Music Recommendation System through Emotion Facial Reactions

Alexandra Przysucha AJP9010@NYU.EDU

Yuhong YZ9134@NYU.EDU

Andrea Cardiel ALC9588@NYU.EDU

Milestone 2

It should be known that though we had initially planned to create a recipe recommendation system, our group identified this would be an issue due to the limitations of our computing power and the large size of our dataset. When working with the dataset, all of us had issues with the disk space in HPC when uploading all the images of the food and recipes, even when it was in a zipped file and we only unzipped during the running of the code. In addition to that, we had other technical issues that were getting in the way of making any headway in our project and thus our group came to our project advisor and discussed our issues, and with the approval of Elaine (our project advisor) during office hours, we decided it was best to work on a project we had an interest in and that was also feasible with the data and the access to the datasets which were notable ones. We decided to make a music recommendation system that through CNNs using the FER dataset that has the facial reactions and labeled emotions associated to those facial emotion images that we can predict what sort of music, from Spotify's API and various music datasets, someone should be recommended given their facial emotion expression.

The goal of our work is to ultimately explore the intersection of music recommendations and emotional context. In particular, we hope to build a recommendation system that works to suggest songs on the basis of Spotify's pre-computed audio feautres (like danceability, energy, and mood) as well as user emotions taken from facial expression analysis. Traditionally, recommendation systems tend to use exclusively user preferences or collaborative filtering. Our system aims to integrate emotional data in an effort to make real-time, mood-based music recommendations that work to improve user experience and satisfaction.

To justify our new project, we found various research papers that work to improve music recommendation systems on the basis of sonic features including the following:

Baxter, Marissa, et al. "Context-Based Music Recommendation Algorithm." *arXiv*, 2021, arXiv:2112.10612, https://arxiv.org/abs/2112.10612.

- This research paper compares the performance of 6 machine learning algorithms (Logistic Regression, Naive Bayes, Sequential Minimal Optimization, Multilayer Perceptron, Nearest Neighbor, and Random Forest) across 3 platforms (Weka,

SKLearn, Orange) for context-based music recommendation using sonic features. It finds that Random Forest achieves the highest accuracy at 84%, and focusing on sonic attributes rather than popularity metrics improves recommendation performance compared to prior work. The neural network achieved similar accuracies but was less consistent across platforms.

Strengths:
1. Comparison of multiple algorithms and platforms.

2. Use of cross-validation to address overfitting.

3. Improved accuracy by focusing on sonic features.

4. Strong performance of Random Forest and Neural Network algorithms.

Limitations:
1. Small and homogeneous dataset, limiting generalizability.

2. Lack of in-depth analysis of algorithm performance differences.

3. Inconsistent performance of the Neural Network across platforms.

4. Unsuccessful optimization attempts for the Random Forest algorithm.

5. Limited discussion of implications and future directions.

Gong, Boning, et al. "Contextual Personalized Re-Ranking of Music Recommendations through Audio Features" ***arXiv*****, 2020, arXiv:2009.02782, https://arxiv.org/abs/2009.02782.**

- This research paper compares the performance of 3 recommendation algorithms (BPR, US-BPR, and CAMF_ICS) with two re-ranking approaches (global and personalized models) for context-aware music recommendation using audio features. It finds that the personalized model consistently outperforms both the global model and baseline algorithms, particularly when combined with BPR and US-BPR. The study demonstrates that audio features can effectively represent user preferences in different contextual conditions, with the personalized approach showing significant improvements in MAP@10 metrics.

