APS360 PROJECT PROGRESS REPORT

Jonas Martins
Student# 1006869907
jonas.martins@mail.utoronto.ca

Chielotam Agbatekwe
Student# 1006988057
chielotam.agbatekwe@mail.utoronto.ca

Jennifer Sunny
Student# 1006998732
jennifer.sunny@mail.utoronto.ca

ABSTRACT

This abstract summarizes a progress report on a project that employs music sentiment analysis to categorize music into moods to help users create emotionally curated playlists. The project leverages deep learning to extract audio features and label music by valence and arousal values. The team adapted their responsibilities after a member left and shifted their project management to a Discord server. They have outlined their progress in data collection, cleaning, and preliminary model building. The data was carefully chosen from royalty-free sources and cleaned using the Librosa library to avoid copyright infringement. Tasks moving forward include model development, data augmentation, and final report and presentation preparations. The project demonstrates a commitment to ethical considerations, balanced data sets for unbiased models, and thorough documentation of the process.

—-Total Pages: 8

1 PROJECT DESCRIPTION

Music has a significant influence on human emotions and behaviours. Exposure to certain melodies, rhythms, and harmonies amongst other things can help alleviate bad moods and increase productivity. Music can also serve as a means of expression and a way to find community and understanding. The goal of this project is to employ music sentiment analysis to categorize popular music into moods. The practical application of this is to grant a user the ability to curate playlists based on their desired emotional state in hopes of enriching their experience. Deep learning plays a vital role in automating the extraction of crucial audio features like MFCCs, Spectral Contrast, and Chroma. Simultaneously, it labels the music with appropriate valence and arousal values using neural networks and training algorithms (See Sarkar et al. (2020). for more information). A Support Vector Regression (SVR) was employed as a baseline model to serve as a benchmark for the Convolutional Neural Network (CNN) outputs. The evaluation was based on the mean squared error and the R-squared (R2) error, with the aim of ascertaining whether the CNN model outperforms the SVR in terms of the accuracy of its predictions compared to the target values. In the final analysis, the emotional state is established by plotting the obtained valence and arousal values within a 2D space, leading to the classification of the output into the pertinent moods which for the purpose of this project have been defined as 'Happy and Energetic', 'Sad and Soothing', 'Intense and Aggressive', 'Calm and Joyful' and 'Neutral'.

2 INDIVIDUAL CONTRIBUTIONS AND RESPONSIBILITIES

After the project proposal submission, our team member Daniel decided to drop the course. As a result, each group member had to adapt to take on more responsibility for the tasks he was assigned.

The team also had to shift the project folder and Google Colab link, as seen in section 4. This is because Daniel was the owner of these files. The team stuck to the plan written in the proposal, having a kick-off meeting to review assigned tasks, compare each other's academic schedules and shift assigned work depending on our availability. To project manage, we communicate and meet on our discord server. We have a dedicated channel for coding questions and send a chat in the channel when the code has been updated and completed. We make sure to communicate with each other when code is being updated and make copies on our own computers every time code has been updated. The progress report was broken down into coding and written tasks, with each team member assigned to bits of the smaller task. A summary of tasks completed with the team member who led it can be seen in Table 1 with internal due dates.

Table 1: Project Progress Report Task Completion

LEAD	TASK COMPLETED	DUE
All Team	Availability, project plan and task assignment meeting	Oct 20
Jennifer and Chielotam	Obtaining data (ensure data is copyright free)	Oct 27
Jennifer and Chielotam	Collection dataset of at least 1000 samples, including wav files of same duration and clear mood classifications	Oct 27
Jonas	Data cleaning of wav samples into MFCCs, spectral contrast, chroma, tempograph	Oct 28
All Team	Meeting to plan baseline model and primary model architecture	Oct 28
Jennifer and Chielotam	Coding and producing 1 Qualitative and Quantitative result of baseline model	Oct 30
Jonas	Feasibility Considerations throughout entire modeling process	Oct 31
Jonas	Coding and producing 1 Qualitative and Quantitative result of primary model	Nov 1
Chielotam	Summary, Explanation and Results	Nov 1
Jennifer	Writing Task: Individual Contributions Summary, Noteable Contributions - Data Processing, Primary Model	Nov 1
Jonas	Writing Task: Noteable Contributions - Primary Model and editing of the document	Nov 2
All Teams	Project finalization and overview	Nov 2
Chielotam	citations	Nov 3

Since one of our team members dropped the course, we must update our planned tasks going forward. Based on our schedules, coding experience and video editing experience, we have delegated the presentation and final report according to Table 2 and Table 3

