Final Project STOR565: Mood in Music

Sanjana Chaudhary, Timothy Deng, Alexander Lin, Leia Reilly, Indira Van Kanegan

Project Summary:

In this project, we explore the classification of mood in music. More specifically, we are guided by a paper that works with our dataset titled *Music Emotion Recognition by Using Chroma Spectrogram and Deep Visual Features* by Er, Mehmet Bilal and Ibrahim Berkan Aydilek. Our dataset was taken from the UC Irvine Machine Learning Repository. It contains 400 total data points with 100 songs for each of the 4 relevant categories, relax, angry, sad, and calm.

Music is a powerful medium that can evoke a wide range of emotions in listeners. In recent years, there has been growing interest in the development of techniques to automatically recognize the emotions expressed in music. Refinements in these techniques could help improve various fields, including, but not limited to, music recommendation systems, music therapy, and human-computer interaction.

One of the most common approaches to music emotion recognition is to use features derived from the chroma spectrogram. The chroma spectrogram is a representation of the frequency content of a music signal, organized in a way that emphasizes the timbral characteristics of the music. Features derived from the chroma spectrogram have been shown to be effective in discriminating between different emotions in music.

In this paper, we explore the use of chroma spectrogram features for musical emotion recognition. We compare the performance of several different classification methods, including support vector machines (SVMs), k-nearest neighbors (KNNs), boosting tree modeling, random forests, multinomial logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and naive Bayes.

Our results suggest that chroma spectrogram features are a promising approach to music emotion recognition. We believe that using machine learning techniques like the ones used in this paper will contribute to the development of more accurate and reliable methods for automatically recognizing the emotions expressed in music.

Disclaimer: The use of AI was not present or used for any part of this project, neither coding nor writing.

Our Data:

As stated in the summary, the dataset comprises of 400 observations, each corresponding to a 30-second audio file of Turkish music. The predictor variables encompass a diverse set of 50 numeric features, providing a comprehensive representation of audio characteristics. For instance, "X_RMSenergy_Mean" reflects the root mean square energy of the audio signal, offering insights into overall signal intensity. "X_Tempo_Mean" captures the average tempo of the music, while "X_MFCC_Mean_1" to "X_MFCC_Mean_13" signify the mean values of the first 13 coefficients of the Mel-frequency cepstral coefficients, encapsulating spectral features. These variables collectively offer a detailed snapshot of the audio's temporal, spectral, and rhythmic attributes.

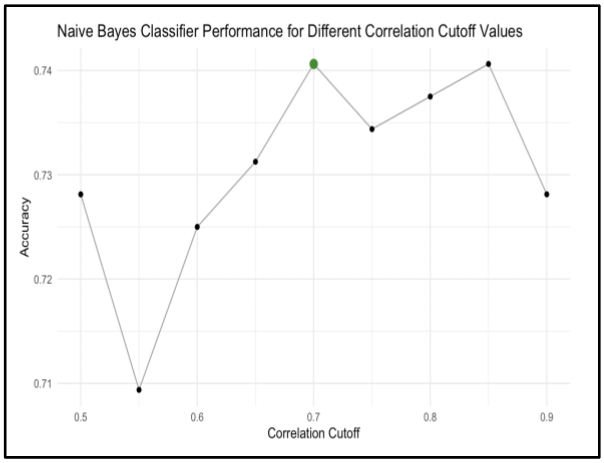
We split the data into a training and test set, with 80% of the data in the training set and 20% in the test set. We used a stratified sampling technique to ensure that the distribution of each class. For a comparison, we tried random sampling without the stratification and found similar test accuracies across the models.

Results:

Naive Bayes

Assuming that there exists independence between the features, the Naive Bayes method predicts by computing the posterior probability of each class given the features. For our data we created three models.

After applying cross-validation, we determined an average test accuracy of **0.7875** for the Naive Bayes model. In an effort to enhance performance, we explored the removal of highly correlated values by systematically identifying and eliminating one variable from each correlated pair. Despite this approach, we observed no improvement in accuracy.



Through cross validation, we found the best correlation cutoff to be 0.7 as seen in the accompanying graph. This provided a training accuracy of 0.74.

Despite removing highly correlated variables with a cutoff of 0.7, the Naive Bayes model's accuracy persisted at **0.7875**, indicating that the model's simplicity and potential complementarity of correlated variables may limit the efficacy of such removal for performance enhancement. The model's reliance on the independence assumption and intricate variable interplay could explain the observed resistance to accuracy improvement through this feature selection approach.