Strengths:

1. Novel use of audio features for contextual re-ranking

2. Comprehensive evaluation using multiple baseline algorithms

3. Validation through opposite re-ranking experiments
4. Strong empirical evidence for personalized model superiority
5. Clear correlation demonstrated between audio features and contextual conditions

Limitations:

1. Evaluation limited to single contextual dimension (time of day)
2. Dataset constraints (#NowPlaying-RS only, after InCarMusic proved too sparse)
3. Lack of optimization guidance for the lambda parameter
4. No online user evaluation to validate offline results
5. Limited exploration of audio feature selection/weighting importance
6. Inconsistent performance of global model across algorithms

This work demonstrates that Spotify's pre-computed audio features have the capacity to model personalized music preferences. Consequently, we propose an initial step towards more sophisticated recommendation systems that combine these features with a CNN and attention mechanism. While these studies show that features including danceability, energy, and valence can lead to effective recommendations through re-ranking, adding an attention mechanism would help us to learn weightings of these features dynamically. Before we potentially incorporate new structures and data sources, this represents a crucial first step, allowing us to start with reliabe features that help isolate and validate the benefits of the attention mechanism itself, the architecture we develop should be extensible to additional audio features, user context, or even raw audio data in future iterations.

Our work, in particular, hopes to answer the question of how we might be able to integrate audio features as well as real-time emotion analysis in order to imrpove recommendations. It is particularly interesting because it has a wide applicability; as in fitness or relaxation apps which feature music designed to match or adjust a user's mood. Not to mention, it combines audio feature processing with emotion analysis, creating a dual-input recommendation system that has yet to be widely implemented in existing literature.

1. Methodology

The methodology for our project involves the use of 2 models; a CNN-based model for processing music and a CNN model for emotion recognition. Given the unique requirements of our project which hopes to merge these, we explored a variety of other tehcniques including RNNs and attention mechanisms in an attempt to capture sequential data in music and focus on the most important features in the data. Thus our methodology included the following steps:

1. Data preprocessing and exploration

The music dataset consisted of data from the Spotify API on various whose audio features we worked to standardize and encode audio for CNN projecting. Additionally, the music dataset includes the track and the artist and the audio features of the song, as well as the mood. The mood is determined based the various audio features such as tempo and valence, and danceability as well as all of the other audio features in the dataset. Additionally, there are other datasets that have the mood included with the track that we were able to utilise and incorporate into our music dataset, with songs from the 1960s till the 2000s.

With respect to the emotion dataset (FER) we worked to rescale and normalized facial expression images. The emotion model is trained on the FER (Facial Expression Recognition) dataset, which contains labeled images of facial expressions. The dataset includes common emotions such as "happy," "sad," "angry," "surprise," and "neutral." Images were resized to a uniform 48x48 pixels to match the input dimensions for the CNN and made to be grayscale to reduce computational complexity.

To accomplish this, we utilized StandardScaler for normalization of music features and ImageDataGenerator for the image aug mention of the emotion data. For the latter, techniques like random rotations, horizontal flips, and slight shifts were applied to help the model learn to recognize emotions from various angles and facial orientations.

2. Model Selection and Hyperparameter Tuning
Where the primary models we used included:

- CNN for music data
  Where the layers processed the audio features derived from the Spotify API including danceability, energy, and valence. These layers are two Conv1D layers with 64 and 128 filters, respectively, each followed by Batch Normalization and MaxPooling1D layers. The layers are meant to find meaningful patterns that would be useful for future recommendations on the basis of musical characteristics. Dropout layers with a rate of 0.3 are applied to prevent overfitting. And lastly, we used dense layers for classification, with a final softmax activation to output probabilities for each mood category. Optimal hyperparameters were determined empirically (batch_size=32). And the model was trained with categorical cross-entropy loss and the Adam optimizer.

- CNN for emotion data
  Which was trained on the FER dataset, which has the facial images of peoples emotions such as "happy", "sad", "neutral" using Conv2D layers with Dropout (also of 0.3) as well as MaxPooling in order to capture emotion-related features in the images. The Dense and Dropout layers worked to enhance abstraction and generalization while the Adam optimizer and categorical cross-entropy helped in efficient, accurate multi-class classification. And the final softmax layer providing probabilities for each emotion class.

- RNN (LSTM) for music data
  Which worked to capture sequential patterns in the audio features. It was thought

that LSTM layers might work to better capture mood given the fact that they are tuned to understand sequential audio dependencies. The RNN works to implement this dual-layer LSTM architecture for sequential music data processing, where the first LSTM layer (32-128 tunable units) is able to maintain temporal sequences and feed them into a second layer with a similar capacity. The model concludes with dense layers for classification and uses dropout throughout (0.2-0.5 rate). For hyperparameter tuning we used the KerasTuner as discussed below.