Table 2: Project Final Report Task Assignment

LEAD	ASSIGNED TASK	DUE
All team	Team meeting regarding availability, assessment of delegated tasks and next steps	Nov 9
All Team	Further development of the model, collect new data	Nov 17
Chielotam	Writing tasks: Introduction, Illustration, Baseline Model, Project Diffuculty	Nov 24
Last Edit and structure, grammar, citations	Nov 24	
Jennifer	Background and Related Work, Data Processing, Ethical Considerations, Discussion	Nov 24
Jonas	Qualitative and Quantitative Results, Architecture, Evaluation of model	Nov 30
Chielotam	Last edit, citations, quality of flow	Nov 30

Table 3: Project Final Report Task Assignment

LEAD	ASSIGNED TASK	DUE
Chielotam	Script and video production of Problem Introduction, Data Processing, Quantitative Results	Nov 26
Jennifer	Script and video production of Data Collection, Model	Nov 26
Jonas	Script and video production of Demonstration, Qualitative and Quantitative Results	Nov 26
Chielotam	Key Takeaways from project and conclusion	Nov 24
Jennifer and Jonas	Video Editing	Nov 30

3 Notable Contribution

3.1 Data Processing

To prevent potential violations of copyright infringement, our group made an effort to collect samples from royalty-free sources. Using the DEAM dataset, compiled by researchers from the University of Geneva. These samples came from the archives of "freemusic.org", "jamendo.com" and the Medley DB dataset. Our team was able to collect 2000 samples of royalty-free music, collected in 45-second excerpts randomly throughout the song. The data set contained music clips in MP3 files as well as dynamic annotations for every 500ms of the song. The annotations were also collected for the entire clip duration. The annotations came in the form of valence and arousal scores of each song, ranked on a scale of 1 to 9 (See DEAM (2018). for more information).

To clean our data we had three crucial steps. First, it was to ensure there was a variety of moods in our data set. This is important because if there are more samples in one category, our model will be highly trained in identifying songs in the specific category while doing poorly in other classes. This creates a biased model. As a result, our team identified 5 different moods based on valence and arousal to categorize our samples. We then selected 200 samples from each of the categories, totalling 1000 total. The category thresholds can be seen in Table 4.

These thresholds are based on the Circumplex Model of Emotions by James A. Russel which models emotions based on the valence and arousal ratings of a situation or event. Valence is defined as the positivity or negativity of emotion while arousal is defined as the intensity of the state. They are both

Table 4: Music Mood Categories based on valence and arousal

Valence Score	Arousal Score	Mood
Greater than 6	Greater than 6	Happy and Energetic
Between 3 and 6	Between 3 and 6	Neutral
Less than 3	Less than 3	Depressed
Greater than 6	Less than 3	Calm
Less than 3	Greater than 6	Aggressive

measured on a one to nine-point scale. Depending on the rating of both variables, different moods can occur. The resulting emotions from this ranking can be best illustrated in the figure below (See Na Du a (2020).

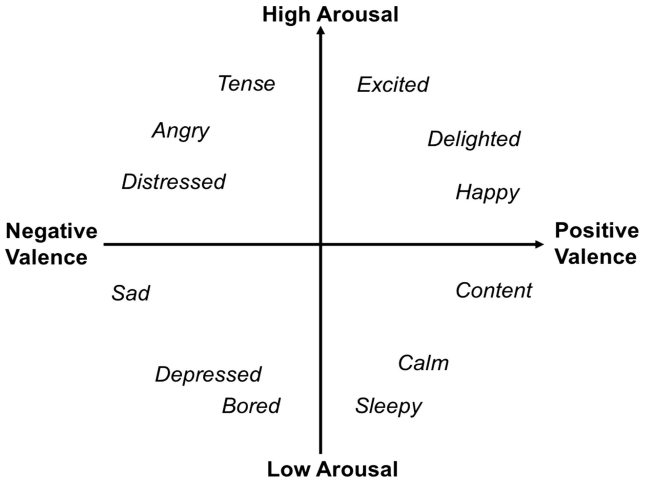

Figure 1: Russel's Circumplex Model of Emotions

After selecting samples based on class, we cleaned the samples using the Librosa Python library. The library loads audio files from formats such as MP3 into a Numpy Array and extracts features. The features we chose to use in extraction were MFCCs, Spectral Contrast, Chroma and Tempogram. MFCCs stands for Mel-Frequency Cepstral Coefficients which capture different frequencies in the music based on human auditory perceptions.