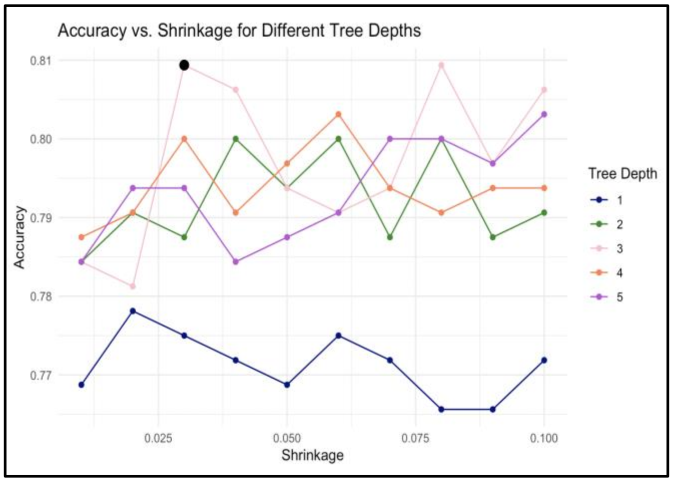
Multinomial Logistic Regression

An extension on binary logistic regression, multinomial logistic regression predicts the probability distribution over all classes as opposed to simply two, which allows it to handle multi-class classification. It is based on the log probability of each class having a linear relationship with the independent attributes.

Our multinomial logistic regression model had an accuracy of **0.725**. Multinomial logistic regression, being more sensitive to overfitting in high-dimensional spaces, may struggle to effectively leverage information from numerous predictors, thus leading to a lower accuracy. We tried removing correlated values in a similar process to that in Naive Bayes; however, that resulted in a lower accuracy rate.

Boosting tree modeling

Boosting is an ensemble technique that creates a single strong learner from a collection of several weak learners, usually decision trees. It functions by gradually training new models to concentrate more on instances that were previously misclassified. We expect boosting to perform well because our dataset had a large number of observations, all numeric.

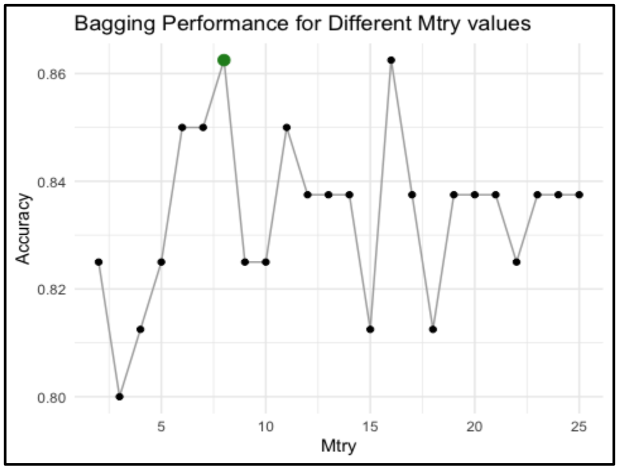


In order to find the best model, we used cross-validation to calculate the optimal tree depth and shrinkage value. The optimal pairing was determined to be a depth of 3 and a learning rate of 0.03.

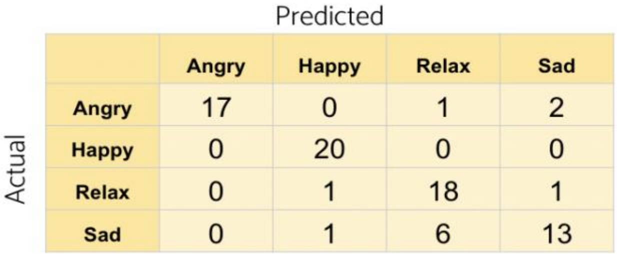
This choice of a tree depth of 3 is well-suited to our dataset, which comprises 50 predictor variables with 4 categories. Given the multi-category classification problem, a depth of 3 allows the decision tree to capture meaningful hierarchical relationships and interactions within the data without overly complex structures. The associated learning rate of 0.03 further complements this by moderating the impact of each weak learner, facilitating a careful and deliberate integration of individual models into a robust model. We found the test accuracy to be **0.8625**, however the individual class accuracies differed greatly from one another. Sad songs were consistently misclassified as relaxing songs at a much higher rate than the other three classes, raising concerns about potential biases or difficulties in capturing the nuances of this specific class. This imbalance in class accuracy suggests the fact that relaxing songs might have more similarities to sad songs than the similarities among other classes.