KerasTuner was primarily employed for hyperparameter tuning, optimizing the number of filters, dropout rates, and layer sizes to enhance the model's accuracy and prevent overfitting. This hyperparameter tuning gives us the best model that is visualized in Figure 6.

3. Evaluation and Visualization
In our evaluation of each model, we utilized several metrics such as accuracy, confusion matrix, as well as precision-recall and ROC curves. We analyzed and visualized model performance by plotting training and validation accuracy/loss, which also aided in overfitting detection. According to these evaluations, we adjusted hyperparameters and applied regularization techniques to optimize performance as necessary.

4. Recommendation and Attention

Given our trained emotion model, the recommendation function works to recommend a certain set of songs based on a detected emotion, enhancing the personalization of song suggestions. we tried to utilize other methods as well. Attention was added to the RNN model to help it focus on specific parts of the sequential data; this should have enhanced music feature analysis by allowing the model to weigh the most significant features more heavily. This employs a custom feature layer with temperature scaling and L1 regularization, which works to learn the weight the importance of different audio features during training through trainable attention weights initialized uniformly between -1 and 1. The model architechture processes the attention-weight features with batch normalization and dropout, followed by dense layers that learn discriminative genre patterns, while the attention mechanism helps determine which audio characteristics are most essential for classification and subsequent recommendations.

2. Results

Describe your results in a succinct and effective way.

track	artist	uri	danceability	energy	key	loudness	mode	speechiness	acousticness	instrumentalness	liveness	valence	tempo	duration_ms	time_signature	chorus_hit	sections	target	decade	mood
Wild Things	Alessia Cara	spotify:track:2ZyuwVvV6Z3XJaXIFbspeE	0.741	0.626	1	-4.826	0	0.0886	0.02	0.0	0.0828	0.706	108.029	188493	4	41.18681	10	1	2010s	happy
Surfboard	Esquivel!	spotify:track:61APOtq2SSCMuK0VSwi2Kgp	0.447	0.247	5	-14.661	0	0.0346	0.871	0.814	0.0946	0.25	155.489	176880	3	33.18083	9	0	2010s	sad
Love Someone	Lukas Graham	spotify:track:2JqnpexI09dmvjUMCaLCLJ	0.55	0.415	9	-6.557	0	0.052	0.161	0.0	0.108	0.274	172.065	205463	4	44.89147	9	1	2010s	sad
Music To My Ears (feat. Tory Lanez)	Keys N Krates	spotify:track:0cjfLhk8WJ3etPTCseKKbk	0.502	0.648	0	-5.698	0	0.0527	0.00513	0.0	0.204	0.291	91.837	193043	4	29.52521	7	0	2010s	energetic