To extract the MFCCs, we used the librosa.feature.mfcc function which calculates the MFCC over a timeframe based on the inputted audio array, sample time, MFCC coefficients, number of samples in each short-time fourier transform frame and hop length. We set the sample time to 44100 seconds, with 13 MFCC coefficients, 2048 short-time fourier transform frames and a hop length of 512.

Spectral contrast extracts the difference between peaks and valleys in a sound spectrum, aiding in the differentiation between timbral tones. Spectral contrast was extracted using the spectral contrast function in Librosa's features based on the inputted audio array, short-time fourier transform frame of 2048, hop length of 512, minimum frequency of 200 Hz, and 6 bands. Bands was set to six as this is the typical amount to capture music sound spectrum.

Chroma provides a snapshot of the energy distribution across twelve pitch classes and represents harmony, chord progression and tonal structures. Chroma can also be extracted using Librosa's feature Chroma function. The function outputs chroma based on the inputted audio array, sample time of 44100, short-time fourier transform frame of 2048, hop length of 512 and chroma of 12, as there are 12 pitches in Western music (See Han et al. (2022)).

A cleaned data sample's features can be seen in the figures below.

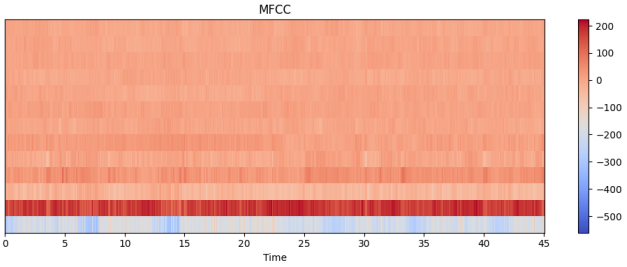

Figure 2: MFCC of a cleaned sample

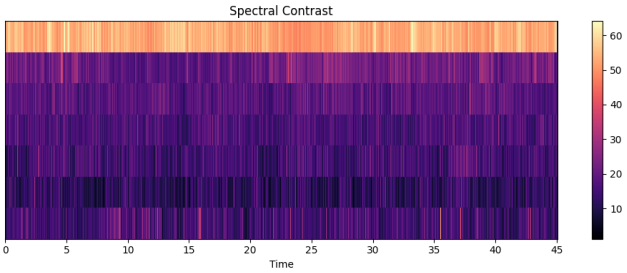

Figure 3: Spectral Contrast of a cleaned sample

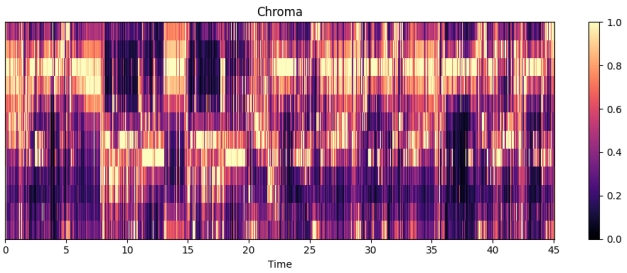

Figure 4: Chroma of a cleaned sample

We also had to clean our data by adding padding, as the different features of the music were not the same size. This will be necessary if we plan to start with an ANN neural network model.

With the data cleaned, we can put it into our baseline and primary model. With consistent data samples with even distribution among classes and accurately extracted features, our model can learn to classify music with less bias.

For testing, we plan to obtain new test data from a royalty-free music source. Similar to our samples, we will crop our MP3 files to 45 seconds. The MP3 files will be cleaned by extracting its features through the Librosa library. To ensure we have samples from all five classes, we will create their annotations based on the tags of the music on royalty-free music sources. Typically, these sources have mood genres or tags for happy, sad, calm, aggressive and neutral music. We plan to slowly collect this test set throughout our project duration, with at least 10 samples from each class.

3.2 Baseline Model

Prior to developing a complex CNN model to classify the music data samples into established moods, we opted to establish a baseline model. The simplicity of the baseline model allowed us to set a performance benchmark for the more complex model and ensure that the primary model can outperform the SVR and yield outputs with less error. It also helped resolve fundamental issues

with loading the data and proved that the problem itself was solvable. The baseline model chosen for this project is the Support Vector Regression (SVR). SVR predicts continuous numerical values by finding a hyperplane that best represents the relationship between the input features and the target outputs. For simplicity, we only used the first three inputs from the extracted MFCCs to obtain the valence and arousal scores.