Random Forest Bagging

In the training phase, the Random Forest employs a method that involves constructing numerous decision trees. These trees collectively contribute to the final prediction by determining the mode of the classes predicted by each individual tree. To enhance diversity among the trees and prevent correlation, the process incorporates randomization. This is achieved by bagging training examples and introducing randomness to the features considered at each split.



For the Random Forest model, we used cross validation to find the optimal number of variables considered at each split. The optimal Mtry value was 8, which was 1 higher than the square root of the number of predictor variables - a typical choice for Random Forest models.



		Predicted			
		Angry	Happy	Relax	Sad
Actual	Angry	17	0	1	2
	Happy	0	20	0	0
	Relax	0	1	18	1
	Sad	0	1	6	13

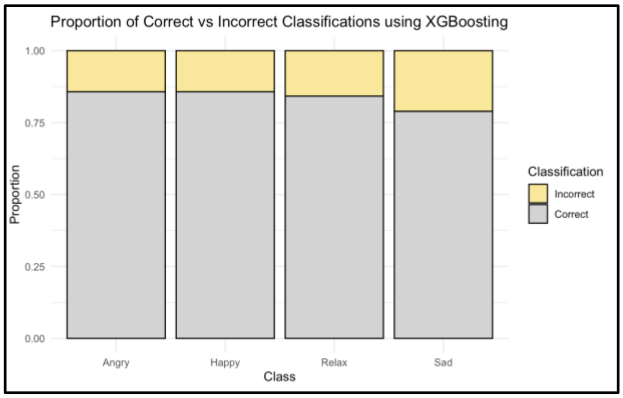
The confusion matrix to the left reveals noteworthy strengths and weaknesses in the Random Forest model's predictive performance. Impressively accurate predictions were observed for the Happy, Angry, and Relaxing classes; however, the model struggled in effectively distinguishing between instances of the Sad and Relax classes. This discrepancy highlights the model's proficiency in certain emotional categories while pinpointing a specific challenge in adequately capturing the nuances between sadness and relaxation. Although the overall test accuracy stands at **0.85**, the difficulty in discerning between Sad and Relax classes leaves us looking to other modeling options.

XGBoosting

Similar to Generalized Boosted Regression, XGBoosting leverages an ensemble approach by amalgamating the predictions of weaker learners to construct a more robust predictive model. Employing a gradient descent boosting algorithm, XGBoosting iteratively builds each tree on the foundation of the preceding one, with a specific emphasis on rectifying the errors made in earlier predictions. This iterative process facilitates the creation of a powerful predictive model that continuously refines its performance by addressing the shortcomings identified in the preceding stages.

However, despite the model's inherent strengths, our specific dataset yielded a test accuracy of **0.8325**, which falls short when compared to the performance of the other tree-based methods. Possible explanations for XGBoosting's lower performance could stem from overfitting concerns, sensitivity to outliers, or suboptimal hyperparameter tuning.

It is noteworthy that XGBoosting exhibits a more evenly distributed misclassification rate across different classes compared to other tree-based methods like random forest bagging and boosting. This balanced misclassification rate is a desirable trait in a model, suggesting that XGBoosting is less prone to disproportionately favoring one category over others. Such evenness in error distribution indicates a more stable and reliable performance across the multiple classes present in our dataset. While the overall accuracy may be lower, this aspect of XGBoosting's performance adds a layer of robustness and fairness, reinforcing its potential as a reliable predictive model in scenarios where balanced class representation is crucial.



Support Vector Machine (SVM)

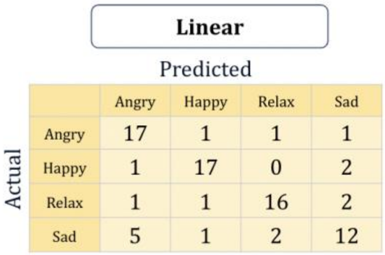
A support vector machine (SVM) operates by seeking an optimal hyperplane to maximize the margin between different classes in a linear fashion. The kernel function, crucial to this process, transforms the original data points into a higher-dimensional feature space. While the linear kernel preserves the original feature space, SVMs become highly versatile when paired with non-linear kernels such as the radial basis function (RBF) and polynomial kernels. The RBF kernel excels in capturing intricate patterns and non-linear relationships by measuring the similarity between data points in the transformed space.