Figure 1: Our music and mood dataset

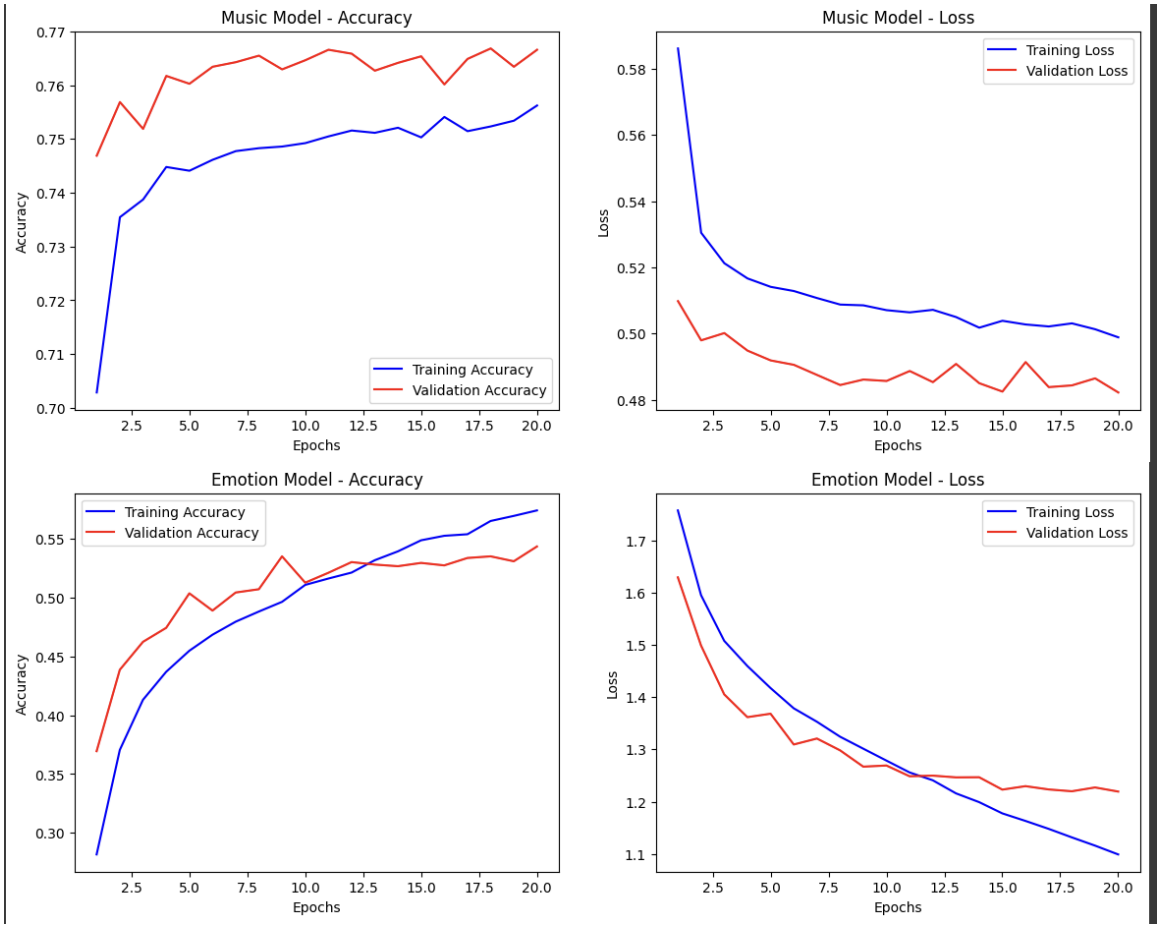

Figure 2: Music and Emotion Models with Accuracy and Loss Over Epochs

The music model performs well with stable accuracy and loss, suggesting effective generalization to unseen data. While for the emotion model, the pattern of accuracy and loss with fluctuation in the later epochs suggests that the model is learning, but also indicates overfitting or the model struggling to generalize, which may require further improvement and augmentation.

