning, model training, evaluation and testing, and endpoint creation will be done.
- Lambda and Step Fucntions. This will be used to deploy the final model endpoint. The algorithm to be followed for making inference will be as shown in the following figure;



DATASETS AND INPUTS

The dataset selected for this project was the Deezer Music Emotion Recognition (MER) Dataset which was mentioned in (1809.07276) and was publicly available in the GitHub repository. The dataset came already split into training, validation, and testing sets, with data points 11000, 3000, and 3000 respectively. The dataset contained the following columns; *dzr_sng_id, MSD_sng_id, MSD_track_id, valence, arousal, artist_name, track_name*.

For this project, I needed to add a lyrics column to the dataset. The API selected for this task was the MusixMatch API, more specifically the lyrics endpoint. More details about the API can be found by following the following link to the API's documentation.
Since I could not hit the endpoint and directly add lyrics to our dataset, I build a *lyrics_service.py* program that could use query the endpoint using the *track_name* and return the collected lyrics. These lyrics were then added to a new data feature/column named *lyrics* using the Numpy library.

I was also using the basic plan version of the MusixMatch API, I could only obtain 30% of a single song's lyrics, which may be sufficient for the project.

BENCHMARK MODEL

The benchmark projects for this project were more related to emotion recognition using audio rather than text, although they still gave a picture as to the kind of performance I should be expecting. Most of the information used for this section was gotten from this site, paperswithcode.com. The paper that contained information related to emotion recognition in music was this one, (2107.05677v1), which demonstrated the usage of a technique called CALM with a performance of about 72.1% on arousal, and 61.7% on valence, both described in the first section. This averages to about 69.9%.

There were no benchmark models publicly available for using lyrics only for this particular project, although, since the dataset and the paper (1809.07276) were both from researchers at Deezer, I can safely assume that their platform uses some aspect of MER for the recommendation of music, and the paper utilizing CALM gave a somewhat related accuracy that I should expect.

EVALUATION METRICS

As for model evaluation, precision and recall were calculated. Precision refers to the number of correct labels among the labels predicted, while recall refers to the number of labels that were successfully predicted, among all the real labels. Accuracy was calculated although not used as the main measure of model performance since there is an existence of class imbalance in the dataset.

An example classification report is as below;

	precision	recall	f1-score	support
0	0.49	0.53	0.51	1048
1	0.38	0.33	0.35	663
2	0.38	0.43	0.40	968
3	0.28	0.22	0.25	594
accuracy			0.41	3273
macro avg	0.38	0.38	0.38	3273
weighted avg	0.40	0.41	0.40	3273

PROJECT DESIGN

The workflow for this project will be;

i. Design or selection of hyperparameter.
ii. Hyperparameter tuning. Determining what hyperparameters work best for the task.
iii. Actual model training with best hyperparameters.
iv. Model evaluation based on the above metrics.
v. Deployment as an endpoint.

vi. Querying the deployed endpoint via Lambda.