Upon evaluating the SVM models, distinct patterns emerge. Using cross validation, we found the optimal cost value to be 1. This suggests that the best model is one more tolerant of misclassifications, allowing for a more flexible decision boundary that accommodates a certain degree of noise in the data. The linear kernel exhibits a respectable test accuracy of **0.775**, suggesting its effectiveness in capturing linear relationships within the data.

Next we tried SVM with the polynomial kernel. Through cross validation, we found the optimal cost to be 8 and degree of polynomial to be 2. A higher cost value in the SVM reflects a stronger penalty for misclassifying training points, which can lead to a more intricate decision boundary. However, when coupled with a low polynomial degree, the model may struggle to capture the complexity of the underlying patterns in the data, resulting in suboptimal generalization to unseen instances. This combination of parameters could indicate that the model is fitting the noise or specificities of the training data too closely, compromising its ability to perform well on new, unseen data. This was shown to be true with the lower test accuracy of **0.6875**. This may imply that the dataset may not inherently possess polynomial structures, as the test accuracy is relatively low.

The radial support vector machine (SVM) yielded a notably low test accuracy of **0.40,** accompanied by a minimal cost value of 1e-4 and a gamma value of 1. The exceptionally low test accuracy, coupled with a minimal cost value of 1e-4 and a gamma value of 1, suggests that the model is likely overfitting to noise or outliers in the data. The low cost value indicates a high tolerance for misclassifications, allowing the model to fit the training data closely, while the gamma value of 1 implies that the influence of a single training example extends relatively far, potentially capturing noise instead of meaningful patterns.

The confusion matrices below shed light on the performance nuances of SVM models with different kernels. The linear SVM demonstrates robust predictive accuracy for Angry, Happy, and Relaxing classes but encounters challenges in accurately distinguishing instances of the Sad class. Similarly, the polynomial SVM struggles with Relax and Sad classifications. In contrast, the radial SVM exhibits notable limitations across all classes, particularly evident in misclassifications of Relax and Sad instances.

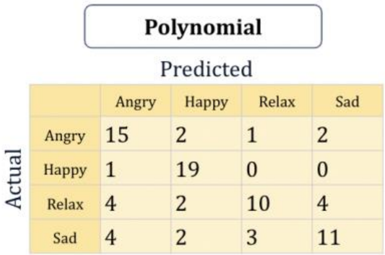


Linear

Predicted

Actual

	Angry	Happy	Relax	Sad
Angry	17	1	1	1
Happy	1	17	0	2
Relax	1	1	16	2
Sad	5	1	2	12

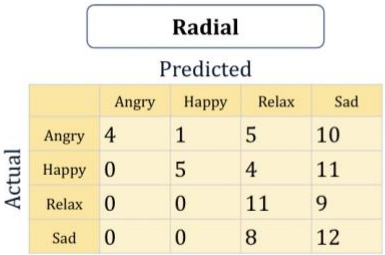


Polynomial

Predicted

Actual

	Angry	Happy	Relax	Sad
Angry	15	2	1	2
Happy	1	19	0	0
Relax	4	2	10	4
Sad	4	2	3	11



Radial

Predicted

Actual

	Angry	Happy	Relax	Sad
Angry	4	1	5	10
Happy	0	5	4	11
Relax	0	0	11	9
Sad	0	0	8	12

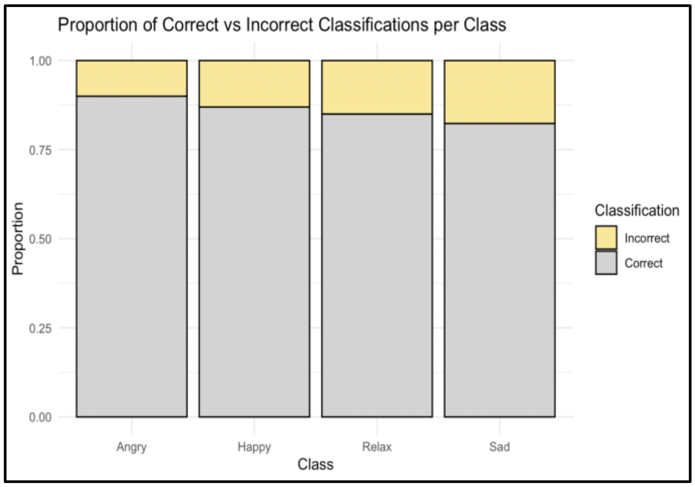
The differences in the models' accuracy and their corresponding optimal parameters indicate that the decision boundary is likely linear instead of radial or polynomial.