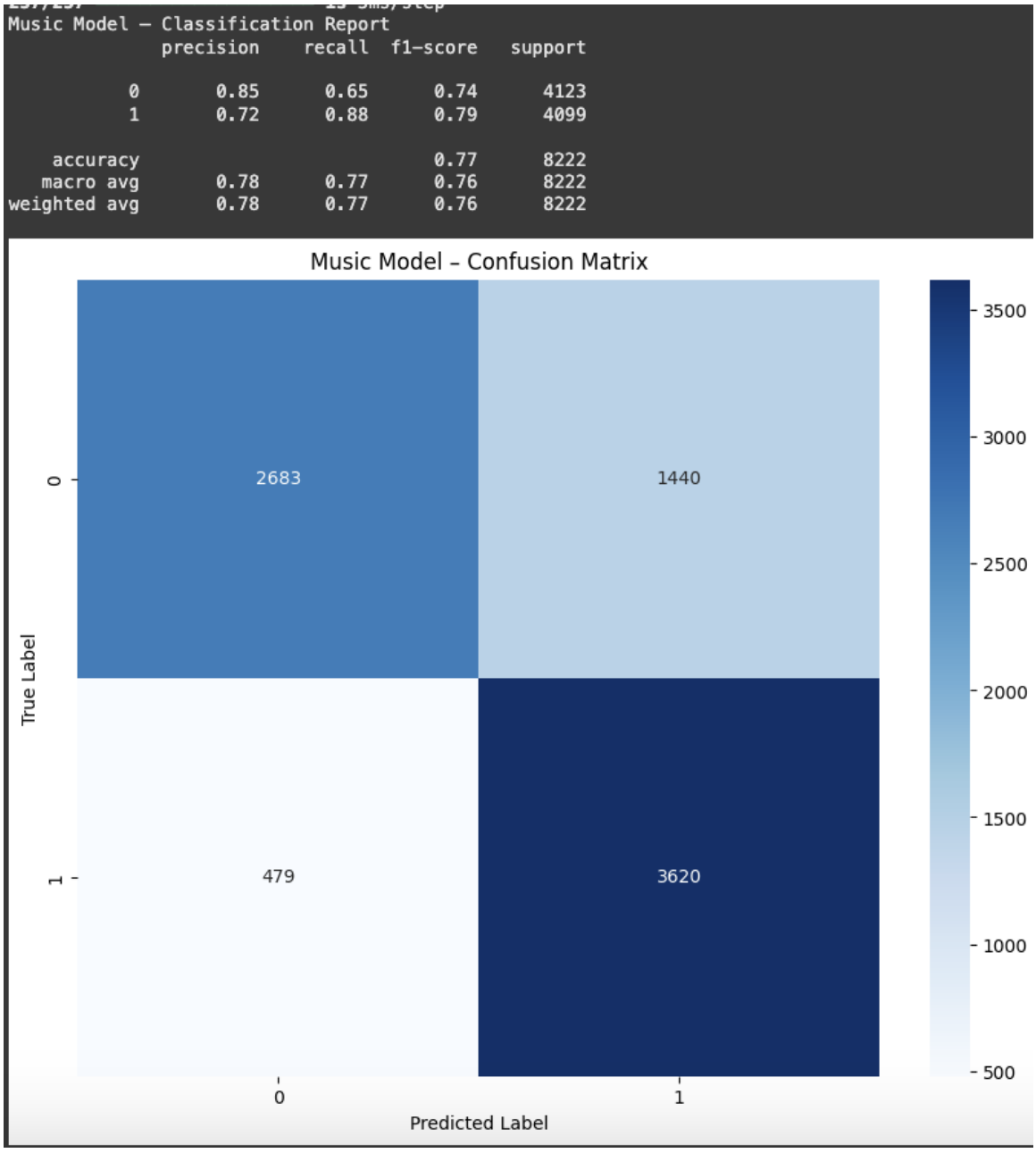

Figure 3: Confusion Matrix of the Music Model

From the confusion matrix and classification report of music model, as binary classification of whether it is a hit song or not, it gives a high-level overview that the model is balanced and perform well currently. A note on this one is that we are using the hit song classification as a place holder for the multiclass classification with all of the moods in the dataset, since we still need to label encode them and we are currently working on that.