The table below provides a comprehensive overview of the quantitative performance of Support Vector Regression (SVR) on a range of parameter settings and dataset sizes. The results are presented as various error metrics, highlighting the effectiveness of each configuration. Each row corresponds to a specific combination of SVR parameters, such as the choice of kernel function (helps map the input data into a higher-dimensional space and capture the non-linearity between the features and labels), the C value (defines the model's tolerance for errors and ability to handle outliers), degree (if applicable), and dataset size. The columns display error metrics, including Mean Squared Error (MSE), and R-squared score (R^2), allowing for a thorough evaluation of SVR's predictive accuracy under different conditions. This table serves as a valuable reference for selecting the most suitable SVR configuration based on the specific dataset and problem at hand, ultimately aiding in the optimization of regression model performance.

Table 5: SVR Results for Linear and RBF Kernels

TEST	SAMPLE SIZE	OUTPUT	KERNEL TYPE	C VALUE	DEGREE	MSE	R^2 SCORE
1	300	VALENCE	LINEAR	1.0	-	0.858	0.527
2	300	AROUSAL	LINEAR	1.0	-	1.601	0.099
3	300	VALENCE	RBF	1.0	-	0.746	0.589
4	300	AROUSAL	RBF	1.0	-	1.392	0.217
5	300	VALENCE	RBF	10.0	-	0.649	0.642
6	300	AROUSAL	RBF	1.0	-	1.392	0.217
7	300	VALENCE	RBF	100	-	0.776	0.572
8	300	AROUSAL	RBF	100	-	1.380	0.224
9	300	VALENCE	POLYNOMIAL	0.8	1.0	0.819	0.548
10	300	AROUSAL	POLYNOMIAL	1.0	3.0	1.428	0.197
11	300	VALENCE	POLYNOMIAL	10.0	2.0	0.737	0.593
12	300	AROUSAL	POLYNOMIAL	10.0	3.0	1.311	0.263
13	300	VALENCE	POLYNOMIAL	10.0	3.0	0.795	0.561
14	300	AROUSAL	POLYNOMIAL	10.0	4.0	1.376	0.226
15	300	VALENCE	POLYNOMIAL	100.0	3.0	0.870	0.520
16	300	AROUSAL	POLYNOMIAL	100.0	3.0	1.360	0.235
17	1000	VALENCE	RBF	10.0	-	0.986	0.453
18	1000	AROUSAL	POLYNOMIAL	10.0	3.0	1.741	0.002
19	500	VALENCE	RBF	10.0	-	0.847	0.492
20	500	AROUSAL	POLYNOMIAL	10.0	3.0	1.198	0.204
21	10	VALENCE	RBF	10.0	-	1.082	-47.111
22	10	AROUSAL	POLYNOMIAL	10.0	3.0	1.777	-13.510
23	100	VALENCE	RBF	10.0	-	1.018	0.247
24	100	AROUSAL	POLYNOMIAL	10.0	3.0	1.357	0.191

The results indicate that the best scenario for valence is test 5 (RBF, with C value = 10) while the best scenario for arousal is test 13 (POLYNOMIAL, with C value = 10, and Degree =3). This combination yielded the highest R^2 score.

A qualitative note on the performance of the model was that the error varied significantly based on the size of the dataset. A smaller dataset resulted in a higher error while a larger dataset resulted in a smaller error.

A challenge encountered while using the model was that the SVR does not take in empty datasets or datasets with a value of NaN because it cannot be scaled based on the set margins. To overcome this, the data was filtered first before being input into the SVR model.

3.3 Primary Model

The CNN is structured to analyze three key audio features: Mel-Frequency Cepstral Coefficients (MFCCs), Spectral Contrast, Chroma. These features are critical for capturing various aspects of audio signals such as timbre, harmony, and rhythm.

Model Architecture: The Convolutional Neural Network (CNN) devised for feature extraction from audio files is architectured with a multi-layer design. It comprises two convolutional layers and associated pooling layers, followed by a series of three fully connected layers. The first convolutional layer starts with 3 input channels to process the audio features (MFCCs, Spectral Contrast, and Chroma) and expands to 32 output channels, utilizing a kernel size of (13, 5) and a stride of (1, 2). A kernel height of 13 was carefully chosen in consideration of the nature of the features. Because all features have a height of 13 (after padding), which is the spectral axis, and the width represents the time axis, we decided it was more appropriate for the convolution to occur only along the time axis. We wanted to capture patterns of change in spectral values over time as this has a higher, and more cohesive semantic significance. The subsequent pooling layer reduces the dimensionality while retaining essential information. The second convolutional layer further refines the feature maps, increasing the depth to 64 channels with a kernel size of (1, 3) and a stride of (1, 1). Each convolutional layer is paired with a batch normalization layer to accelerate convergence and improve the generalization of the model.