LDA (Linear Discriminant Analysis)

Linear Discriminant Analysis (LDA) is a statistical technique designed to identify a linear combination of features that effectively distinguishes between different classes of data. This method operates by minimizing the variance within each class while maximizing the variance between classes. Unlike some models that may be susceptible to overfitting due to high-dimensional data, LDA is advantageous in its ability to reduce dimensionality. By removing less important features that do not contribute significantly to class differentiation, LDA fosters a more parsimonious model that is less prone to overfitting, making it a valuable tool in various statistical and machine learning applications.

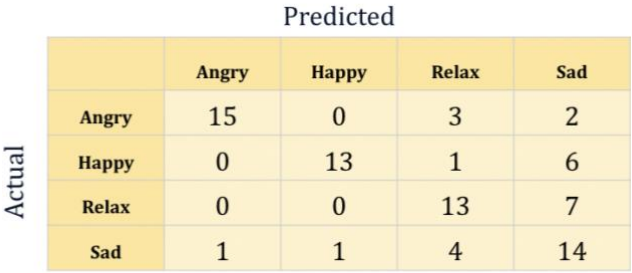
LDA relies on the assumption that variables adhere to a Gaussian distribution. Although a considerable portion of our 50 predictor variables conforms to this distribution, several deviate from it. Initially, we anticipated that this deviation might present a challenge to the LDA model. However, it is noteworthy that LDA demonstrates a notable degree of robustness, with a significant number of the variables exhibiting an approximate Gaussian distribution. This resilience contributes to the efficacy of the LDA model even in the presence of deviations from the ideal distribution in certain predictor variables.

The results of our Linear Discriminant Analysis (LDA) model demonstrate a commendable test accuracy of **0.8625**, positioning it as one of the top-performing models alongside the boosting method. However, the LDA model exhibits a distinct advantage in achieving a more balanced misclassification rate across all four categories compared to the boosting model. While both models share the same overall accuracy, the boosting method's higher misclassification rate for one specific class raises concerns about its robustness in handling that particular category. In contrast, the LDA model showcases its strength in achieving a more uniform distribution of misclassifications across the predictor variable categories. This phenomenon aligns with the fundamental principles of LDA, which strives to maximize the separation between classes while minimizing within-class variance. The boosting method, characterized by its sequential learning approach, may prioritize improving accuracy overall, potentially leading to imbalances in class-specific misclassifications. These findings underscore the significance of considering both overall accuracy and class-specific performance when evaluating classification models, shedding light on the nuanced strengths of LDA in maintaining a more equitable predictive performance across diverse categories.



QDA (Quadratic Discriminant Analysis)

Quadratic discriminant analysis (QDA), a prominent supervised learning classification algorithm, extends the principles of Linear Discriminant Analysis (LDA) by incorporating quadratic functions in its modeling approach. This adaptation allows QDA to capture non-linear relationships within the data, enhancing its flexibility compared to LDA. Despite this increased flexibility, QDA maintains the Gaussian assumption, a crucial characteristic that contributes to its robustness in handling complex datasets.



Predicted

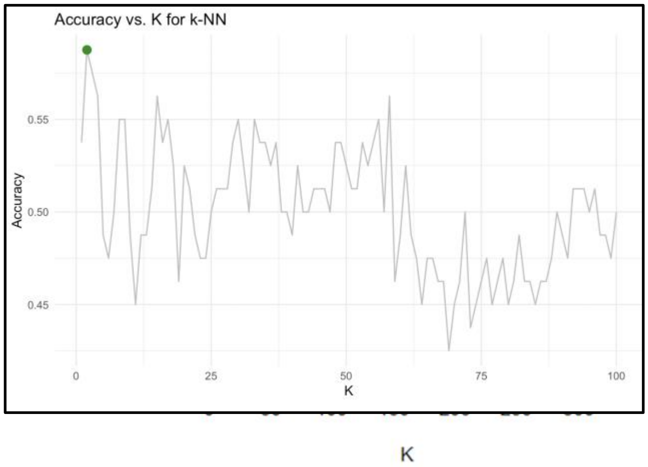
Actual

	Angry	Happy	Relax	Sad
Angry	15	0	3	2
Happy	0	13	1	6
Relax	0	0	13	7
Sad	1	1	4	14

Unfortunately, our QDA model exhibited signs of overfitting with a training accuracy of 100% - indicative of a perfect classification model. However, when subjected to testing, the model displayed a notably lower accuracy of **0.6875**. As shown in the confusion matrix of the test predictions, the Happy and Angry classes had low misclassification rates but the Relax and Sad classes exhibited suboptimal performance. The confusion matrix also reveals a bias towards the Relax and Sad genres and incorrectly misplaced Angry or Happy observations into Relax or Sad. When compared to LDA, it is clear that our data is linearly separable based on the QDA's model performance versus the LDA's model.