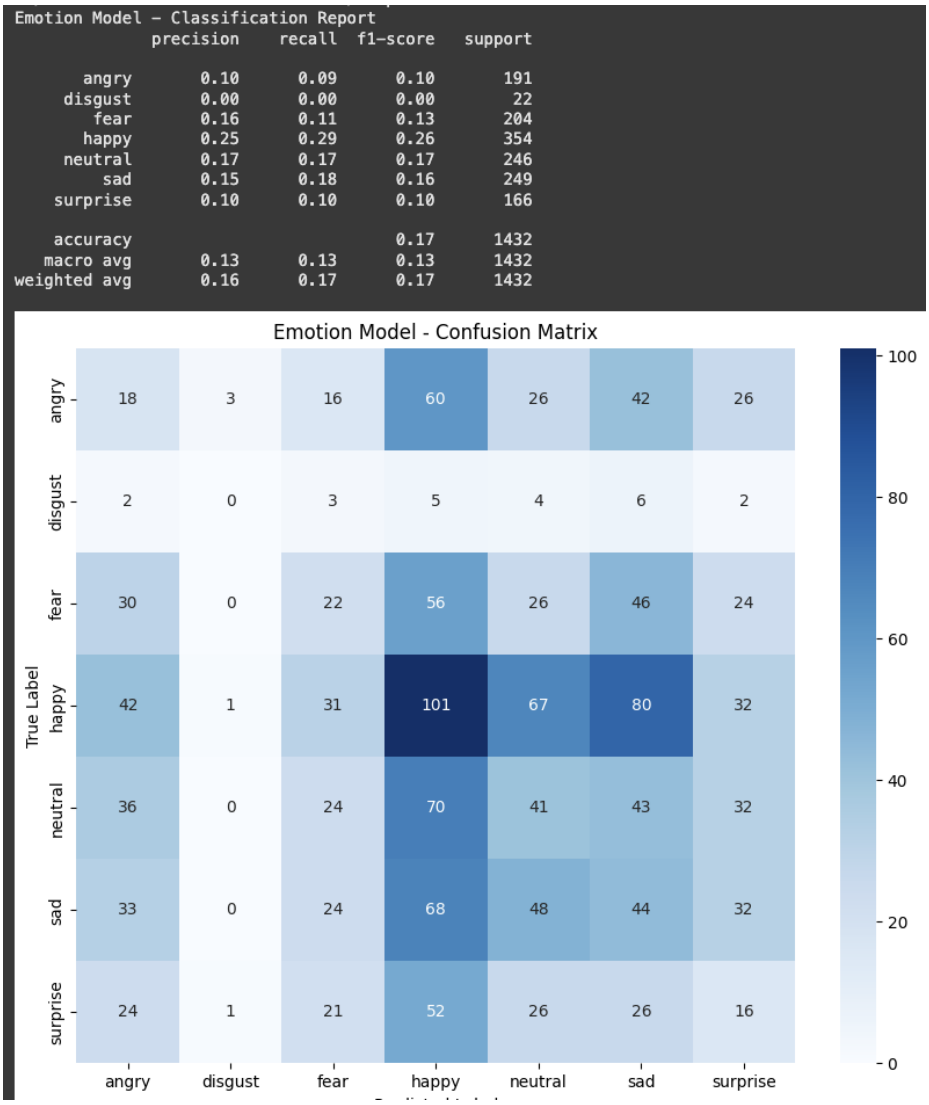

Figure 4: Emotion/Face Image Model's Confusion Matrix

Though with over 50% accuracy in total for the model, there's a class imbalance of the dataset and the model has a tendency to favor majority classes like happiness.Some more improvements are needed, such as balancing the dataset or using class weights, to enhance the model's ability to recognize minority emotions and achieve better balanced accuracy across all classes.