Pooling layers with a window of (1, 2) follow each convolutional layer, further condensing the data into a more manageable form. The fully connected layers are dimensioned to distill the convolved features into predictions, with the first layer taking the flattened output from the preceding layers and condensing it into 256 neurons. The subsequent layers narrow the focus to 64 and finally to the 2 output neurons, corresponding to the desired predictions. The model incorporates dropout with a probability of 0.2 after the first fully connected layer to prevent overfitting.

Results: The model yields a training loss of 0.0163 and a validation loss of 0.0182. The Mean Absolute Error is 0.1019 for the training set and 0.1089 for the validation set. These metrics indicate that the model has a reasonable fit to the data without substantial overfitting. Furthermore, albeit the number of epochs is somewhat low (three), the preliminary results gathered are sufficient to find a strong performance on the validation data, at a lower computational cost. The figures below depict the loss and MAE over the epochs:

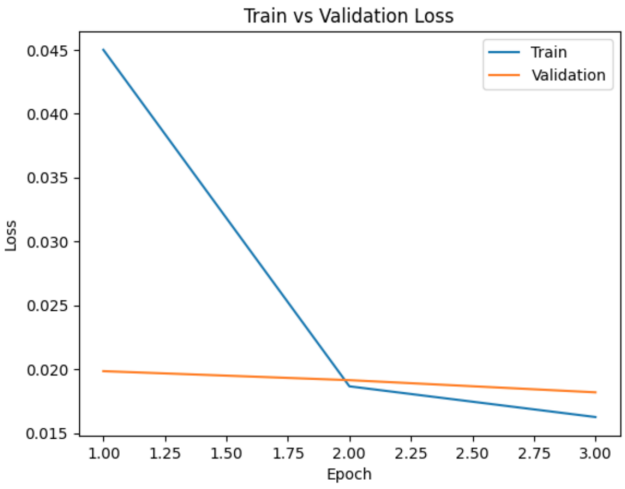

Figure 5: Training and Validation loss over three epochs

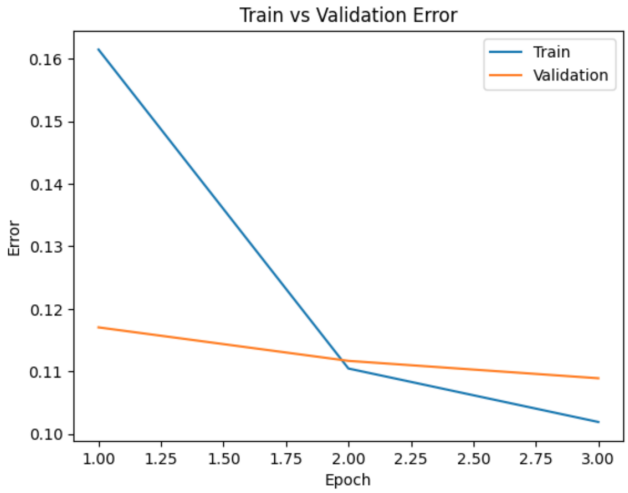

Figure 6: Training and Validation Mean Absolute Error over three epochs

Challenges: Throughout the development process, we encountered challenges in data preprocessing, particularly in normalizing the input feature sets. The model's sensitivity to the initial conditions and hyperparameter settings required careful tuning to ensure consistent convergence during training.

4 Google Collab Link

`https://colab.research.google.com/drive/16_IbshXTR5p3TA8T4NJTwFgIrxPF12x3?usp=sharing`

References

University of Geneva CVML Group DEAM. Deam: A database for emotion analysis using music, 2018. URL `https://cvml.unige.ch/databases/DEAM/manual.pdf`.

Donghong Han, Yanru Kong, Jiayi Han, and Guoren Wang. A survey of music emotion recognition. *Frontiers of Computer Science*, 16(6):16, 2022.

Elizabeth M. Pulver c Dawn M. Tilbury d Lionel P. Robert e Anuj K. Pradhan f X. Jessie Yang Na Du a, Feng Zhou b. Examining the effects of emotional valence and arousal on takeover performance in conditionally automated driving, 2020.

R. Sarkar, S. Choudhury, S. Dutta, A. Roy, and S. K. Saha. Recognition of emotion in music based on deep convolutional neural networks. *Multimedia Tools and Applications*, 79(1-2):765–783, 2020.