KNN (K-Nearest Neighbor)

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that classifies a new data point based on the majority class of its k nearest neighbors in the feature space. The algorithm operates on the assumption that similar instances share similar labels. However, its effectiveness can be compromised by the curse of dimensionality, as the model's performance tends to degrade in high-dimensional spaces. Additionally, KNN is sensitive to outliers and noise, and the choice of the parameter k requires careful consideration to balance bias and variance in the model.



As shown in the plot to the right, we used cross validation to find the optimal k value. Choosing the optimal k value is crucial for the success of KNN because it directly influences the trade-off between bias and variance in the model. A smaller k value can lead to a more flexible model that closely fits the training data but may be sensitive to noise, resulting in overfitting. We found the optimal value for our training dataset to be 2.

We obtained a test accuracy of **0.5875** for the K-Nearest Neighbors (KNN) model, using a k-value of 2. The relatively lower test accuracy may be attributed to the challenges posed by the high dimensionality of the dataset, featuring 50 predictor variables across four distinct categories. KNN models are known to be sensitive to the curse of dimensionality, where the effectiveness of the algorithm diminishes as the number of predictors increases. In this context, the sheer volume of predictor variables probably led to increased computational complexity and a reduced ability of KNN to discern meaningful patterns in the data.

Conclusion and Limitations

We encountered certain limitations in our modeling process. Firstly, the data did not conform to all the necessary assumptions for the individual models, such as the Gaussian assumption for Linear Discriminant Analysis (LDA). Furthermore, our dataset comprised only 400 instances, and having a larger set of data points could have proven beneficial. Additionally, the classification of variables into the four categories was not within our control but instead one of the variables in our dataset. Therefore, further investigation into the potential overlap in how songs were categorized into each group could offer valuable insights, particularly regarding the less accurate predictions for the "Relaxed" and "Sad" categories.

In summary, our ability to classify emotional categories achieved an accuracy range of approximately 70% to 85% on the test set. This suggests that features within a song can serve as reliable indicators of its overall mood. Notably, both Linear Discriminant Analysis and Boosting Tree Modeling demonstrated a high accuracy rate of 86.25%. However, our preference lies with LDA due to the presence of a linear boundary and, more importantly, the even distribution of misclassifications. This even distribution contrasts with the uneven misclassification rates observed in Boosting Tree Modeling, making LDA a more robust choice.

Throughout our modeling process, key predictors such as harmonic change, tempo, spectral centroid, and chromogram means emerged with high variable importance. These factors proved instrumental in capturing the emotional signatures within the songs. It's worth noting that not all emotional classes exhibited the same prediction accuracy. Happy and Anger consistently achieved prediction rates of approximately 80% to 90%, while Sad and Relaxed faced challenges with lower accuracy depending on the model employed.

Future Plans

To enhance the robustness of our approach, several avenues for improvement can be explored. One key strategy involves augmenting our dataset by collecting more extensive training data. This expansion aims to bolster model accuracy and generalization. Additionally, we can enrich the input modalities by incorporating user-generated content such as comments, ratings, and listening patterns, offering a more comprehensive understanding of the emotional impact of a song on listeners. Techniques like SHAP values could be employed to pinpoint the specific sections of a song's audio that wield the greatest influence on mood classification.

In terms of model refinement, instead of employing individual models sequentially, an alternative approach is to leverage ensemble methods, such as voting or stacking, to amalgamate the strengths of multiple models. This ensemble strategy has the potential to elevate overall model performance. Furthermore, future iterations could delve into more sophisticated neural network architectures, including Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Tailoring these architectures specifically for audio analysis could significantly enhance accuracy, particularly when dealing with intricate feature representations. As our models become more adept, they could be integrated into practical applications, such as systems for personalized music recommendations and playlist generators, extending the utility of our research into real-world scenarios.

Bibliography

Er, Mehmet Bilal and Ibrahim Berkan Aydilek. "Music Emotion Recognition by Using Chroma

Spectrogram and Deep Visual Features." Int. J. Comput. Intell. Syst. 12 (2019): 1622-1634.

Relevant Code from our project can be found through this link:

Code for Final Project