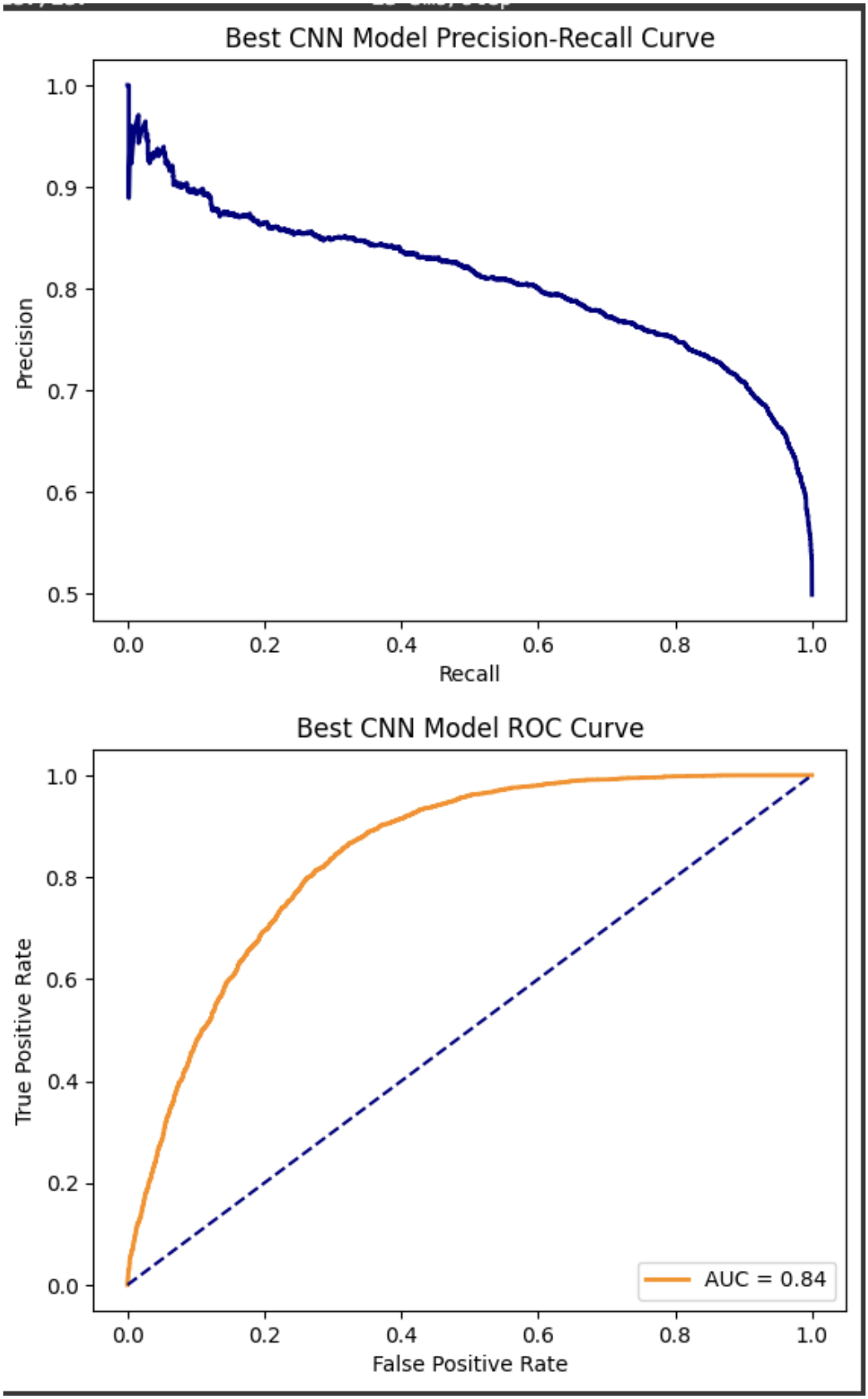

Figure 5: PR and ROC curves for the Music Model

Here we use the Keras Tuner to help with optimize the model architecture and hyperparameters for the music model that predicts whether a song will be a "hit" or not.Also attention-RNN is used to focus on the most relevant parts of the sequence, thereby improving its ability to recognize complex data patterns. The overall shape of the PR curve, where precision remains above 0.5 even at higher recall levels, indicates that the model maintains decent performance, but there's room for improvement in distinguishing between classes, especially as it captures more examples. And the area under the ROC curve (AUC) of 0.84 indicates good classification performance,and it achieves a high TPR with a relatively low FPR, suggesting that it's effective in identifying positives while maintaining a lower rate of false positives.

3. Analysis

What conclusion can you draw from your results? Negative results are fine, as long as you justify them adequately. Are these results expected?

- I think the conclusion we can draw from the results was that the music model did well with the hyperparameter tuning, but we need to apply it to the CNN model for the image detection. Also, I think that once we are able to adjust the music model and the emotion model to hyper parameters. Using the Keras Tuner also was big help in getting the best results. Currenlty, based off of the work we have, then these results seem expected. I believe with more work and as we continue with the wrokplan, I think we will continue getting better results.

4. Plan for additional analysis

- At the moment, it is clear that the most pressing area for improvement is in the emotion recognition model, which is drastically underperforming with 17% accuracy. This poor performance represents a dire bottleneck in our recommendation system; even if our music classification is good, the emotion-based recommendations will be unreliable. The severe class imbalance, given the 22 samples for 'disgust' alone is likely a big contributor, and thus, a high-priority next step would be for us to implement a combination of data augmentation techniques for facial expressions (such as subtle rotations, brightness adjustments, and controlled affine transformations) and use class weights or SMOTE to balance the classes. We might also consider using a pre-trained model such as VGGFace or some more recent vision transformer that has already been trained on facial recognition tasks, but fine-tune it for our particular emotion recognition needs.

  Conversely, the music prediction model is performing considerably better at 77% accuracy, but displays an unfortunate asymmetry in its performance - high precision (0.85) but lower recall (0.65) for non-hits, and the opposite pattern for hits (0.72 precision, 0.88 recall). This suggests that the model is biased in its predictions which has the potential to be problematic. Since our data does originate from the Spotify API, it is possible that we could enhance this through the incorporation of more temporal features (i.e. how a song's popularity changes over time) and

contextual features (artistic historical performance or genre-specific success). The attention mechanism would become more valuable here if implemented and restrained to focus on these new features. Not to mention that since this is time-series data from Spotify, we could also try implementing a time-based validation split instead of random splitting to better mirror real-world usage where one is always predicting future song performance based on past data.

- After fine-tuning the two models, the next logical step is to explore ways to combine them into a cohesive recommendation process. Building connections between the emotion recognition and music classification models could yield a more seamless and personalized recommendation system. If the emotion recognition model continues to underperform, we may consider incorporating supplementary background analysis of the photos to enhance emotion-based recommendations. For example, adding context from the user's environment (e.g., lighting conditions, background objects) could provide additional data to inform the recommendation system. This integrated approach would not only make the system more robust but also ensure that it delivers relevant recommendations even when emotion recognition is challenging.

5. Work plan

Alexandra: Focus on emotion recognition model improvement
Andrea: Focus on music classification and feature engineering
Yuhong: Focus on system integration and data pipeline

6. Weekly Work Plan

Week	**Andrea**	**Yuhong**	**Alexandra**
Week 8 (11/4) Milestone 2	Analyze classification bias Research Spotify features Set up validation	Create data pipeline Set up GitHub Implement logging	Research emotion models Study FER2013 Set up experiments
Week 9 (11/11)	Extract Spotify features Implement validation split Create visualizations	Build preprocessing Create quality checks Automate extraction	Implement augmentation Add class weights Start transfer learning
Week 10 (11/18)	Add temporal features Add genre features Test attention models	Create evaluation Implement testing Start recommendation	Complete transfer learning Test architectures Start ensemble work
Week 11 (11/25)	Finalize classification Create reports Document engineering	Connect pipeline Create architecture	Finalize ensemble Create validation Document architecture
Week 12 (12/2)	Run model tests Create ROC curves Write methodology	Test full system Create metrics Write system docs	Run emotion tests Create visualizations Write technical section
Week 13 (12/9)	Finalize docs Create model slides Fix remaining issues	Finalize system docs Fix integration	Complete model docs Create emotion slides Address model issues
Week 14 (12/16) Milestone 3	Prepare visualizations Create summary Document limitations	Finalize demo Practice presentation Prepare deployment	Practice presentation Create final reports Document improvements

Table 1: Project Timeline

References

Farooq Ansari. The spotify hit predictor dataset (1960-2019). kaggle, 2019.

et al. Baxter, Marissa. Context-based music recommendation algorithm, 2021. URL `https://arxiv.org/abs/2112.10612`.

et al. Gong, Boning. Contextual personalized re-ranking of music recommendations through audio features, 2020. URL `https://arxiv.org/abs/2009.02782.`

Musicblogger. Spotify music data to identify the moods. kaggle, 2020.
(Gong, 2020) (Baxter, 2021) (Ansari, 2019) (Musicblogger, 2020)